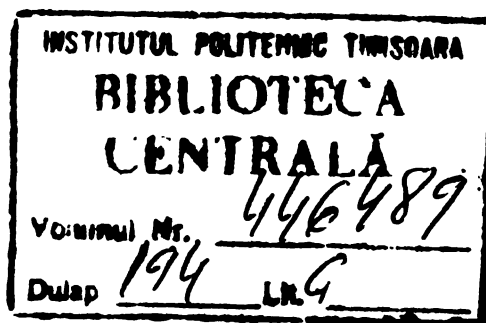


MINISTERUL ÎNDRUMĂRII ȘI ÎNSTRĂRII  
INSTITUTUL POLITEHNIC DE CONSTRUCȚII ȘI MĂȘINI  
TIMIȘOARA

ING. SĂBĂDOARĂ ȘTEFAN

CEZA PENTRU OBTINEREA TITLULUI DE  
DOCTOR ÎN INGINERIE

SISTEM DE SUPRĂÎNCĂLZIRE PENTRU ÎNCĂLZIREA  
ÎN PĂRȚI CALORIFICE (194-195)



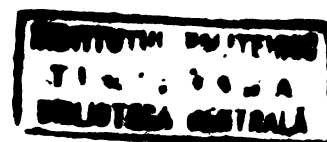
BIBLIOTECA CENTRALĂ  
UNIVERSITATEA "POLITEHNICA"  
TIMIȘOARA

CONFIRMARE DE ÎNREGISTRARE

Prof. Dr. Ștefan Ștefănescu

## C U P R I N S

1. PREZENTAREA GENERALA A SISTEMULUI DE EXPLOATARE	
PENTRU LEGATURA UC-UC . . . . .	pag. 1
1.1. Introducere . . . . .	" 1
Prezentarea lucrării . . . . .	" 5
1.2. Descrierea sistemului de exploatare pentru legă- tura UC-UC . . . . .	" 7
2. DEFINIREA SISTEMULUI DE EXPLOATARE;	
CONCEPTIA DE REALIZARE . . . . .	" 9
2.1. Introducere . . . . .	" 9
2.2. Organizarea generală a sistemului . . . . .	" 12
2.3. Procedura de exploatare pentru legătura UC-UC . . . . .	" 21
Comunicația . . . . .	" 24
Emisia . . . . .	" 29
Recepția . . . . .	" 36
2.4. Procedura de comandă . . . . .	" 42
Comenzi de stabilirea legăturii UC-UC . . . . .	" 44
Comenzi de efectuarea schimburilor de date în linie . . . . .	" 49
Comenzi de suspendare a schimburilor de date . . . . .	" 51
Efectuarea schimburilor în linie . . . . .	" 52
2.5. Procedura de gestiune a sarcinilor . . . . .	" 54
Sincronizarea proceselor. Semafoare . . . . .	" 56
Alocarea resurselor . . . . .	" 67
3. EVALUAREA PARAMETRILOR DE BAZA AI SISTEMULUI . . . . .	
3.1. Introducere . . . . .	" 72
3.2. Concepția de modelare matematică . . . . .	" 79
3.3. Metoda de calcul analitică, exactă . . . . .	" 82
3.4. Metode de calcul prin aproximare . . . . .	" 90
4. SISTEMUL DE EXPLOATARE PENTRU LEGATURA UC-UC, INTRE DOUA CALCULATOARE DE TIP FELIX C-256.	
PRODUSUL PROGRAM . . . . .	" 108
4.1. Introducere . . . . .	" 108
4.2. Procedura de exploatare pentru legătura UC-UC la calculatoarele Felix C-256 . . . . .	" 114
Recepția și interpretarea mesajelor de la mașina de scris - pupitru . . . . .	" 115



Lectura în fișiere . . . . .	pag.	118
Formarea mesajelor de transmis în linie . . . . .	"	121
Emisia mesajelor în linie . . . . .	"	126
Recepția și interpretarea mesajelor din linie . . . . .	"	126
Acrierea în fișiere . . . . .	"	131
4.3. Sistemul de gestiune sarcini . . . . .	"	133
Funcțiunile sistemului de gestiune sarcini . . . . .	"	134
Tabelele sistemului de gestiune sarcini . . . . .	"	137
Subprogramele sistemului de gestiune sarcini . . . . .	"	140
4.4. Comenzile sistemului de exploatare pentru legătura UC-UC . . . . .	"	148
Limbajul de comandă . . . . .	"	149
Comenzile legăturii UC-UC . . . . .	"	150
Prelucrarea comenzilor . . . . .	"	154
4.5. Extinderea sistemului de exploatare pentru legătura UC-UC la o rețea de calculatoare . . . . .	"	157
4.6. Experimentarea produsului program corespunzător sistemului . . . . .	"	159
5. CONCLUZII . . . . .	"	166

## A N E X E

1. Programele sursă (Procedura de exploatare, Sistemul de gestiune sarcini, Comenzi);
2. Lista obținută la mașina de scris - pupitru (consolog);
3. Vidajul partiției 1;
4. Vidajul partiției 2;
5. Datele transmise.

## 1. PREZENTAREA GENERALA A SISTEMULUI DE EXPLOATARE

### PENTRU LEGATURA UC-UC

#### 1.1. INTRODUCERE

Infăptuirea mărețelor obiective ce stau în fața societății noastre socialiste, înscrisă pe traiectoria progresului rapid și multilateral, este legată într-o măsură crescîndă de introducerea în toate domeniile de activitate a informaticii și în mod deosebit a teleinformaticii, tehnică bazată pe prelucrarea datelor la distanță.

Pe baza acumulărilor în procesul de formare a structurilor informatice ale țării noastre, "Programul-directivă de cercetare științifică, dezvoltare tehnologică și introducere a progresului tehnic în perioada 1981-1990 și direcțiile principale pînă în anul 2000", adoptat la cel de-al XII-lea Congres al partidului, prevede continuarea într-o concepție unitară a eforturilor de dezvoltare a informaticii românești.

Programul directivă menționează, în mod expres, că "Cercetarea științifică va trebui să elaboreze soluții optime pentru construirea și funcționarea Sistemului informatic național, inclusiv a Rețelei de calculatoare și de transmisie a datelor, care va marca o schimbare profundă în conducerea vieții economico-sociale".

Se poate arăta că, în general, pe plan mondial în ultima perioadă de timp, în special în ultimul deceniu, s-a trecut vertiginos la interconectarea sau legătura cu ajutorul liniilor de transmisie, în mod singular sau într-un ansamblu, a două sau mai multe calculatoare; astfel, s-au dezvoltat puternice sisteme de teleprelucrarea datelor și rețele generale de calculatoare prin care utilizatorii pot beneficia în orice moment de cele mai variate resurse fizice și logice, situate în diferite puncte geografice, asigurînd astfel servicii de teleprelucrarea datelor, respectiv de culegere date de la sursă, prelucrarea acestora în interiorul rețelei la un anumit calculator și prezentarea rezultatelor acolo unde se dorește de către utilizator.

Această nouă tehnică oferă largi posibilități în informatică permițînd încărcarea eficientă a resurselor de calcul (memorii, unități centrale, unități de schimburi multiple, periferice, linii de comunicații, etc), creșterea vitezei de prelucrare, sporirea gradului de

fiabilitate, mărirea capacității de calcul prin sporirea numărului de echipamente și dispozitive, echilibrarea sarcinilor calculatoarelor și altele.

Interesul pentru elaborarea și introducerea sistemelor de teleprelucrare și a rețelelor de calculatoare a crescut foarte mult în ultima perioadă de timp existînd o veritabilă explozie în acest domeniu, unde foarte multe calculatoare noi sînt instalate în medii de teleprelucrare. Totodată, o bogată literatură de specialitate ilustrează numeroasele rezultate ale cercetărilor teoretice și realizărilor practice din acest domeniu. Toate acestea arată că în informatică s-a conturat un domeniu nou, de mare eficiență, acela al teleprelucrării datelor și al rețelelor de calculatoare.

Astfel, în străinătate au fost elaborate și date în funcțiune rețele generale de calculatoare dintre care mai importante sînt: ARPA și ARPANET anul 1972 - USA, TRANSPAC anul 1978 - Franța, TELENET anul 1970 - SUA, EURONET anul 1976 - Europa și altele, precum și rețele specializate dintre care menționăm cîteva: rezervarea de locuri de avion (AIR FRANCE, SABRE, BEA, etc), colectarea și prelucrarea de date meteorologice (METEO), telecomunicații (SITA) și altele. De asemenea sînt realizate pe plan mondial o serie de sisteme moderne complexe de informatică ca de exemplu: ESOPE-sistem conversațional în timp partajat, SYNCOP-sistem de comutare de procese pentru teleinformatică și rețele de calculatoare, SPECTRE-sistem pentru exploatare și comandă în timp real, CP/CMS-IBM-sistem de partajare resurse și altele.

În țara noastră introducerea sistemelor informatice în toate sectoarele de activitate și în acest cadru, la toate nivelele de conducere și execuție și integrarea acestora progresiv în Sistemul informatic național impune utilizarea celor mai moderne tehnici de culegere, prelucrare, stocare și transmitere la distanță a datelor, respectiv a sistemelor de teleprelucrare și rețelelor de calculatoare electronice pe baza echipamentelor fabricate în țară (de tip Felix).

În acest context au existat și există preocupări privind realizarea de sisteme de teleprelucrare și de rețele de calculatoare.

S-au obținut astfel, unele rezultate pozitive în ceea ce privește crearea și dezvoltarea bazei materiale, în prezent fabricația românească acoperă, într-o oarecare măsură, cerințele privind echipamentele de teleprelucrare (calculatoare de teleprelucrare și chiar cele de comutare de mesaje precum și celelalte echipamente necesare). De asemenea, există preocupări și rezultate privind ela-

borarea și introducerea de sisteme de teleprelucrare; încercările făcute pînă în prezent în acest domeniu s-au bazat, în special, pe software-ul existent al calculatorului Felix C-256, care este totuși limitat în posibilități și performanță.

In continuare se preconizează, ca urmare a studiilor efectuate, elaborarea și darea în exploatare, în următoarea perioadă de timp, a Rețelei naționale de calculatoare care este destinată a forma suportul Sistemului informatic național. Realizarea unei Rețele naționale de calculatoare este o problemă deosebit de complexă de rezolvat necesitînd un efort sporit atît în fabricarea de calculatoare rapide și de mare capacitate și în realizarea de echipamente de telecomunicații cît și în elaborarea de software de teleprelucrare.

In acest domeniu de vîrf al software-ului de teleprelucrare, la care se referă și lucrarea noastră, există mari cerințe și de aceea rezultă necesitatea ca în continuare să se intensifice lucrările de cercetare, proiectare, și implementare prin abordarea de aplicații concrete.

Puteam arăta totuși că realizările existente în acest domeniu pe plan mondial, datorită complexității lor și în absența unei teorii generale unitare, prezintă o serie de deficiențe: nu sînt bine fundamentate matematic în ceea ce privește determinarea parametrilor pentru a optima performanțele și dimensionarea sistemului, nu dispune de metode generale de calcul și de construire, au o arie de cuprindere limitată, nu sînt dezvoltate sistematizat ci la cerere după necesități, rezolvă unilateral probleme, etc.

Avînd în vedere perspectivele deosebite pe care le oferă introducerea și dezvoltarea sistemelor de teleprelucrarea datelor și rețelelor de calculatoare în raport cu imperativele dezvoltării economico-sociale ale țării noastre și încercînd a elimina deficiențele menționate mai sus în acest domeniu, mi-am propus elaborarea de lucrări de cercetare, proiectare, implementare și experimentare a unui sistem de teleprelucrarea datelor (pentru legătura între unitățile centrale a două calculatoare), cu extindere la o rețea de calculatoare, respectiv a unui software de bază de teleprelucrare generalizabil concretizat într-un produs program și pe care l-am denumit "Sistem de exploatare pentru legătura UC-UC".

Lucrarea noastră reprezintă o realizare științifică proprie, originală, elaborată pe baza conceptelor teoretice și practice personale, cu utilizarea unor idei și metode moderne dezvoltate și adaptate corespunzător cerințelor noastre și poate constitui o contribuție teoretică și practică în vastul și complexul domeniu al

teleprelucrării datelor și rețelelor de calculatoare.

Astfel, prin această lucrare am pus la dispoziție, în ceea ce privește realizarea sistemelor de teleprelucrare generalizabile, o teorie proprie din care am fundamentat concepții teoretice și practice constituite în metode generale de calcul și de construire și cu o valorificare practică printr-un produs program pentru calculatoarele din familia Felix. Realizările teoretice s-au soldat cu metode generale de calcul în ceea ce privește evaluarea parametrilor prin tratarea matematică a comportamentului sistemului și în ceea ce privește arhitectura sistemului, prin definirea elementelor de bază ale acestuia. Realizările practice s-au finalizat cu metode privind concepția aplicativă, proiectarea tehnică, construirea programelor, implementarea și experimentarea sistemului, respectiv a produsului program pentru calculatorul Felix. Am considerat că elaborarea unei concepții teoretice, fără o aplicație practică, concretă, nu este suficientă și nu prezintă interes în informatică și mai cu seamă în domeniul teleprelucrării datelor unde activitatea este încă la început și numai o realizare practică ne permite o validare a soluțiilor teoretice.

Pot arăta că, pe lângă importanța deosebită privind realizarea și introducerea sistemelor de teleprelucrare în economie în etapa actuală, am optat pentru abordarea acestei tematici și datorită faptului că personal am avut preocupări de cercetare, de proiectare, și de implementare cu rezultate concrete în acest domeniu, lucrări care sînt prezentate în bibliografie; în fond, prezenta lucrare este o continuare a lucrărilor anterioare-care mi-au servit ca sursă de inspirație la elaborarea concepției generale teoretice și practice în special la definirea și stabilirea comportamentului și arhitecturii sistemului, situîndu-se la un nivel superior, prin noile contribuții teoretice și practice și prin utilizarea unor tehnici și metode moderne.

Totodată, am avut în vedere, preluînd critic și adaptat la specificul nostru, principii și metode din realizările cele mai importante existente pînă în prezent privind sistemele de teleprelucrare și rețelele de calculatoare din țară și din străinătate (prezentate în bibliografie).

Această lucrare a fost posibil de realizat datorită îndrumării competente și permanente atît științific cît și metodologic pe care mi-a acordat-o tovarășul Profesor Alexandru Rogoian căruia îi exprim pe această cale sincere mulțumiri. De asemenea mulțumesc celorlalți tovarăși care au sprijinit realizarea acestei lucrări.

## PREZENTAREA LUCRĂRII

Capitolul I. Se arată necesitatea și oportunitatea realizării unui sistem modern de teleprelucrare generalizabil și implicit elaborarea unor metode generale teoretice și practice de construire de astfel de sisteme, avînd în vedere, atît cerințele economiei naționale, avantajele și eficiența introducerii teleprelucrării cît și absența unei teorii generale cît și a unor concepții unitare teoretice și practice, lipsa unor performanțe optime și a posibilităților de generalizare, etc.

Este prezentat în general Sistemul de exploatare pentru legătura UC-UC. Sistemul asigură și gestionează prelucrările și schimburile de informații la distanță între unitățile centrale a două calculatoare, constituind o prelungire a sistemului de exploatare standard și totodată o interfață cu utilizatorul. Sistemul oferă, ca software de bază specializat pentru teleprelucrare, servicii privind conversia de date (fișiere) și eventual repartizarea de lucrări la distanță. Pentru a realiza aplicațiile de teleprelucrare, sistemul efectuează o serie de activități complexe ca stabilirea și întreruperea legăturilor (subcanalelor) între calculatoare, punerea de acord a proceselor (sarcinilor) din cele două calculatoare pentru ca acestea să se înțeleagă între ele, organizarea logică a informației pentru transmiterea și prelucrarea la distanță și altele.

Capitolul II. Este prezentată concepția teoretică privind elementele de bază ale sistemului, arhitectura acestuia, punînd accent pe organizarea, structura și funcțiile resurselor logice, în situația unui sistem ipotetic realizat pe o mașină virtuală.

Sînt definite funcțiile de bază, structura și organizarea sistemului. Sînt determinate modelele și algoritmi celor trei componente principale ale sistemului: ansamblul resurselor logice (proces) care asigură realizarea aplicației de teleprelucrare. - Procedura de exploatare pentru legătura UC-UC, ansamblul de funcții care permit ordonarea proceselor și care formează un executiv. - Procedura de gestiune sarcini, ansamblul de funcții transpuse în comenzi, care stabilesc legătura între calculatoare și efectuează schimburile de informații la distanță. - Comenzile sistemului.

S-au dezvoltat cu ajutorul unui aparat matematic adecvat unele probleme de interes, atît în ceea ce privește, sincronizarea proceselor paralele cu ajutorul semafoarelor în două medii (spațiu -



timp) diferite și disjuncte, cât și alocarea de memorie pentru tam-  
poanele de teleprelucrare necesară subcanalelor la executarea comen-  
zii de stabilire a legăturii.

Capitolul III. Se dezvoltă o teorie matematică proprie  
privind analiza comportamentului sau încărcării sistemului pentru  
a determina parametri de bază în vederea obținerii unor performanțe  
superioare și a unei dimensionări optime a resurselor.

Abordarea matematică, este susținută prin instrumente moderne  
și eficiente de matematică ca, teoria așteptării și teoria probabili-  
tăților și a condus la elaborarea unor studii teoretice și în con-  
tinuare a unor metode de calcul generale de rezolvare a sistemelor  
de teleprelucrare și rețelelor de calculatoare. Sistemul este repre-  
zentat prin modele stochastice foarte complexe cu șiruri de aștep-  
tare în serie, cu capacitate finită, etc. și se urmărește determinarea  
încărcării maxime a sistemului, respectiv a numărului maxim de mesa-  
je care pot trece prin sistem în unitatea de timp.

Aceste metode generale de calcul, elaborate comparativ, au atît  
un caracter analitic și sînt limitate la cazurile în care sistemul  
este reprezentat prin modele mai simple, cu număr redus de stații și  
cu distribuție exponențială, cât și un caracter de aproximare (prin  
procese de difuzie și prin echivalență) utilizate pentru cazurile  
complexe în care sistemul este reprezentat prin modele cu un număr  
mare de stații și cu distribuții generale care reflectă cel mai co-  
rect realitatea.

Capitolul IV. Este prezentată realizarea practică a  
sistemului, respectiv a software-ului de bază de teleprelucrare prin-  
tr-un produs program generalizabil pe calculatorul Felix C-256; este  
tratată concepția aplicativă, proiectarea tehnică, construirea pro-  
gramelor, implementarea și experimentarea produsului program.

Plecînd de la concepția teoretică, sînt determinate cu mici modi-  
ficări, componentele de bază ale sistemului:

- Procedura de exploatare pentru legătura UC-UC, care realizea-  
ză aplicația de teleprelucrare cu ajutorul sarcinilor sale și este  
compusă din: Comunicația cu utilizatorul, Emisia și Recepția,

- Sistemul de gestiunea sarcinilor, un executiv pentru calcula-  
torul Felix, care realizează ordonanțarea sarcinilor asigurînd buna  
funcționare a sistemului și este compus din funcții ca: activarea  
sarcinii, dezactivarea sarcinii, punerea în așteptarea sarcinii, etc.

- Comenzile sistemului, care realizează cu ajutorul unui limbaj  
propriu, stabilirea legăturii între calculatoare și efectuarea schim-

burilor de informații la distanță.

S-au utilizat facilitățile calculatorului Felix C-256, hardware și software (legătura fizică directă între memoriile calculatoarelor multiprogramarea, simultaneitatea prelucrării cu intrări/ieșiri, așteptarea multiplă a evenimentelor, gestiunea fișierelor, gestiunea liniei de transmisie și altele) cu adaptările corespunzătoare; s-au introdus modificări în ceea ce privește Sistemul de gestiunea fișierelor și Sistemul de gestiunea transmisiilor.

Produsul program elaborat a fost implementat și experimentat practic pentru aplicația de teleprelucrare referitoare la teleconversia datelor (citirea unui fișier de intrare pe cartele la calculatorul sursă de date și scrierea datelor într-un fișier de ieșire la imprimanta calculatorului receptor de date). S-au testat obiectivele principale ale produsului program ca: transmisia de mesaje între mașinile de scris, transmiterea de comenzi de stabilirea sau închiderea legăturii între calculatoare, transmisia de date între calculatoare.

S-a abordat extinderea sistemului (care anterior s-a referit la o singură legătură bipunct) la o rețea de calculatoare care este considerată, evident, ca un ansamblu de legături bipunct.

## 1.2. DESCRIEREA SISTEMULUI DE EXPLOATARE

### PENTRU LEGATURA UC-UC

Sistemul nostru reprezintă ansamblul procedurilor și resurselor în special logice interdependente, de concepție omogenă, care asigură legătura logică și fizică între unitățile centrale a două calculatoare aflate la distanță, în vederea efectuării schimburilor de informații și prelucrării de date. Sistemul se poate extinde la o rețea de calculatoare, legătura între două calculatoare constituind elementul de bază al unei rețele de calculatoare, aceasta fiind în fond un ansamblu de legături bipunct.

Am tratat numai problemele privind software-ul (ansamblul de programe) necesar pentru legătura între două calculatoare și extinderea la o rețea de calculatoare. Partea de hardware nu a fost luată în considerare presupunând că echipamentele de calcul folosite dispun de elementele necesare interconectării calculatoarelor și transmisiei pe liniile de comunicație conform normelor și standardelor în vigoare.

Sistemul face parte din clasa sistemelor de exploatare dato-

rită caracteristicilor sale: gestionarea proceselor paralele, sincronizarea proceselor cu ajutorul semafoarelor, controlul desfășurării sarcinilor cu ajutorul unui executiv, alocarea optimă a resurselor, coordonarea șirurilor de așteptare, etc; este de fapt, o prelungire a sistemului de exploatare standard al calculatorului, care prezintă insuficiențe și inconveniențe privind aplicațiile de teleprelucrare, însă se situează în raport cu acesta, la un nivel logic superior, avînd posibilitatea de a coordona.

Sistemul poate oferi, în principiu, un ansamblu de servicii orientate spre aplicațiile de teleprelucrare a datelor: conversii de fișiere, interogări și extracții de fișiere, acces direct al unui program la fișiere situate în altă parte, acces la baza de date comună pentru mai mulți utilizatori, partajarea echipamentului de calcul în situații de supraîncărcare și altele. Așa cum a fost realizat concret, sistemul nostru rezolvă aplicațiile referitoare la schimburile de date la distanță respectiv teleconversia de fișiere, simultan în cele două sensuri, adică efectuează transmiterea datelor citite de un calculator, la un fișier al celuilalt calculator și imprimarea la un calculator a informațiilor trimise de la fișierul celuilalt calculator. Ca extindere, poate lua eventual în considerare aplicațiile privind repartizarea de lucrări între calculatoare adică, un program depus în unul dintre calculatoare să poată fi executat în celălalt calculator.

Sistemul constituie o interfață între utilizator și calculator fiind accesibil utilizatorului, cu care poate comunica prin intermediul mașinii de scris pupitru, printr-un limbaj de comandă.

Sistemul asigură sincronizarea și comunicarea între calculatoare cu ajutorul unor comenzi - legate în cazul nostru de o structură a datelor de transmis, introduse de la mașina de scris de pupitru și cu ajutorul unor proceduri - protocoale pe mai multe nivele privind schimburile de informații realizînd un control al transmisiei datelor printr-o reacție de confirmare a primirii mesajului. Sistemul trebuie să realizeze obligatoriu o legătură logică, - subcanal, între calculatorul sursă de date și calculatorul receptor de date. Legătura logică - subcanal, între sursă și receptorul de date permite conectarea a două componente (terminale, periferice, cel mai adesea fișiere).

Practic, pe calculatorul Felix, sistemul face apel direct la facilitățile acestuia, în special ale sistemului de exploatare ca; gestiunea intrărilor/ieșirilor pe linie și derularea schimburilor de informații în linie, simultaneitatea prelucrărilor cu intrări/ieșiri pentru a permite lucru în paralel și a realiza sincronizarea calculatoarelor și altele.

## 2. DEFINIREA SISTEMULUI DE EXPLOATARE; CONCEPTIA DE REALIZARE

### 2.1. INTRODUCERE

În această parte a lucrării am prezentat concepția de realizare a sistemului nostru cu referire la arhitectura internă și care cuprinde principiile de bază privind organizarea, structura și funcționarea resurselor logice, proceselor și resurselor fizice, în special a tampoanelor de memorie și altele.

Pentru a asigura un optim din acest punct de vedere și pentru a răspunde cerințelor actuale, am considerat sistemul nostru ca un sistem ipotetic, care dispune de calculatoare cu atribute de mașini virtuale, unde procesele pot evolua în permanență, execuția lor fiind asigurată cu resurse fizice necesare iar sincronizarea lor fiind realizată cu semafoare.

În partea conceptuală de realizare a Sistemului am acordat o atenție deosebită funcțiilor de bază și organizării Sistemului precum și utilizării unor tehnici și metode informatice ca: procese și sincronizarea lor, alocarea de memorie și altele.

Funcțiile de bază ale Sistemului au fost stabilite ca urmare a cerințelor de teleprelucrare a datelor și se referă la:

- Realizarea aplicațiilor de teleprelucrare prin definirea și elaborarea unor procese paralele care să reflecte funcțiile principale ale aplicației de teleprelucrare;
- Gestionarea proceselor paralele printr-un executiv sau supervisor în vederea ordonanțării acestora;
- Stabilirea legăturii între calculatoare și asigurarea schimburilor de date între acestea cu ajutorul unor comenzi.

Ca urmare, am procedat la construirea sistemului nostru pe care l-am structurat în trei mari părți componente:

- Procedura de exploatare pentru legătura UC - UC,
- Procedura de gestiune a sarcinilor,
- Procedura de comandă.

Arhitectura internă a Sistemului a fost fundamentată și reprezentată prin modele și algoritmi matematici de tip determinist.

Soluționarea, din punct de vedere informatic, a fost realizată

cu ajutorul unor concepte informatice moderne ca: procese paralele, sincronizarea proceselor cu ajutorul semafoarelor, ordonanzarea sau planificarea proceselor, gestiunea șirurilor de așteptare, așteptare multiplă, alocarea dinamică a memoriei și altele.

Am efectuat în această direcție lucrări de cercetare pentru determinarea unei arhitecturi optime de sistem (stabilirea și elaborarea de modele și algoritmi matematici care reprezintă această arhitectură, în special funcțiile de bază) precum și pentru găsirea și utilizarea metodelor și tehnicilor informatice avansate, pe care le-am considerat cele mai corespunzătoare, prin modificări și adaptări adecvate la cerințele noastre.

Prezint în continuare pe scurt, unele lucrări de cercetare personale elaborate pentru soluționarea problemelor de importanță majoră ridicate în cadrul concepției de realizare pe componente a Sistemului.

Astfel, pentru Procedura de exploatare, lucrările s-au axat pe utilizarea conceptului de procese paralele care a fost adaptat și dezvoltat corespunzător; de asemenea, am conceput și elaborat modele proprii care reprezintă funcțiile de bază de teleprelucrare (procesele paralele).

Pentru Procedura de gestiune a sarcinilor, luând în considerare principiile de bază ale unui executiv în general, am trecut la realizarea acestei proceduri utilizând unele concepte din tehnicile și metodele informatice moderne pe care le-am adaptat și dezvoltat conform cerințelor noastre. Conceptul de sincronizare cu ajutorul semafoarelor a proceselor care evoluează într-un singur mediu (spațiu - timp) al lui Dijkstra l-am modificat, pentru a fi utilizat de noi în două medii (spațiu - timp) care să facă posibilă sincronizarea celor două calculatoare la distanță. De asemenea, am luat în considerare tehnici de gestiune a șirurilor de așteptare, de acordare a priorității și altele pentru implementarea lor în această procedură.

Pentru Procedura de comandă am plecat de la unele idei privind protocoalele de comunicație existente și pe baza unor concepte proprii am stabilit funcțiile și algoritmi necesari pentru elaborarea unor comenzi și unui limbaj de comandă propriu.

Se poate arăta, în concluzie, că la elaborarea lucrării am folosit atât unele concepte proprii din lucrări anterioare (bibliografia) cât și unele concepte de realizare arhitecturală a unor sisteme și rețele de calculatoare prezentate în lucrări recente, valoroase, de specialitate (bibliografia cu notația 2); aceste con-

cepte am încercat să le valorific în cadrul lucrării noastre, evident, cu modificările și adaptările necesare în raport cu cerințele noastre. Astfel de lucrări pe care le-am luat în considerare au fost cele referitoare la rețelele generale de calculatoare în special ARPA și CYCLADES și la sisteme moderne de informații ca: ESCORT-conversațional și timp partajat; SYNCOP-comutare procese pentru teleinformatică; SPECTRE-exploatare și comandă în timp real; CP/CMS, IBM-partajare resurse precum și altele.

Aceste lucrări sînt totuși rezolvări particulare din punct de vedere al arhitecturii de sistem, nu sînt sistematizate și sînt deosebit de empirice; ele nu pot oferi o concepție generală de realizare a sistemelor de teleprelucrare sau a rețelelor de calculatoare. Cauzele deficiențelor menționate constau în special în faptul că sistemele respective nu au obiective comune, tehnicile și instrumentele informatice folosite la realizarea lor sînt în general diferite de la caz la caz, iar echipamentele de calcul utilizate în aceste sisteme sînt foarte variate.

Din punct de vedere arhitectural Sistemul nostru este deosebit față de sistemele menționate fiind mai judicios realizat pe ansamblu, funcțiile sînt bine precizate, structura este omogenă și organizarea este corectă. În plus, la elaborarea acestei arhitecturi am folosit cele mai moderne concepte informatice ca: procese paralele, sincronizarea proceselor cu ajutorul semafoarelor, alocare memorie, etc, pe care le-am modificat și îmbunătățit pentru a răspunde eficient nevoilor noastre.

Considerăm că această parte din lucrarea noastră care abordează concepția de realizare a sistemului, cu referire la arhitectura sa internă - principii de bază, funcții, structură generală, organizare și altele, datorită unor sistematizări corecte și a posibilităților de generalizare în ansamblu și pe componente are caracterul unei metode de lucru generală pentru realizarea sistemelor de teleprelucrare constituind prin aceasta o contribuție personală în domeniul realizării sistemelor de teleprelucrare a datelor, cu un puternic caracter de originalitate.

Deci, putem spune în concluzie că am elaborat o metodologie de lucru în ceea ce privește concepția de realizare a sistemelor de teleprelucrare și cu extindere la rețelele de calculatoare.

În continuare prezentăm la nivel de concepție cele trei mari părți componente ale sistemului; Procedura de exploatare pentru legătura UC - UC, Procedura de comandă și Procedura de gestiune a sarcinilor.

## 2.2. ORGANIZAREA GENERALĂ A SISTEMULUI

Studiile și cercetările efectuate de noi în această direcție, privind modul de organizare generală a sistemului nostru, au avut ca scop elaborarea unei concepții teoretice de organizare care cuprinde unele principii de bază de organizare și unele modalități de funcționare; astfel au fost studiate și prezentate probleme esențiale privind sistemul, ca locul, poziția și configurația, organizarea fizică și logică a resurselor, structura ierarhizată și altele.

Aceste lucrări sînt originale și pot aduce o contribuție personală în domeniul concepției de realizare al sistemelor de teleprelucrare și a rețelelor de calculatoare.

Lucrările de specialitate, teoretice și chiar practice apărute recent cu tematică în acest domeniu, prezintă în majoritate un cadru prea general și toate, fie că se referă la rețele de calculatoare sau la sisteme de teleprelucrare, sînt concepute, în ceea ce privește organizarea, pe un specific oarecare și au un caracter particular în funcție de aplicațiile ce urmează să le realizeze sau de echipamentele pe care le folosesc.

Evident, unele studii mi-au servit pentru documentare și chiar au constituit surse de inspirație în ceea ce privește concepția de organizare a sistemului.

Aceste lucrări le-am menționat anterior în bibliografie; cele mai importante însă sînt cele referitoare la ARPA, CYCLADES și altele prezentate de către autori ca Martin, Davies și alți.

Revenind la subiect, arătăm în continuare unele moduri de rezolvare ale problemelor noastre legate de configurația, locul, poziția și conținutul sistemului nostru.

Configurația sistemului nostru este formată din două calculatoare electronice, sau mai multe în cazul unei rețele de calculatoare, dotate cu toate echipamentele necesare teletransmisiei de date și care dispun de un software de bază adecvat pentru teleprelucrare.

Locul sistemului se găsește între nivelul fișierelor și nivelul liniilor de transmisie.

Din punct de vedere logic se situează la un nivel superior în raport cu cele două nivele și în plus, le coordonează.

Accesul la fișiere și la linie se asigură de către sistemul de gestiune a fișierelor - SGF, și Sistemul de gestiune a transmisiilor - SGT, ale calculatorului.

Configurația și locul sistemului de exploatare pentru legătura între calculatoare se arată în fig.2.1.

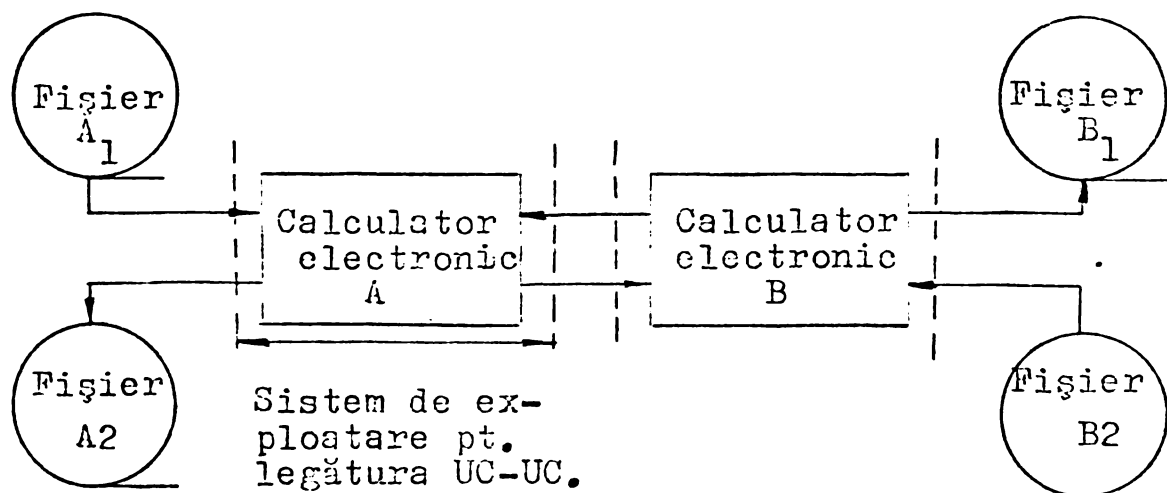


Fig.2.1. Configurația și locul sistemului de exploatare pentru legătura UC - UC.

Poziția sistemului, în raport cu mediul exterior este prezentată în fig.2.2.

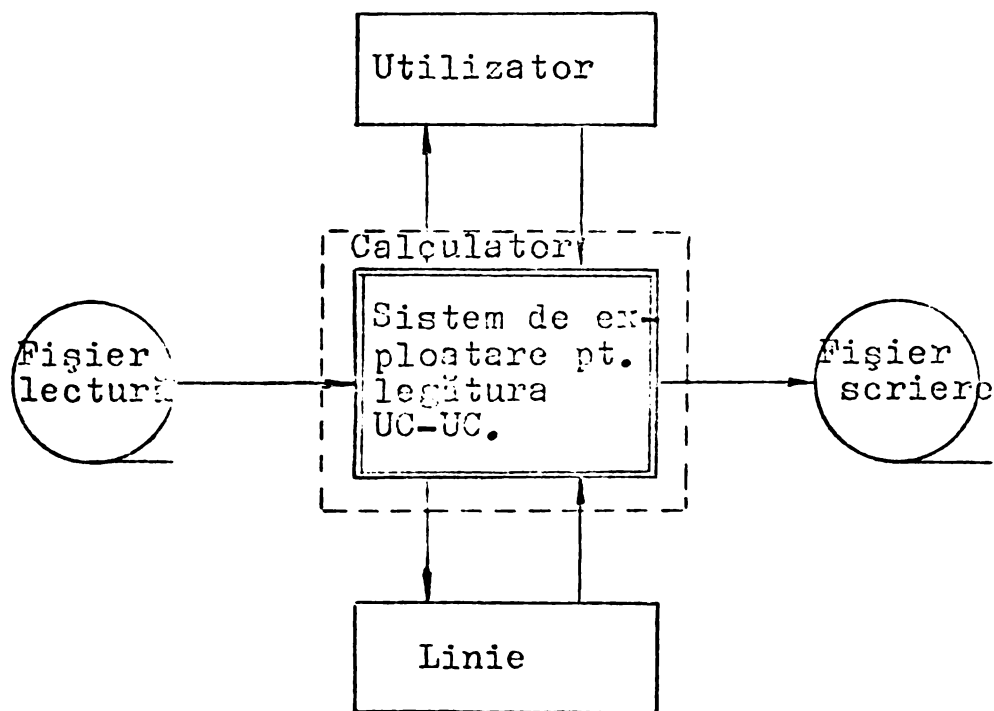


Fig.2.2. Poziția sistemului în raport cu exteriorul

Sistemul cuprinde întreg ansamblu de resurse logice și fizice care realizează aplicația de teleprelucrare a datelor. Intrarea în sistem este considerată a fi fișierul de intrare al sursei de date (calculatorul care emite date în linie), iar ieșirea din sistem este considerată a fi fișierul Receptorului de date (calculatorul



care primește date din linie).

Sistemul gestionează și asigură schimburile de informații între cele două calculatoare dealungul întregului traseu - în intrări (ieșiri), în linie și în mod deosebit în interiorul calculatoarelor respectiv, în unitatea centrală.

Sistemul de exploatare pentru legătura UC - UC, în cazul ipotetic, luând în considerație cerințele maxime de teleprelucrare se compune din trei mari părți componente corespunzătoare funcțiilor de bază ale sistemului: Procedura de exploatare pentru legătura UC - UC, Procedura de gestiunea sarcinilor și Procedura de comandă.

O parte din sistem, trebuie să fie în permanență în memorie la cele două calculatoare, pentru a exista posibilitatea ca acestea să se înțeleagă prin cereri de stabilire a legăturii între calculatoare, introduse de la mașina de scris locală, sau transmise de calculatorul corespondent.

Organizarea sistemului este prezentată în raport cu funcțiile principale ale acestuia.

Funcțiunile principale ale Procedurii de exploatare, Emisia și Recepția, trebuie să dispună de tampon de memorie, organizate corespunzător.

Pentru funcția de Emisie va trebui să existe:

- Un tampon component pentru un subcanal Sursă de date, adică al calculatorului care devine sursă de date, adaptat la mărimea maximă a articolului din fișierul citit;

- Unul sau mai multe tampon de emisie în linie care servesc la emiterea mesajelor.

Funcția Recepție va dispune de:

- Unul sau mai multe tampon de recepție din linie, în care se primesc mesajele provenind de la calculatorul corespondent;

- Tampon component, pentru diferite subcanale (legături) în care articolele din fișier vor fi reconstituite în aceste tampon de memorie.

Fiecare din aceste tampon va fi adaptat la mărimea maximă a articolului.

Aceste tampon, nu există toate în realitate, dar prezentarea în acest fel este sugestivă.

Diferitele organizări posibile ale Sistemului sînt redată în fig.2.3.

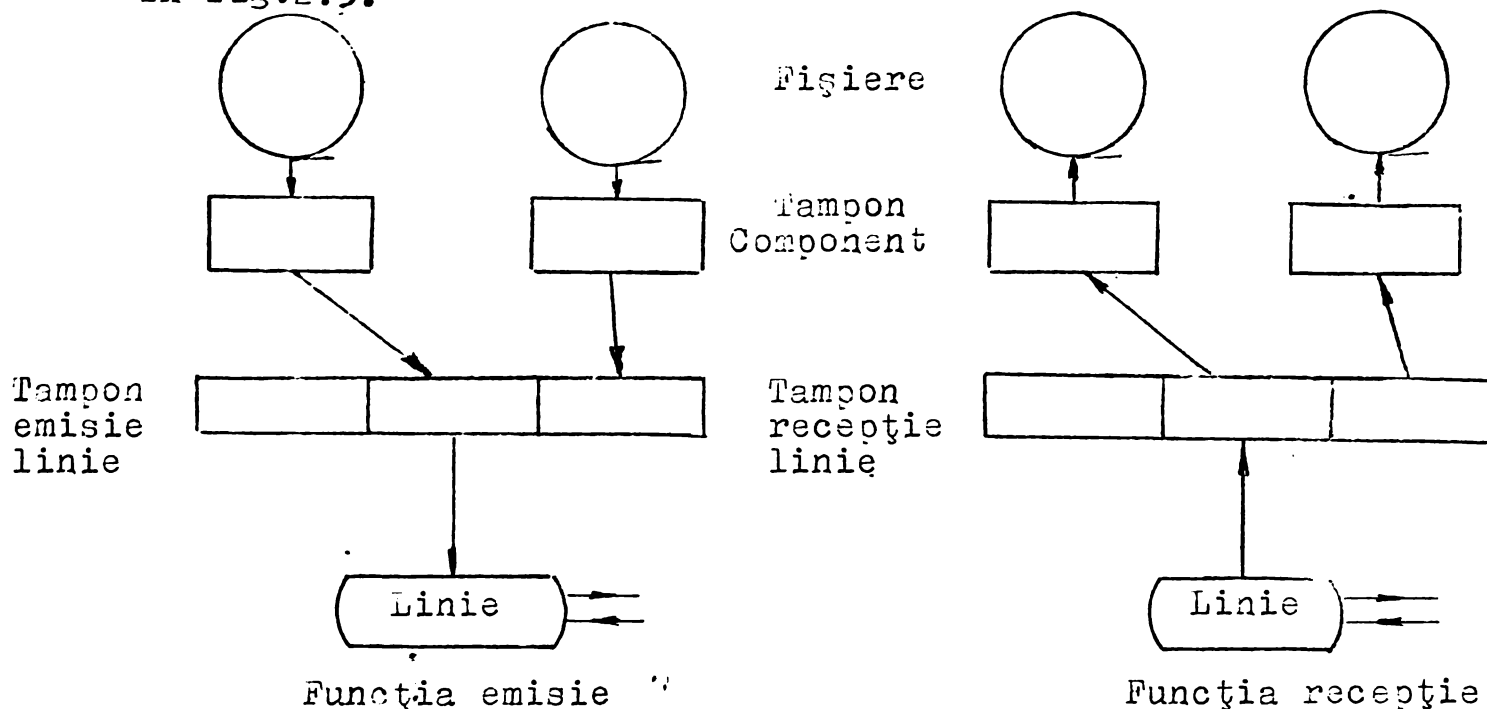


Fig.2.3. Organizarea generală a Sistemului de exploatare pentru legătura UC-UC.

Funcția Emisie se poate realiza prin folosirea următoarelor soluții:

- Alocarea efectivă la fiecare subcanal Sursă de date, a unui tampon component și rezervarea unui tampon de emisie în linie. Aceasta prezintă avantaje, deoarece realizează o gestiune simplă, prin faptul că se golesc tampoanele component, pentru a forma elemente în tamponul de emisie în linie și permite o citire mai ușoară, prin anticiparea articolului următor din fișier, pe care îl plasează în tamponul component. Ca inconvenient, se poate menționa că există o cerere mare de memorie, care este impusă de transferuri de lanțuri de caractere în unele tampoane, care sînt operații lungi.

- Emiterea, direct de la tampoanele component, prin suprimarea tamponului de emisie în linie, păstrînd un tampon component de intrare pentru un subcanal sursă. Aceasta prezintă avantaje deoarece necesită loc mai puțin în memorie și prezintă mai multă rapiditate, pentru că evită în parte, recopiarea tampoanelor. Ca inconvenient, apare dificultatea pentru zona comenzilor și subcanalul de comunicații între mașinile

de scris pupitru și deci este necesar a se crea un tampon auxiliar, în care se vor forma comenzile de transmis. Această soluție prezintă interes, dacă se reușește să se înlănțuiască informațiile cu comenzile subcanalului, pentru a le transmite în linie.

- Multiplexarea subcanalelor, în timp determinat, dacă nu este posibil a se realiza înlănțuirea de date pentru emisia în linie și crearea mai multor tampoane de emisie în linie, în care se vor citi articolele și care vor servi și ca tampoane component, fig.2.4.,

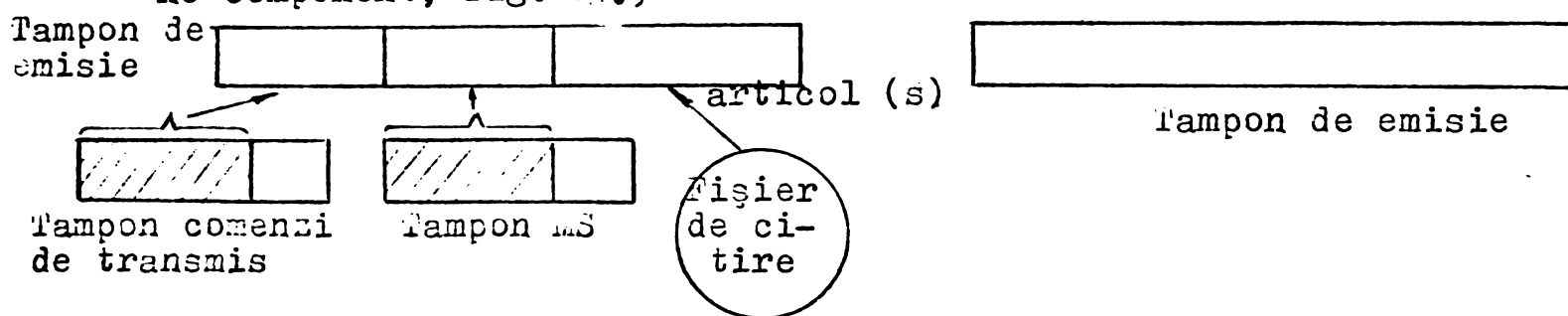
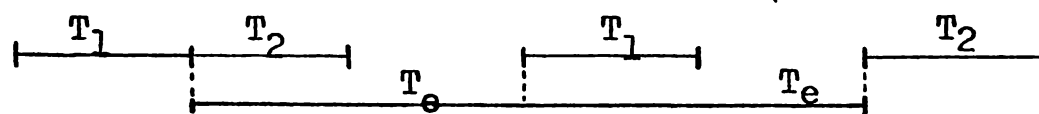
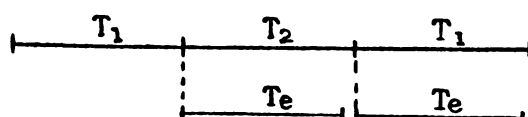


Fig.2.4. Tampoane emisie linie din care se citesc articolele.

se va putea limita la două tampoane de emisie care să lucreze în basculă. Totuși, acest mecanism prezintă dificultăți în ceea ce privește articolele de mărime superioară mărimei mesajului, deoarece nu va fi posibilă nici o citire nouă înainte de golirea completă a tamponului. Se atinge optimum cu acest sistem, dacă mărimea articolelor este inferioară lungimii mesajelor, când intervalul de timp care separă două emisii consecutive în linie, este inferior sau egal cu timpul de umplere a tamponului, fig.2.5.



a. - timp de emisie superior timpului de umplere;



b. - timp de emisie inferior timpului de umplere.

Fig.2.5. Folosirea a două tampoane de emisie.

Unde:

T1 - timp de umplere a tamponului 1;

T2 - timp de umplere a tamponului 2;

Te - timp care separă două emisii consecutive în linie.

Pentru funcția de Recepție există mai multe soluții pentru rezolvarea problemei:

- Multiplexarea în spațiu, adică prezența simultană în mesaj a mai multor elemente și dacă numărul și ordinea elementelor este necunoscută, atunci se poate adopta soluția de a avea un tampon de recepție din linie și atâtea tamponuri componente câte subcanale Receptor există, cum se arată în fig. 2.6. Avantajul principal al acestei soluții este acela că perifericele lente nu frânează perifericele rapide; această soluție e bine adaptată pentru procesele paralele.

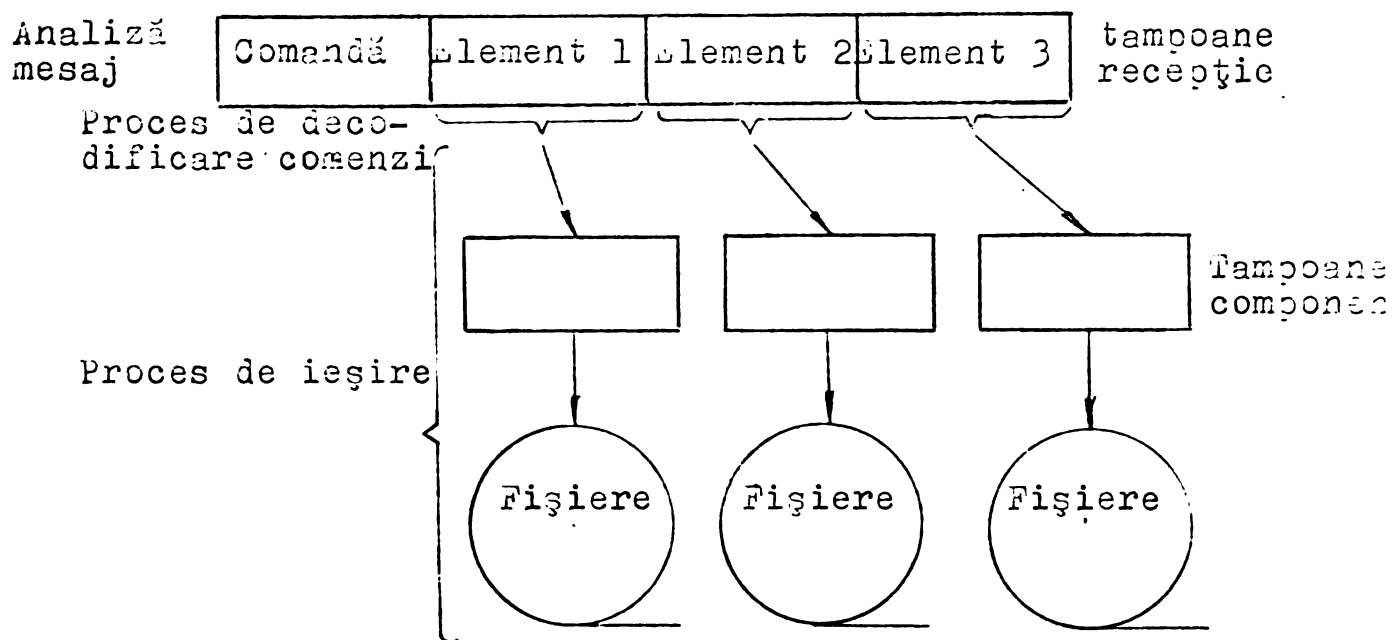


Fig.2.6. Procesul de ieșire care asigură transferul elementului în tamponul component și scrierea articolului în fișier.

446489  
194 G

După o analiză a mesajului primit, se vor activa procesele de decodificare - interpretare comenzi de ieșire. Fiecare proces de ieșire, asigură transferul elementului în tampon și dacă articolul nu este incomplet, scrierea sa în fișier. Inconvenientul soluției constă în cererea mare de memorie. Dacă numărul de tamponne este inferior numărului de subcanale Receptor, gestiunea va fi complicată prin faptul că recepția unui ACC - confirmare pozitivă pentru un subcanal dat, nu va autoriza transmiterea elementului următor.

- Multiplexarea în timp, necesită o zonă de comenzi, un element mașină de scris și un element dintr-un alt subcanal. Se poate suprima tamponul de recepție linie și se va primi mesajul direct în tamponnele componente. Principala dificultate în realizarea acestei soluții, rezultă din faptul că, articolele au trebuit să fie segmentate pentru transmisie; problema nu se pune dacă nu se consideră că fișierele au fost special create pentru transmisie și în care se limitează lungimea articolelor, în așa fel ca să fie inferioară lungimii mesajului de transmis în linie.

Aceasta, va conduce la efectuarea de recopieri a tamponnelor, din cauza zonei comenzilor și a mașinii de scris. Pentru fiecare tampon va trebui rezervată o lungime  $l$ , care să fie,

$$l = l_{\max} + l_{ce}, \text{unde :}$$

$l_{\max}$ , este lungimea maximă a unui articol;

$l_{ce}$ , este lungimea zonei comenzilor și a elementului subcanalului mașină de scris.

Nu se câștigă în memorie decât pînă la un anumit număr  $k$  de subcanale Receptor. Acest număr va putea fi determinat aproximativ de relația:

$$k \times l_{ce} = l_{\text{tampon linie}};$$

$$k = \frac{\text{lungime tampon linie}}{l_{ce}}$$

- Existența unui tampon de recepție linie și mai multe tamponne component dar mai puține în număr decât numărul de subcanale. Aceasta se motivează prin faptul că un singur

tampon component este utilizat la fiecare recepție , numai în cazul multiplexajului în timp. Tampoanele componente nu sînt legate într-un mod fix la componentele date, dar vor servi în basculă la diferitele subcanale. Dificultatea apare pentru articolele fragmentate, căci tamponul se găsește blocat atît timp cît articolul nu a fost reconstituit; trebuie în consecință, să se transmită părțile articolului, unele după altele, cu prioritate față de alte subcanale și se impune ca numărul subcanalelor Receptor cu articole fragmentate, să nu depășească numărul tampoanelor component.

- Existența unui singur tampon, utilizat atît pentru recepție cît și la ieșirea în fișier. Aceasta se justifică în cazul cînd timpul de transmisie va fi lung, comparat cu timpul de scriere în fișier.

După cum rezultă, această varietate de soluții conduce la alegerea unei organizări convenabile a sistemului.

Structura sistemului am conceput-o să fie de tip ierarhizat. Pentru această structură a fost necesar să se elaboreze interfețe între diferite părți hardware sau software din Sistemul de exploatare, ca urmare a aplicării unei proceduri de comunicație sau protocol, care se bazează pe principiul de comunicare cu confirmare.

Aceste interfețe permit existența unui set comun de semnale utilizate pentru legătura între mai multe dispozitive (componente periferice) sau sisteme și care traversează o frontieră identificabilă numai de aceste semnale; această situație este corespunzătoare funcționării Sursei și Receptorului de date, pentru a se asigura executarea schimbului reciproc de semnale.

În continuare sînt prezentate, pentru cazul nostru unele concepte privind interfețele standard.

Sistemul nostru din acest punct de vedere are o sarcină dificilă; realizarea teleconversiei de fișiere . schimbarea mediului care servește ca suport de fișier, la distanță.

Sistemul în final îl putem reprezenta ca o structură ierarhizată pe nivele, unde fiecare nivel corespunde la o etapă în parcursul informației. Are avantajul că extinderea sistemului este ușor și simplu de realizat și asigură schimburile de informații pe întreg ansamblul sistemului de la linie pînă la componente. Astfel, avem

- Nivelul liniei, care permite schimbul de blocuri conținînd comenzi și date;

- Nivelul stației, care prepară mesajele pentru nivelul liniei. la emisie , asigură transferul informațiilor din tamponul sau tampoanele component în tamponul de emisie, sau gestionează alege-

rea subcanalului al cărui element trebuie să facă parte din mesaj iar la recepție, prepară transferul tamponului recepție în tamponul component;

- Nivelul transport element, care asigură la emisie umplerea tamponelor componente, iar la recepție participă la golirea tamponelor componente.

- Nivelul dispozitiv, care asigură schimbul de informații între fișiere.

In fig.2.7. se arată forma pe care o ia informația de transmis, la fiecare interfață, precum și poziția Sistemului de gestiunea fișierelor - SGF - și Sistemul de gestiunea transmisiilor - SGT -.

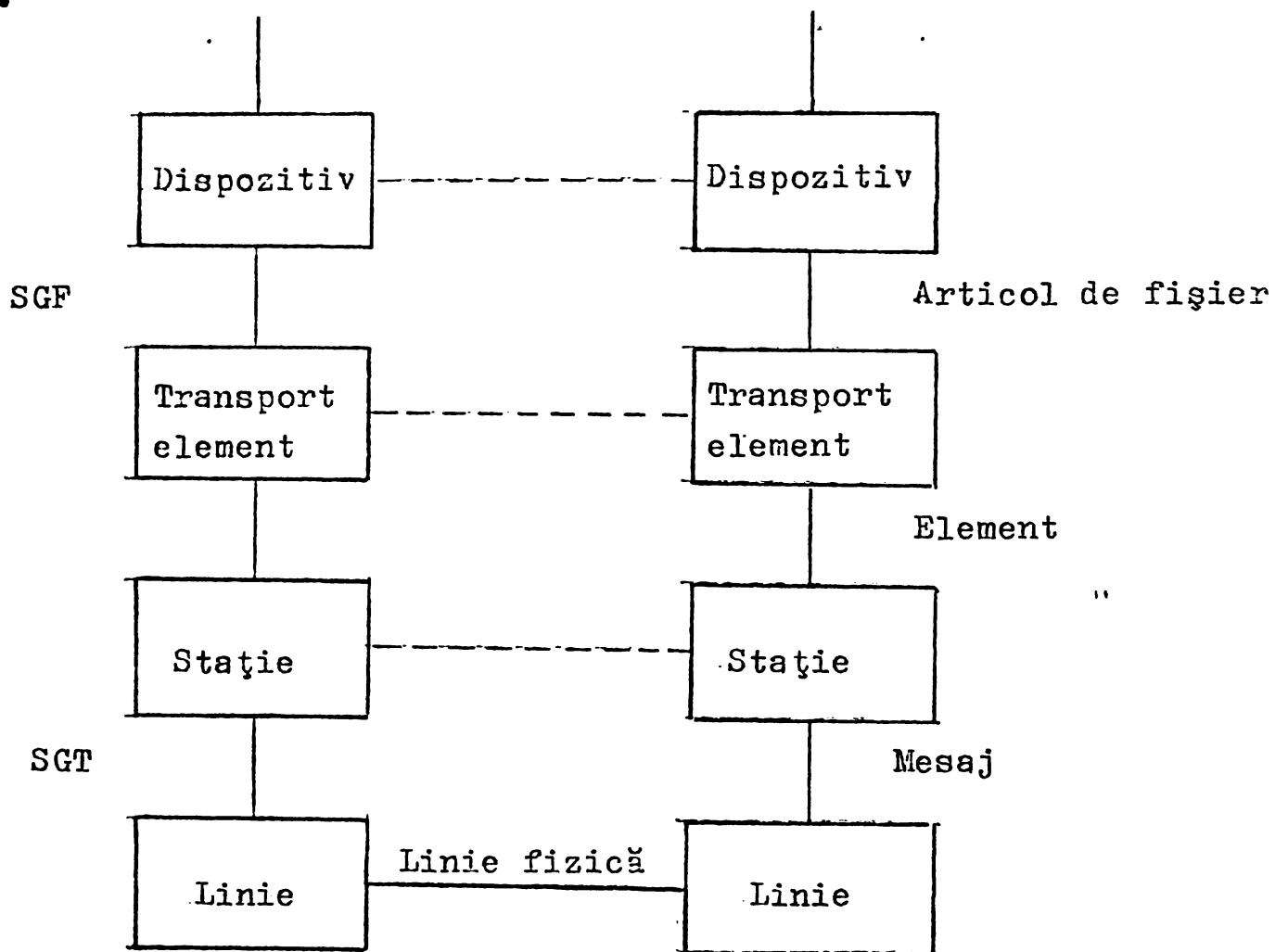


Fig.2.7. Informația de transmis prin interfețe

### 2.3. PROCEDURA DE EXPLICATARE PENTRU LEGATURA UC - UC

În această parte a lucrării am prezentat un studiu teoretic care cuprinde concepte proprii privind componenta de bază a sistemului nostru, Procedura de exploatare pentru legătura între calculatoare. Cercetările efectuate în acest scop, ne-au condus la stabilirea și elaborarea unei concepții arhitecturale a sistemului axată în mod deosebit pe structurarea și funcționarea proceselor care reprezintă funcțiile principale ale aplicației de teleprelucrare. Ca urmare, am elaborat modele deterministe care să reflecte cât mai corect procesele sistemului bazându-ne pe noțiunile moderne de procese și de funcționare a acestora în paralel.

Lucrările prezentate în continuare privind concepția teoretică de realizare a sistemului sînt originale și pot constitui contribuții personale importante în domeniul sistemelor de teleprelucrare a datelor datorită posibilităților de generalizare și caracterului lor metodologic.

Se poate afirma că lucrările de specialitate din acest domeniu referitoare la concepția de realizare a sistemelor, nu oferă soluții generale pe care le-am fi putut adopta cu ușurință ci numai soluții foarte particulare, empirice; de aceea realizările teoretice și practice sînt de o mare diversitate și nu sînt sistematizate. Studiile noastre pe această direcție au ținut cont de lucrările și realizările cele mai recente în domeniu, multe dintre ele avînd idei valoroase, pe care le-am valorificat într-un concept propriu mai sistematizat și cu grad mai mare de generalizare.

Lucrările mai importante care au constituit un îndrumar pentru noi, sînt cele referitoare la rețelele ARPA și CYCLADES și la sisteme ca ESOPE, CP - IBM și altele precum și la unele studii prezentate de autori ca Davies, privind structura rețelelor de calculatoare și J. Martin privind funcțiile și organizarea sistemelor de teleprelucrare. Alte lucrări importante consultate sînt prezentate în bibliografie (2).

Ca urmare, Procedura de exploatare am conceput-o ca să aibă caracteristicile sistemelor cu sarcini multiple, fiind astfel compusă dintr-un număr de procese paralele care reprezintă funcțiile ale sistemului și a căror cooperare conduce la realizarea funcției generale a aplicației de teleprelucrare a datelor.

Noțiunea de proces, în accepțiunea noastră și în general, în programarea de sistem, nu este decît o facilitate de concepție,



de organizare și de scriere; procesul este reprezentat de o entitate dinamică adică, de secvența în timp a tuturor acțiunilor elementare necesare pentru realizarea unei lucrări. Se caracterizează prin: algoritmi (reguli de calcul care descriu complet o lucrare), - date de intrare (elemente cu care operează procesul), - date de ieșire (rezultatele furnizate de proces), - resurse (mijloace materiale necesare pentru executarea procesului, resurse pasive, care conservă informațiile fără a le modifica ca de exemplu, memoriile sistemului și resurse active numite și procesoare, care pot executa algoritmi ca de exemplu unitățile de schimb, unitatea centrală).

Un proces se poate găsi pe o mașină virtuală în două stări: activ, când el este executat pe un procesor și blocat, când evoluția sa este oprită. Introducând resursele fizice ale sistemului, stările procesului vor fi: de execuție, când dispune de toate resursele fizice și logice, activat, când dispune de resurse logice dar fără procesor, în așteptare, când nu dispune de resurse fizice și dezactivat, când nu dispune de nicio resursă și este scos din tabelele sistemului. Utilizarea proceselor paralele la elaborarea sistemului nostru duce la o serie de avantaje ca, descompunerea aplicațiilor de teleprelucrare în funcțiuni independente, decuplarea funcțiilor, adaptarea proceselor cu ușurință la morfologia aplicației, integrare progresivă, siguranță în funcționare, performanțe ridicate în exploatare, programare ușoară și altele.

În concepția sistemului nostru, datorită complexității lui, utilizarea proceselor paralele constituie obiective obligatorii deoarece duc la optimizări în funcționare și la posibilități de generalizare și extindere. Procedura de exploatare, în această fază de concepție este elaborată ca să răspundă cerințelor unui sistem ipotetic, presupus a dispune de posibilitatea de a crea și de a utiliza procese paralele, posibilitatea de înlănțuire a informației pentru emisia în linie, alocarea de memorie efectuată după principiul paginei la cerere, posibilitatea de a crea și utiliza semafoare.

În această fază de concepție de realizare am definit Procedura de exploatare în ansamblu și pe componente, am determinat și elaborat modelele matematice de tip determinist care reprezintă funcțiile de bază ale aplicației de teleprelucrare și am utilizat unele concepte informatice privind procesele paralele. Cu alte cuvinte, am putea spune, că am studiat și elaborat arhitectura sistemului.

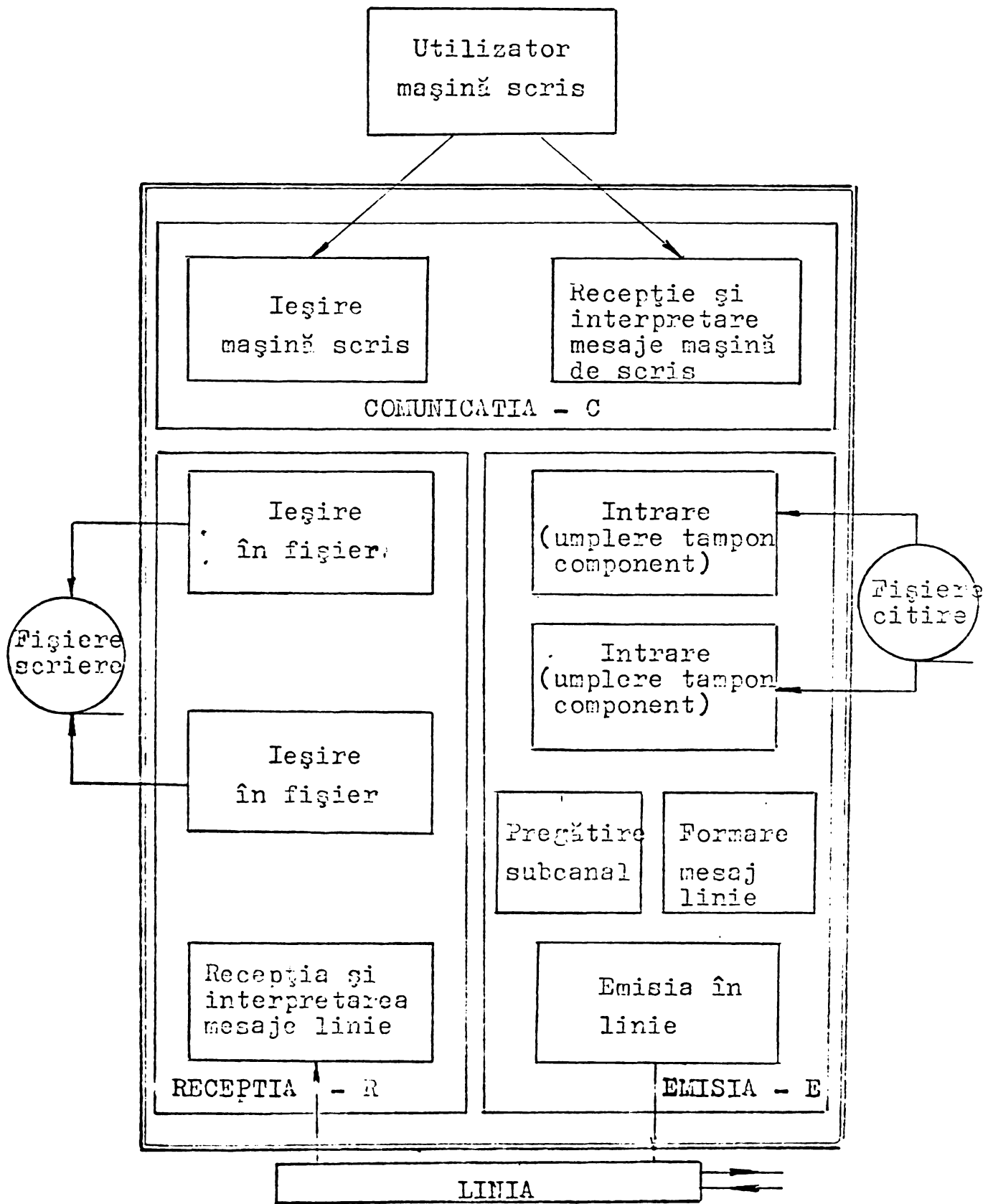


Fig.2.8. Structura Procedurii de exploatare pentru legătura UC - UC.

Procedura de exploatare dispune de trei componente: Comu-  
nicația cu utilizatorul, Emisie și Recepție, ca în fig 2.8.

### COMUNICATIA - C

Funcțiunea Comu-  
nicația - C trebuie să permită dialogul  
om-mașină sau comunicația cu Utilizatorul. Se poate realiza  
prin două procese; unul care se va ocupa de recepția și de  
interpretarea mesajelor introduse prin mașina de scris pu-  
pitru și altul care se va ocupa de ieșirea mesajelor la ma-  
șina de scris pupitru.

#### Procesul de ieșire pe mașina de scris

Mesajele care ies, respectiv adresele lor sînt ținute  
într-o tabelă, care va conține la început adresa mesajului  
inițial și se face cunoscut utilizatorului de existența aces-  
tor mesaje. Accesul la tabelă este împărțit între toate pro-  
cesele care doresc să emită un mesaj, excluderea mutuală  
fiind asigurată printr-un semafor "acces la adresă mesaj".

Acestă tabelă va avea o mărime limitată și pentru a se  
evita riscurile depășirilor se va introduce un semafor "nu-  
măr de mesaje".

Procesul este activat prin intermediul semaforului "me-  
saj de ieșire". Orice proces care are un mesaj de ieșire pe  
mașina de scris, va trebui să execute secvența următoare:

- P (număr de mesaje);
- P (acces adresă mesaje);
- .
- .
- .
- Inscrierea adresei în tabelă;
- V (acces adresă mesaj);
- V (mesaj de emiterie-ieșire).

Ordinea celor două operații P este esențială pentru a  
nu se risca blocarea atât a procesului care efectuează aces-  
tă secvență cât și a procesului de ieșire.

Algoritmul este prezentat în fig.2.9.

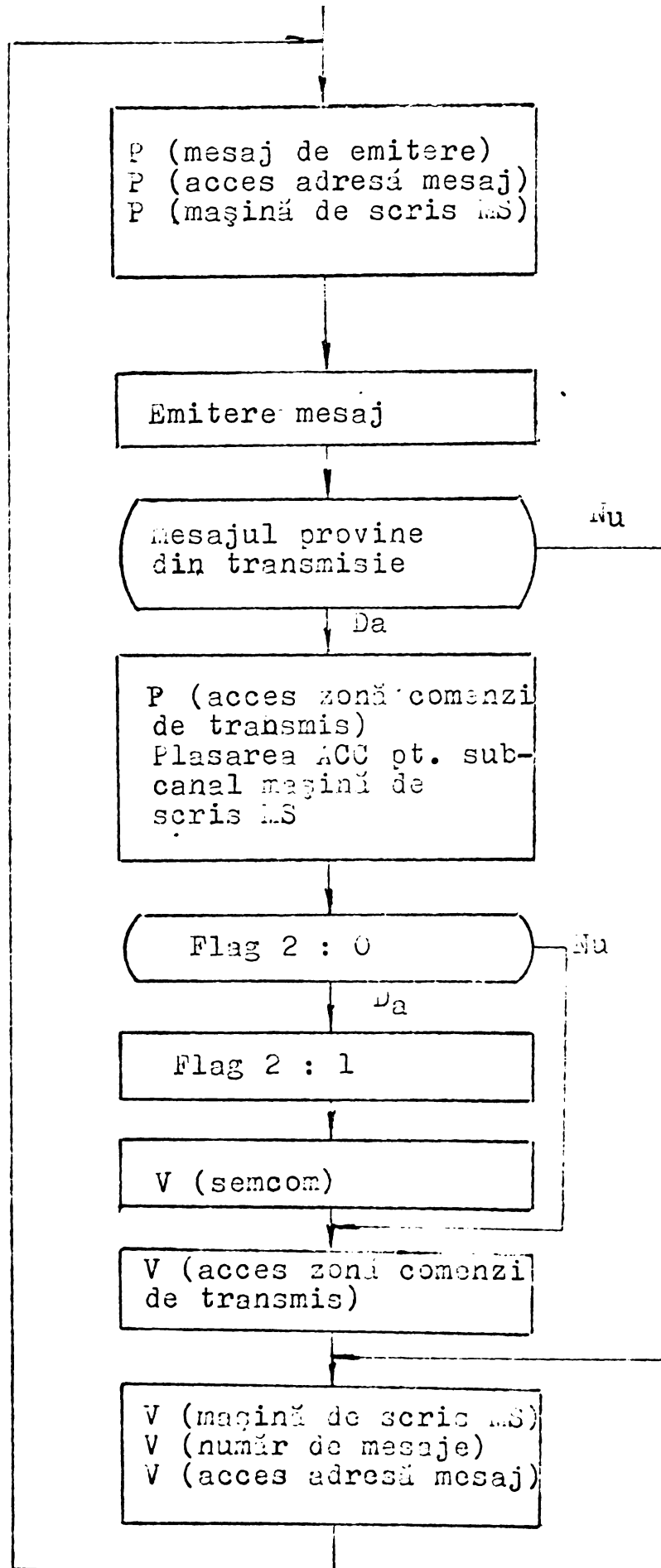


Fig.2.9. Algoritmul de ieșire la mașina de scris pupitru

Semaforul MS indică excluderea mutuală la accesul la mașina de scris.

Dacă mesajul care va ieși pe mașina de scris provine din linie, se prepară comanda de confirmare recepție - ACC, pentru a fi transmisă; aceasta va fi plasată într-o zonă specială numită "zona comenzilor de transmis" al cărui acces este protejat cu ajutorul unui semafor.

Drapelul "flag 2" este poziționat când se referă la prima comandă plasată în zonă; atunci o operație V este efectuată pe semaforul "semcom".

Deci, în rezumat, procesul utilizează tabela de adrese ale mesajelor de emiterie, zona comenzilor de transmis și șase semafoare cu valorile lor de inițializare; acces adresă mesaj: 1, mesaj de ieșire: 1, mașină de scris-MS:1, număr de mesaje: mărimea tabelii, acces zone de comenzi de transmis: 1, semcom: 0.

#### Procesul de recepție și interpretare a mesajelor de la mașina de scris - MS.

Mesajele de la mașina de scris pupitru - MS sînt primite într-un tampon special.

Primul caracter al mesajului permite să se cunoască dacă este comandă sau element de transmis. În acest ultim caz drapelul "flag 1" va fi poziționat, ceea ce va semnala procesului de formare a mesajului care va fi transmis în linie, prezența în tampon a unui element de la mașina de scris MS. Atît timp cît acest element nu va fi transmis în linie, procesul de recepție și interpretare mesaje de la mașina de scris va rămîne blocat datorită operației P pe semaforul de "așteptare".

Un semafor comun la mai multe procese "semcom" servește la activarea procesului de formare a mesajului de transmis în linie.

Excluderea mutuală a lecturii și scrierii pe mașina de scris pupitru este realizată prin semaforul "MS".

Se remarcă că așteptarea unui mesaj poate fi întreruptă de o ieșire de mesaj.

Dacă este o comandă validată, aceasta va fi analizată și executată de partea "interpretare comenzi a procesului.

Comenzile de transmis sînt formate într-o zonă specială a cărui acces este protejată prin semaforul de excludere mutuală "acces zonă comenzi de transmis". Procesul de recepție și interpretare mesaje de la mașina de scris poziționează drapelul "flag 1" și utilizează semaforul "semcom" pentru activarea procesului de formare mesaj pentru transmis în linie".

Algoritmul procesului de recepție și interpretare mesaje de la MS, în fig.2.10.

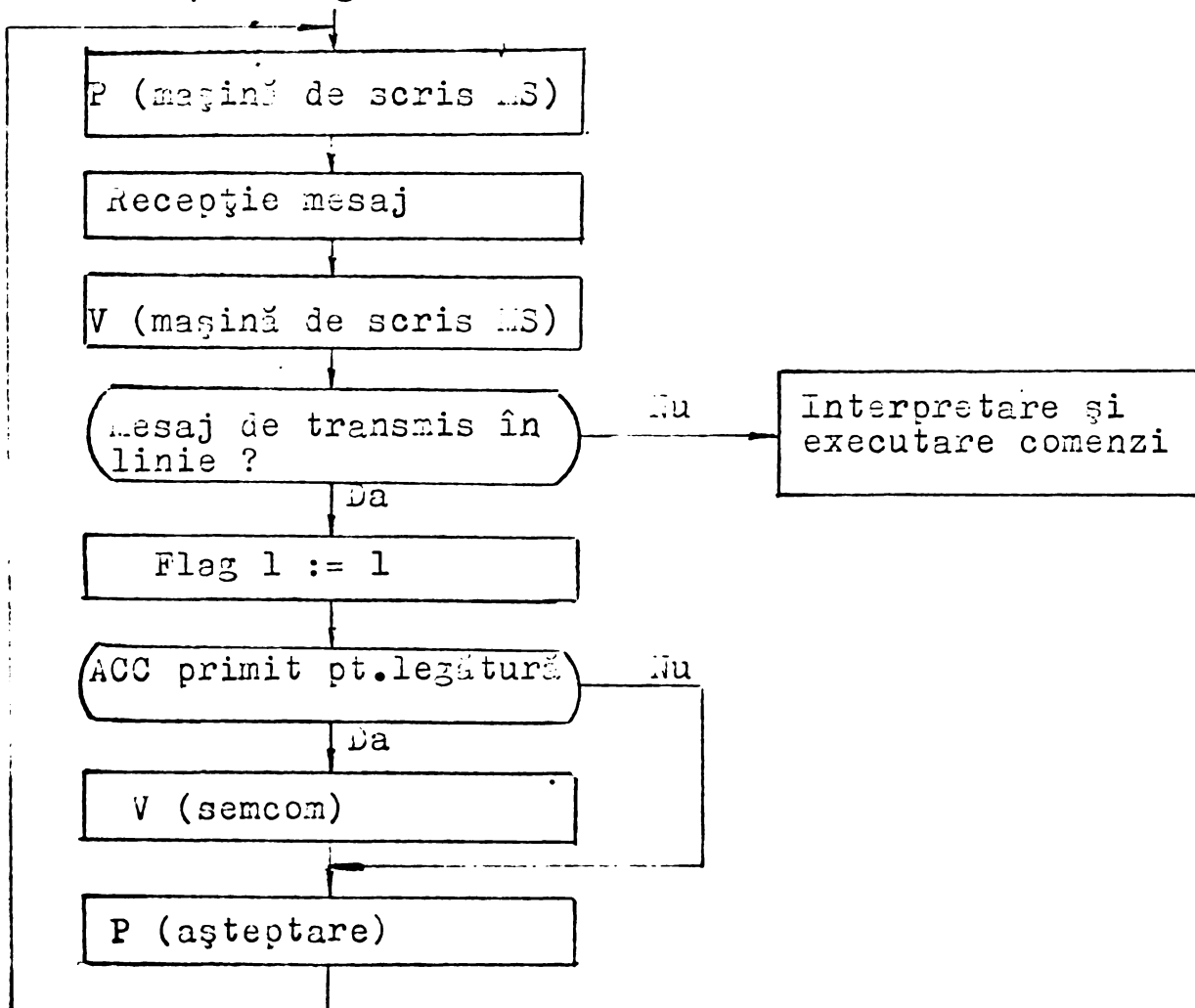


Fig.2.10. Algoritmul procesului de recepție și interpretare mesaje de la mașina de scris.

Comenzile introduse de la mașina de scris, se referă la stabilirea sau întreruperea unui subcanal - la reacțiile la anomalii și la oprirea generală a programului.

Informațiile privind subcanalele stabilite sau în curs de stabilire, vor fi ținute în tabele distincte dacă se referă la subcanale Sursă de date sau Receptor de date.

Pentru prima categorie, subcanale Sursă de date, se crează o Tabelă de Subcanale Active - TSCAS; excluderea mutuală a accesului la aceste tabele va fi asigurată printr-un semafor "acces TSCAS" inițializat la valoarea 1.

Fiecare intrare în TSCAS va fi formată din informații ca; număr de subcanale, numele și caracteristicile fișierului citit, numele terminalului care servește de suport fișierului, adresa de început a tamponului component, lungimea acestui tampon.

Se admite, că fiecare subcanal va dispune de tamponul său component, care va fi alocat la interpretările comenzii de stabilire subcanal, efectuată de către mașina de scris -MS.

Pentru a doua categorie subcanale Receptor de date se crează o Tabelă de Subcanale Active Receptoare-TSCAR, asemănătoare cu tabela TSCAS.

Semaforul "acces TSCAR" inițializat la valoarea 1, va asigura excluderea mutuală a accesului la această tabelă.

La definirea comenzii de stabilire a subcanalului-CER 2, se crează o tabelă de subcanale Receptor în curs de stabilire, care conține informații ca; număr provizoriu de subcanale, dispozitivul, rezervarea de memorie, număr subcanale stabilite de CER 1.

Dacă subcanalul a fost luat de CER 1 înainte de recepția CER 2, atunci legătura stabilită prin CER 2 va fi, prin trimiterea unei comenzi RPL, întreruptă.

În caz contrar un ACC va fi emis de Receptorul de date; Sursă de date nu generează ACC pentru legăturile cerute de CER 2, căci aceasta poate conduce la transmiterea de date pentru un subcanal pe care Receptorul îl va anula dacă comanda RPL nu parvine la timp.

### EMISIA - E

Funcțiunea Emisie - E este constituită dintr-un ansamblu de procese cooperante. Aceste procese realizează citirea datelor din fișiere, formarea elementelor și mesajului de transmis în linie și emisia mesajului în linie.

Se alege organizarea cu un număr de tamponare component egal cu numărul de subcanale Sursă de date existente; emisia în linie se realizează prin înlănțuirea zonelor de comenzi și de date - o proprietate importantă a acestui sistem.

#### Procesul de intrare - umplerea tamponului component.

Se consideră întâi procesele de intrare care asigură umplerea tamponelor component prin citirea din fișierele legate de subcanale corespunzătoare.

Algoritmul procesului de intrare, în fig.2.11.

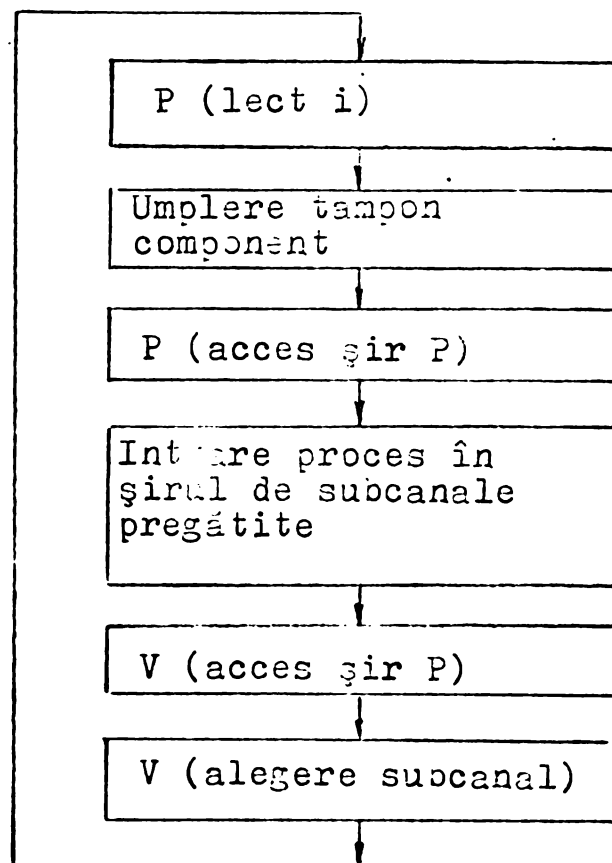


Fig.2.11. Algoritmul procesului de intrare - umplerea tamponului component.



Sincronizarea procesului cu alte procese se face cu ajutorul semaforului "lect i"; lectura articolelor din fișier antrenează o consultare a tabelii TSCAS, excluderea mutuală a accesului se realizează datorită semaforului "acces TSCAS".

Există o dificultate atunci când apar, în timpul citirii articolele de lungime variabilă; pentru a avea siguranța că nu se depășește tamponul, este necesar să nu se ceară citirea articolului următor, decât dacă locul disponibil în tamponul component este cel puțin egal cu lungimea maximă a articolului plus doi octeți.

Tamponul component fiind umplut, subcanalul este considerat pregătit și va fi plasat în șirul de așteptare a subcanalelor pregătite "șirul P"; accesul este protejat prin intermediul semaforului "acces șir P". Efectuându-se o operație V pe semaforul "alegere subcanal" se avertizează procesul care are sarcina să aleagă subcanalul, al cărui element va face parte din mesajul următor.

Semafoarele și valorile de inițializare utilizate de proces sînt; lect i : 0, acces șir P : 1, acces TSCAS : 1, alegere subcanal : 0.

#### Pregătirea subcanale Sursă de date - Sirul P

Orice subcanal pregătit, adică al cărui element se găsește în tamponul component, cu excepția subcanalului de comunicație între mașinile de scris pupitru, intră în șirul de așteptare a subcanalelor pregătite - Sirul P; intrarea în șir se face în ordinea șirurilor de așteptare.

Ieșirea din șir a primului element se realizează prin parcurgerea de la început a șirului, alegînd primul subcanal care a primit sau a generat o comandă de confirmare recepție pozitivă - ACC, în fig.2.12.

Subcanalul astfel ales părăsește Sirul P, afară de cazul unui articol incomplet pentru transmisie, când subcanalul rămîne în continuare în Sirul P.

Un proces special de gestiune a Sirului P reprezentat în fig.2.13. realizează algoritmul de ieşire.

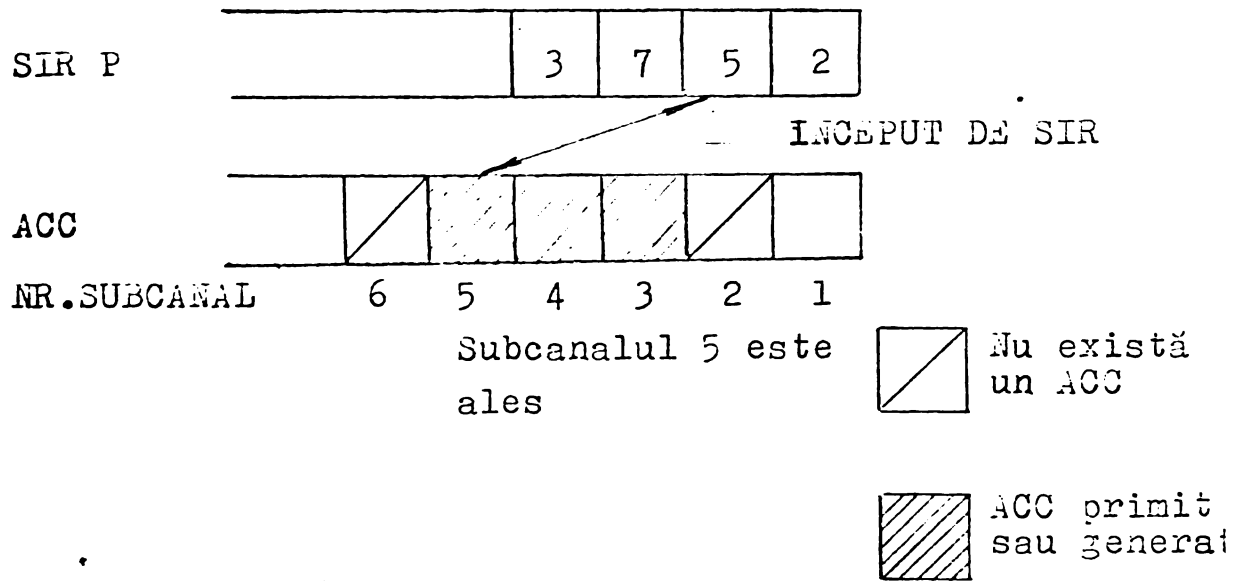


Fig.2.12 Ieşire din Sirul P

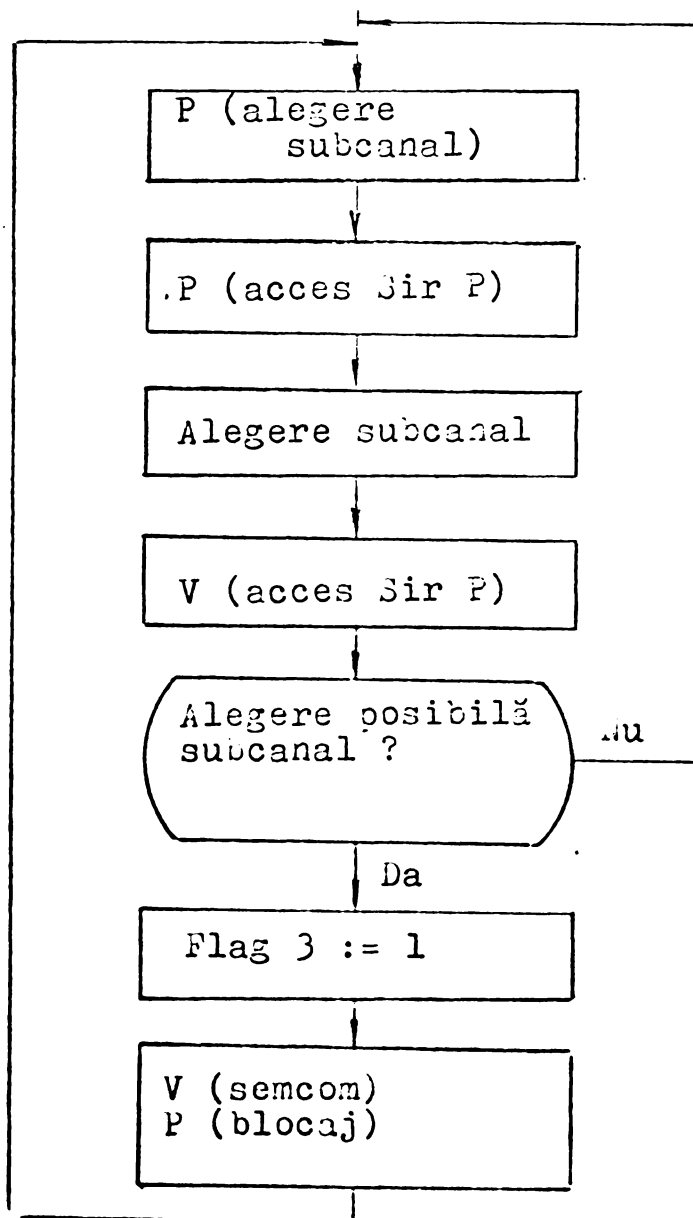


Fig. 2.13. Procesul de gestiune a Sirului P.

Procesul nu alege decît un singur subcanal și aranjează într-o tabelă specială informațiile necesare.

Dacă lungimea elementului nu impune o fragmentare a articolului, subcanalul este luat din Sirul P; în toate cazurile ACC corespunzător este șters.

Se poziționează drapelul "flag 3" și se efectuează o operație V pe semaforul "semcom" pentru activarea procesului de formare a mesajului de transmis în linie care va trebui să efectueze o operație V pe semaforul "blocaj".

Semafoarele utilizate și valorile lor de inițializare sînt; alegerea subcanalului: 0, semcom: 0, blocaj: 0, acces șir P: 1.

#### Procesul de formare a mesajului linie

Procesul de formare a mesajului linie începe prin așteptarea sosirii cel puțin a unui eveniment din cele trei care urmează:

- comenzi de transmis;
- un element pentru subcanalul mașină de scris - MS și un ACC primit pentru acest subcanal;
- un element pentru un alt subcanal care a fost ales prin procesul de gestiune a Sirului P.

Această așteptare este realizată prin semaforul comun "semcom".

Procesul fiind activat, acesta testează drapelul "flag 2" care îi permite să știe dacă sînt comenzi de transmis. Dacă este cazul, se prepară datele de înlănțuire pentru a le transmite în linie.

Punînd drapelul "flag 4" pe 1, se marchează că va trebui efectuată o operație V pe un semafor "acces zonă comenzi de transmisie".

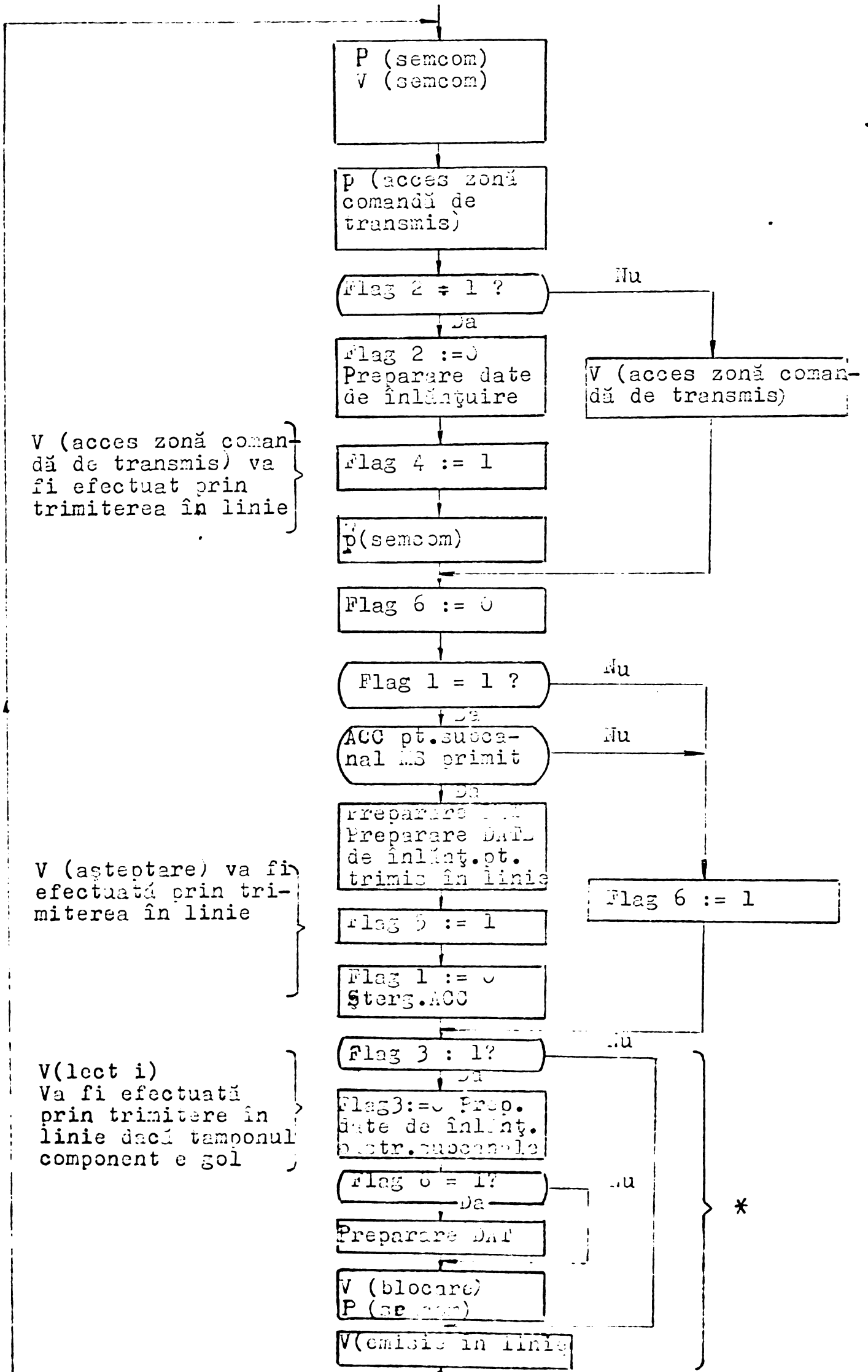


FIG.2.14 Algoritmul procesului de formare mesaj de transmis.

Procesul continuă, testînd valoarea drapelului "flag 1"; cînd un element pentru subcanalul de comunicație între mașinile de scris pupitru așteaptă în tamponul său și o comandă de confirmare recepție pozitivă - ACC a fost primită pentru acest subcanal, se înlănțuiește acest element care va fi precedat de identificatorul zonei de date (DAT) pe care se poate plasa la urma comenzilor de transmis, ceea ce va fi marcat prin poziționarea drapelului "flag 6".

Comanda de confirmare ACC pentru subcanalul mașină de scris MS este ștearsă, "flag 1" este pus la 0; valoarea 1 a drapelului "flag 5" va permite, ca procesul de emisie în linie, să știe că o operație V trebuie să fie făcută pe semaforul "așteptare".

Se verifică valoarea "flag 3"; dacă este pe 1, se înlănțuiește elementul subcanalului ales de procesul de gestiune a Sirului P, care este deblocat datorită operației V pe semaforul "blocaj".

Procesul de emisie a mesajului în linie este activat prin intermediul semaforului "emisie linie".

Dacă se dorește să se realizeze o multiplexare în spațiu - mai multe subcanale în același mesaj, se repetă secvența marcată cu asterix din fig. 2.14.

Semafoarele utilizate și valorile de inițializare sînt: semcom : 0, acces zonă comenzi de transmis : 1, blocaj : 0, emisie linie : 0.

Algoritmul procesului este arătat în fig. 2.14.

#### Procesul de emisie a mesajului în linie

Datele de înlănțuire pentru subcanal, au fost pregătite de procesul de formare a mesajului.

Este suficient pentru a efectua emiterea, de a autoriza accesul la zona de comenzi de transmis și de a activa procesul de recepție - MS și procesul de umplere a tamponului component.

Algoritmul procesului de emisie în linie în fig.2.15.

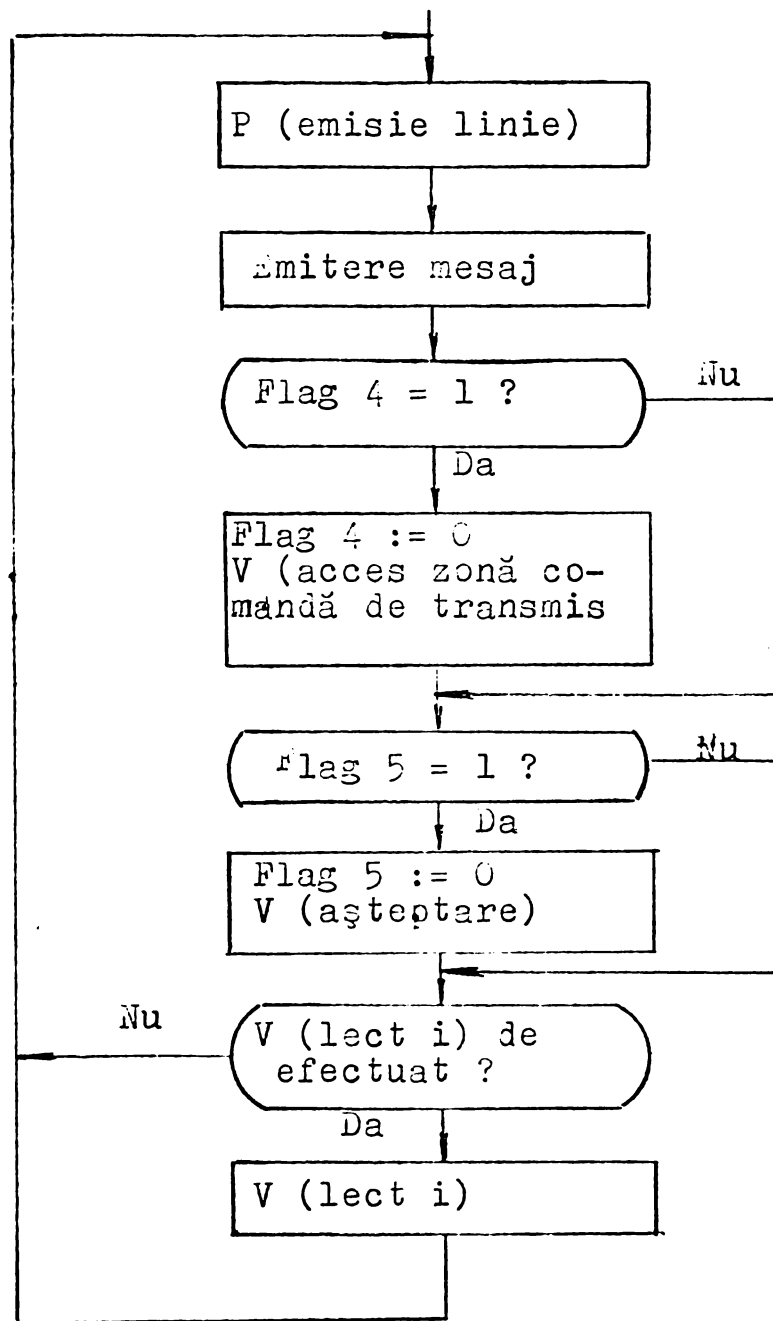


Fig.2.15. Algoritmul procesului de emisie în linie.

Semafoarele utilizate cu valorile lor de inițializare sînt; emisie linie : 0, acces zonă comenzi de transmis : 1, așteptare : 0, lect i : 0.

## RECEPȚIA - R

Funcția Recepție - R este concepută ca un ansamblu de procese cooperante care asigură recepția și analiza mesajelor din linie, interpretarea și executarea comenzilor existente în zona de comenzi, dacă acestea sînt prezente, reconstituirea articolelor în tamponanele component cu elemente prezentate în zona de date a mesajului și scrierea acestor date în fișierele corespundente subcanalelor respective.

Se alege organizarea cu un tampon de recepție linie pentru mesajele transmise și un tampon component pentru un subcanal Recepție.

Subcanalul de comunicație între mașinile de scris pupitru va fi tratat într-un mod analog cu celelalte subcanale pentru transmisia în linie.

### Procesele de recepția și de interpretare a mesajelor din linie.

Procesul de recepție și interpretare a mesajelor din linie, are algoritmul reprezentat în fig.2.16.

Se efectuează întîi un polling pe linie, adică se efectuează o punere în așteptarea unui mesaj emis de calculatorul corespondent.

Mesajul fiind primit, se efectuează o analiză care are ca scop de a determina zonele prezentate și în interiorul zonei de date - dacă acesta face parte din mesaj, de a delimita elementele, pentru a comunica astfel adresele lor diferitelor procese de ieșire reactivitate de operația V pe semaforul "proces i".

Procesele de ieșire vor transfera, în primul rînd elementele în tamponanele lor component; sfîrșitul acestui transfer este așteptat de o serie de operații P al cărui număr este egal cu numărul proceselor de ieșire activate.

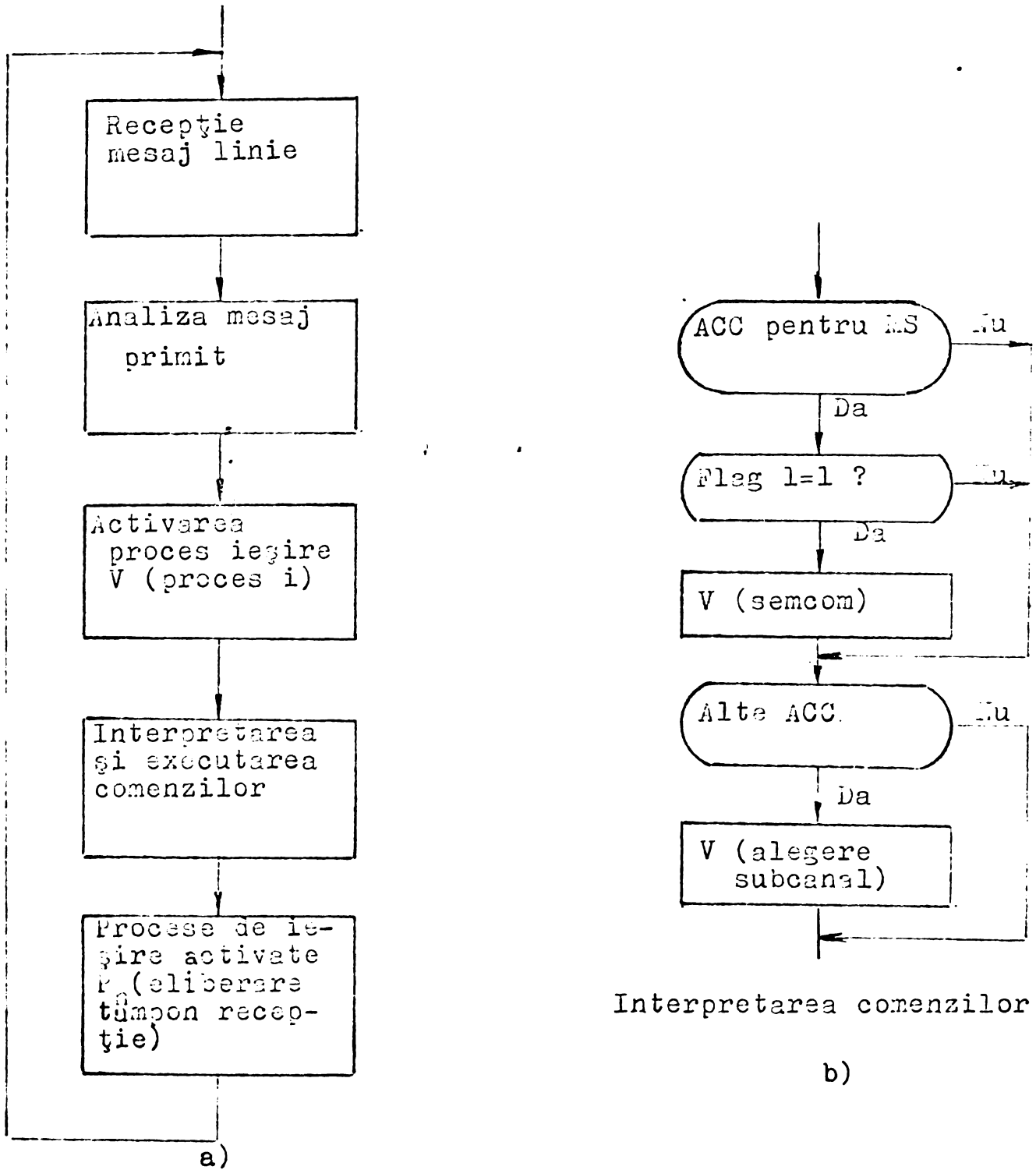


Fig. 2.16. Algoritmul procesului de recepție a mesajelor de linie și de interpretare comenzi.

a. recepție mesaje

b. interpretare comenzi.



În acest caz pentru a facilita rezolvarea, se introduce o operație pe un semafor, numită  $P_n(s)$ ,  $n \geq 1$ , întreg, care este echivalentă cu o suită de  $n$  operații  $P$  pe  $s$ . Se poate defini într-un mod analog operația  $V_n(s)$ .

Interpretarea și executarea comenzilor din zona de comenzi antrenează consultarea și modificarea tabelelor TSCAS și TSCAR; excluderea mutuală a proceselor care lucrează cu aceste tabele este asigurată de semafoarele "acces TSCAS" și "acces TSCAR".

Dacă comenzile de confirmare recepție pozitivă sau negativă ACC și NACC sînt în zona de comenzi, acestea se plasează într-un tabel special. Comenzile pe care procesul va trebui să le transmită în linie, vor trebui formate în zona comenzilor de transmis al cărui acces este protejat prin semaforul "acces la zonă comenzi de transmis". Se poziționează drapelul "flag 2" dacă era la 0 și se efectuează operația  $V$  pe "semcom".

Cînd operația  $P_n$  pe semaforul "eliberare tampon recepție" a putut fi terminată, este sigur că tamponul de recepție linie poate fi șters și recepția unui nou mesaj este astfel posibilă.

În rezumat, semafoarele utilizate cu valorile lor de inițializare sînt; eliberare tampon recepție : 0, proces i : 0, semcom : 0, acces TSCAS : 1, acces TSCAR : 1, acces zonă comenzi de transmis : 1, acces tabelă subcanale în curs de stabilire : 1, alegere subcanal : 0.

#### Procesul de ieșire pe fișier

Procesele de ieșire sînt sincronizate cu cele de recepție și interpretarea mesajului în linie prin intermediul unui semafor "proces".

Se transmite proceselor de ieşire pe fişier, la activare, adresa de început a elementului din tamponul de recepţie.

În fig.2.17. se arată forma unui element destinat unui subcanal, care nu se referă la elementul de comunicaţie între maşinile de scris - MS.

Transferul elementului din tamponul de linie, în tamponul component depinde de octetul structură a cărui valori posibile sînt; articol complet, mai multe articole complete, început articol, parte articol, sfîrşit articol.

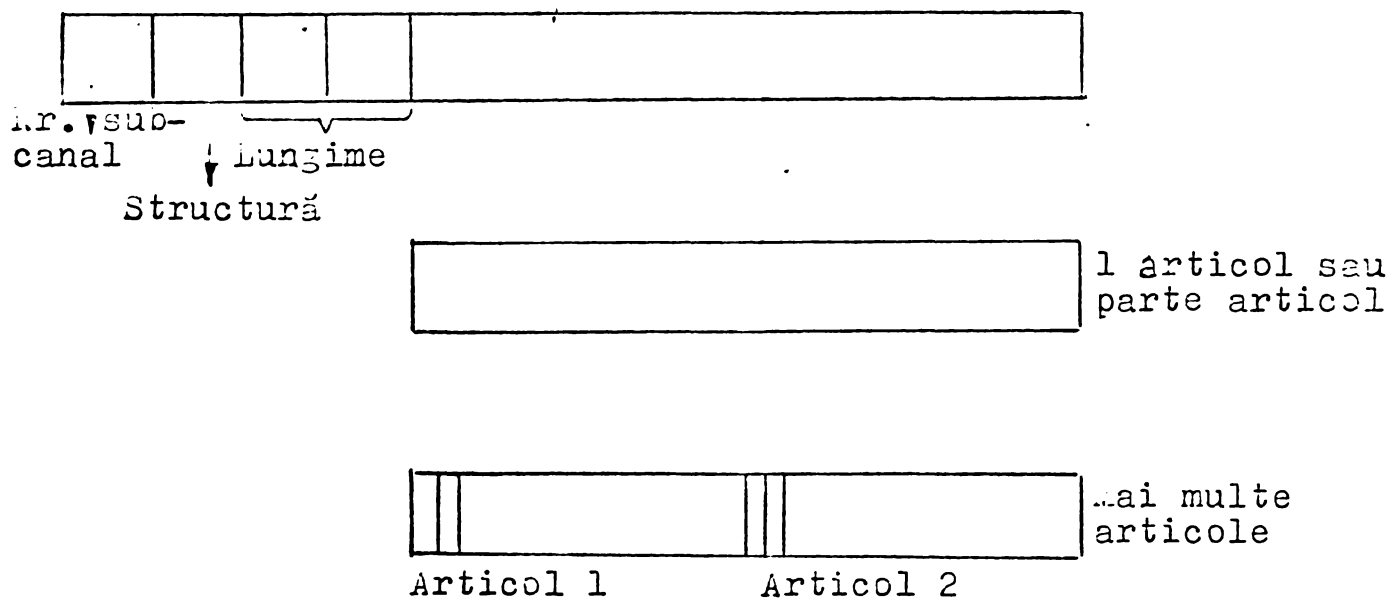


Fig.2.17 forma unui element destinat unui subcanal.

În primele trei cazuri se transferă tot elementul, iar cînd este o parte sau sfîrşit de articol, se recopiază partea de articol plasînd-o în tamponul component ca în fig.2.18.

Procesele de ieşire trebuie să ţină în evidenţă adresa curentă în tampon; aceasta poate fi consemnată în tabela TSCAR.

Cînd articolul este complet se goleşte tamponul printr-o scriere în fişierul legat la subcanal.

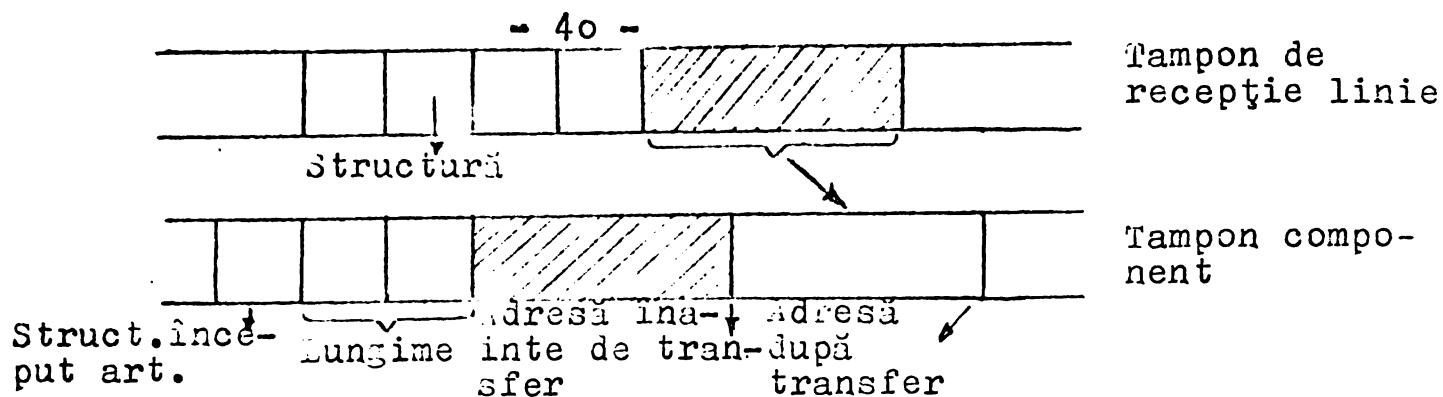


Fig. 2.18 Parte articol tampon recepție, în tampon component.  
 Algoritmul procesului de ieșire se arată în fig. 2.26

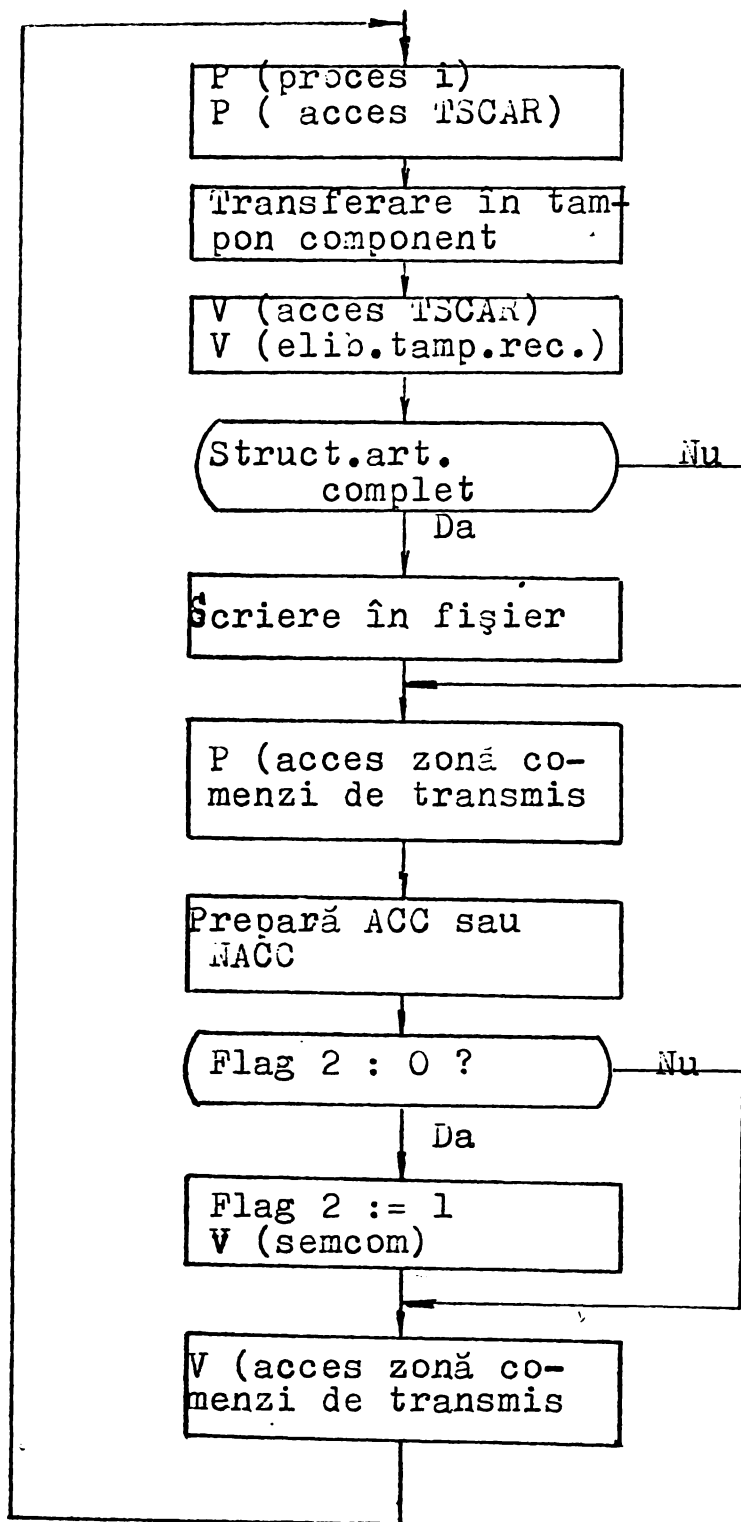


Fig. 2.18 Algoritmul procesului de ieșire

Semafoarele utilizate de proces cu valorile lor de inițializare sînt; proces 1 : 0, tampon liber recepție : 0, semcom : 0, acces TCAR : 1, acces zonă comenzi : 1,

Drapelul "flag 2" este poziționat și semnalizează prezența comenzilor în zona comenzilor de transmis.

Există un proces de ieșire mai particular, acela legat de subcanalul de comunicație între mașinile de scris.

Singurele sale funcțiuni constau în transferarea elementului începînd de la octetul lungime în fig.2.19, în tamponul de ieșire mașină de scris - MS și plasarea adresei acestui tampon în tabela procesului de ieșire pe mașina de scris - MS.

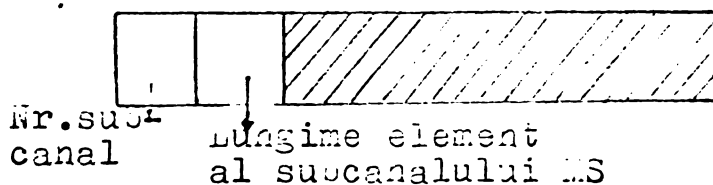


Fig.2.19 Transferul elementului în tamponul de ieșire mașină de scris -MS.

Procesele care doresc ieșirea unui mesaj la mașina de scris MS locală, trebuie să se sincronizeze cu ajutorul semafoarelor "număr de mesaje" și "acces adresă mesaj".

Algoritmul procesului de ieșire a subcanalului MS este arătat în fig. 2.20.

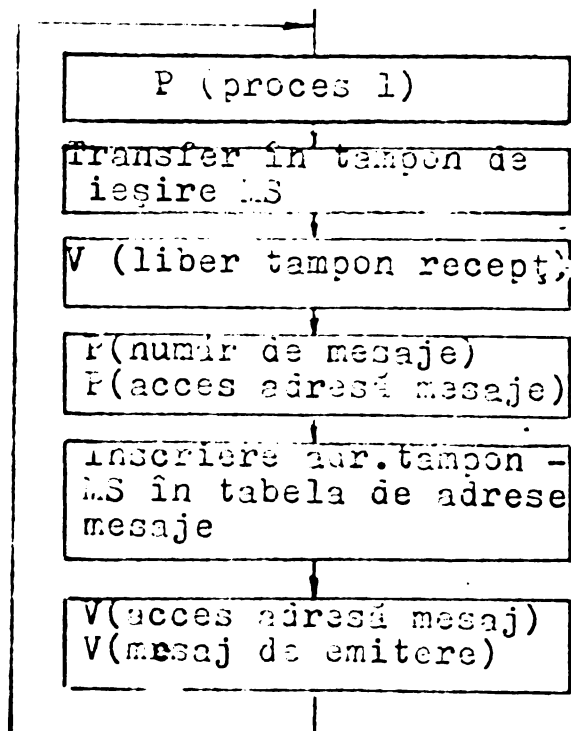


Fig. 2.20. Algoritmul procesului de ieșire a subcanalului MS.

## 2.4. PROCEDURA DE COMANDA

Elaborarea unei alte componente de bază a sistemului nostru, cea care asigură legătura între calculatoare și schimburile de informații între acestea, a constituit obiectul unor studii și cercetări pe care le-am efectuat în acest sens și care sînt prezentate în continuare ca o concepție teoretică proprie.

Sistemul nostru pentru a-și asigura funcționarea în bune condiții dispune de reguli sau protocoale, legate strîns de noțiunea de comandă și supraveghere.

Procedura de comandă, cum a fost denumită în lucrare cuprinde deci, comenzile necesare pentru stabilirea legăturii între calculatoare aflate la distanță și pentru asigurarea schimburilor sau transferurilor de informații. După cum am arătat, legătura fizică și logică între calculatoare realizată prin sistemul nostru asigură și interacțiunea acestora în scopul coordonării modului de transmitere și de recepționare a mesajelor în ambele sensuri și coordonează modul în care un sistem transmite iar celălalt recepționează mesajele. Fiecare dintre calculatoare va trebui să cunoască dacă celălalt este pregătit pentru a coopera la transferul de informații și să ia măsuri de eliminarea eșecurilor și erorilor de transmisie, fără timp lung de așteptare.

În cadrul sistemului nostru am considerat necesară existența unor modalități de comandă privind stabilirea legăturii și a schimburilor de informații între calculatoare care să dispună de o legătură inversă de reacție (după transmiterea datelor să fie cunoscut modul de recepționare și efectul rezultat).

Am ales conceptul de "transmiterea cu confirmare" care dispune de legătură inversă, avînd astfel posibilitatea de a realiza transferul controlat al informațiilor, una din părți confirmînd efectuarea transferului unei unități de informații (mesaj sau pachet). Acest concept a fost adaptat cerințelor sistemului nostru și a rezultat o serie de comenzi necesare care formează Procedura de comandă.

Astfel, în principiu, acest mecanism funcționează după cum urmează: la acceptarea unui mesaj de către Receptorul de date, se înapoiază Sursei de date o confirmare pozitivă; dacă nu se confirmă că mesajul a fost recepționat corect (se presupune că mesajul sau nu a fost primit, sau s-a pierdut confirmarea, sau calculatorul receptor este ocupat), se va proceda la retransmiterea mesajului respectiv.

Modulul de "transmiterea cu confirmare" reprezintă o inter-blocare a calculatoarelor,cel care a transmis așteptând confirmarea celuilalt;de aceea se mai numește și "transmitereși așteptare".

În fig.2.21. este reprezentată Procedura de comandă cu transmiterea cu confirmare a informației în cadrul legăturii între calculatoare,în care sînt redate stările,la Sursa de date și la Receptorul de date;semnalele care trec de la Sursă la Receptor și invers sînt necesare unei funcționării corecte pentru ca un singur element (mesaj) să fie transferat la fiecare ciclu complet,în care sînt incluse și pauzele pentru așteptarea semnalelor de la cele două calculatoare.

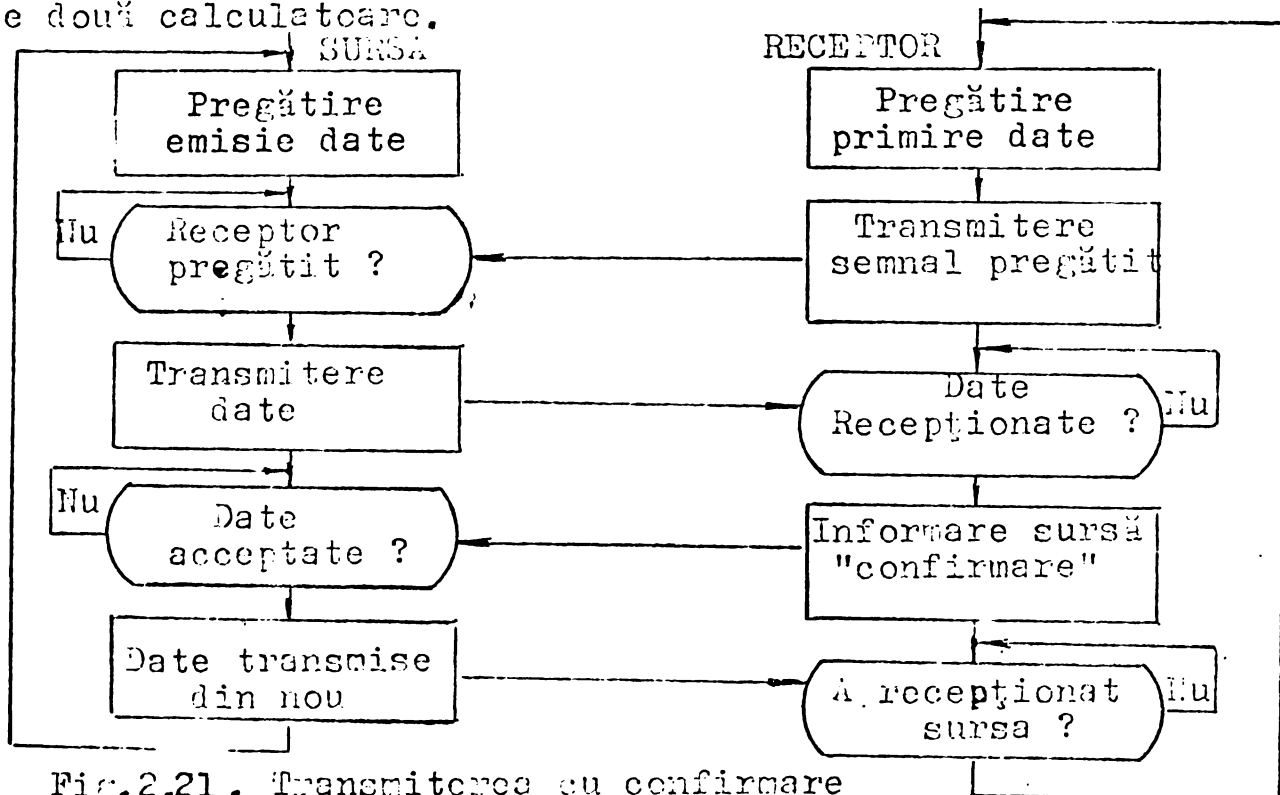


Fig.2.21. Transmiterea cu confirmare

Cercetările pe care le-am efectuat în această direcție pentru elaborarea unei concepții privind comenzile sistemului și prezentate în lucrare,au un caracter original și pot constitui contribuții proprii în domeniul prelucrării datelor în special în ceea ce privește protocoalele de teleprelucrare.

Unele lucrări de specialitate teoretice și practice abordează problema protocoalelor de teleprelucrare;acestea au totuși unele insuficiențe datorită caracterului lor particular,fiecare pe un specific dat. Ideile mai valoroase din aceste lucrări am încercat să le valorificăm în lucrarea noastră prin realizarea unui concept propriu de protocol,evident cu modificările corespunzătoare. Si aici,de un real folos au fost lucrările privind rețelele ARPA și CYCLADES precum și altele pe care le-am menționat în bibliografie (2).

Rezultă necesitatea ca sistemul nostru să dispună de protocoale, pentru stabilirea și întreruperea legăturii pentru asigurarea schimburilor pe linie și pentru alocarea de memorie.

Sînt definite, în continuare comenzile Sistemului de exploatare pentru legătura între calculatoare, prezentîndu-se principiile de bază privind realizarea acestora.

Comenzile Sistemului au fost grupate în următoarele părți componente:

- Comenzi de stabilirea legăturii UC-UC;
- Comenzi de schimburi de date în linie;
- Comenzi de suspendarea schimburilor în linie.

#### COMENZILE DE STABILIREA LEGATURII UC-UC

Introducerea unei cereri de stabilire a unei legături UC-UC, respectiv a unui subcanal de la mașina de scris de la unul din cele două calculatoare, implică satisfacerea unor condiții reunite.

Se verifică mai întîi dacă s-a atins numărul maxim de subcanale, un răspuns pozitiv la condiția de mai sus, oferă mai multe soluții;

- se retrimite înapoi cererea de legătură, avertizîndu-se utilizatorul printr-un mesaj la mașina de scris pupitru, utilizatorul putînd apoi reîncepe după un timp oarecare;
- se plasează automat cererea respectivă într-un șir de așteptare și se va căuta să fie satisfăcută în momentul cînd un subcanal va deveni disponibil,
- se avertizează utilizatorul de imposibilitatea satisfacerii imediate a cererii sale și să lasă acestuia la alegere, între a renunța sau ca cererea să intre într-un șir de așteptare.

În cazul că numărul de subcanale nu a atins maximum, se verifică dacă există disponibil terminalul; fișierul respectiv și memoria pentru a crea diferite tamponare necesare unui subcanal.

Această ultimă condiție poate fi implicit satisfăcută, când numărul maxim de subcanale nu este depășit și depinde de modul cum se administrează memoria.

Dacă una din aceste condiții lipsește din cauza unei indisponibilități pasagere, se poate lăsa decizia utilizatorului.

Apar astfel, ca necesare comenzi pe care cele două calculatoare vor trebui să le schimbe pentru a se pune de acord și a se avertiza reciproc în ceea ce privește condițiile de funcționare.

Aceste comenzi se referă la stabilirea subcanalului; se disting în funcție de poziția unde se găsește solicitantul, la viitoarea "Sursă de date" sau la viitorul "Receptor de date".

Cereri de stabilirea legăturii CER1 și respectiv CER2, răspunsuri de acceptare cereri, ACER1 și respectiv ACER2 și răspunsuri de neacceptare cereri, NACER1 și respectiv NACER2, sînt prezentate în continuare.

Comanda CER1. Această comandă, CERere de stabilire legătură CER1, este emisă de către partea care va deveni, dacă cererea este satisfăcută "Sursă de date".

CER1 conține informațiile necesare pentru stabilirea subcanalului ca în fig.2.22. fiecare bloc reprezintă un octet.

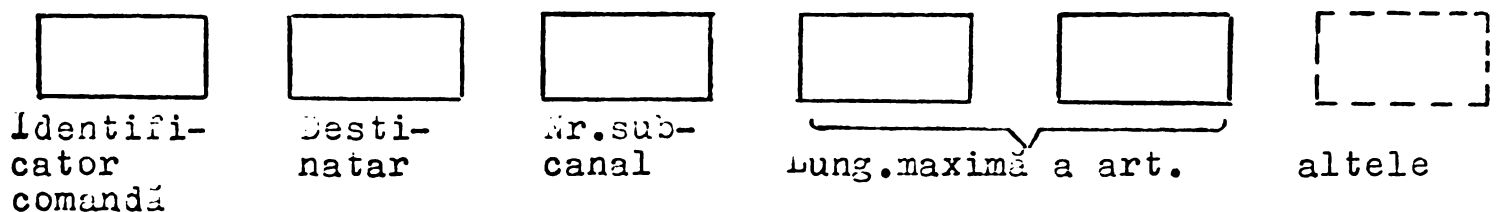


Fig.2.22.Compunerea comenzii CER1

Blocul destinatar determină terminalul - device.care va servi ca suport de fișier pentru Receptorul de date.

Sursa de date potențială asigură un număr pentru legătură, iar prin acest număr cele două părți vor identifica subcanalul.



Se comunică la viitorul Receptor de date lungimea maximă a articolului și alte informații cum ar fi tipul fișierului de creat și altele.

Uneori vor fi indispensabile conversiile de suport; acestea vor fi efectuate la Receptorul de date.

Sursa va trebui la sfârșit, să indice numărul de conversie; se presupune că fiecare parte dispune de tabele de conversie.

Semnificația comenzii este; stabilirea unui subcanal.

Comanda ACER1. Această comandă, Acceptare CERere de legătură ACER1, este un răspuns de acceptare trimis de Receptorul de date în cazul cererii CER1. Compoziția comenzii se arată în fig. 2.23.

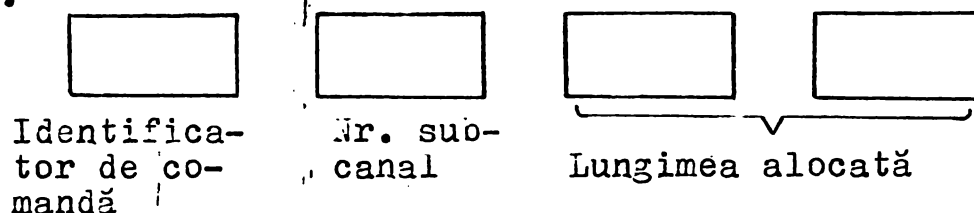


Fig.2.23. Componerea comenzii ACER1.

Lungimea alocată privește tamponul de la care se face scrierea unui articol sau a articolelor în fișier; această lungime trebuie să fie un multiplu al lungimii maxime a articolului.

Semnificația comenzii este; subcanal stabilit.

Comanda NACER1. Această comandă, NeAcceptare CERere... NACER1, este un răspuns negativ transmis de Receptorul de date când comanda CER1 este respinsă. Compoziția comenzii NACER1 este arătată în fig. 2.24.

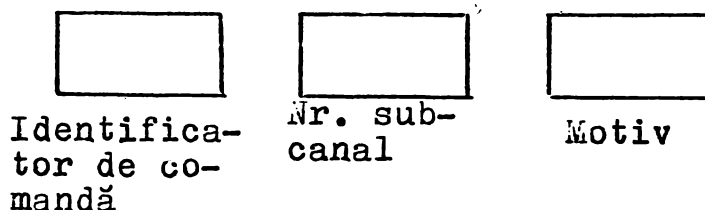


Fig.2.24 Componerea comenzii NACER1.

Blocul "motiv" relevă cauza pentru care a fost respinsă comanda CER1 de către corespondent ca; lipsă de loc în memorie, indisponibilitate component, pană component,

inexistent, etc.

Se pot lua următoarele măsuri:

- Respingerea comenzii CER1, lăsînd utilizatorului expeditor posibilitatea de a reîncepe;
- Formarea unui șir de așteptare cînd motivele de nereușită sînt trecătoare;
- Lăsarea la alegerea utilizatorului.

A doua soluție este mai simplă și reduce traficul de cereri în linie, dar complică gestiunea.

Semnificația comenzii este; subcanal nestabil.

Comanda CER2. Această comandă, CERere de legătură CER2, este transmisă de către, partea care va deveni, dacă cererea este satisfăcută, Receptor de date.

Introducerea acestei comenzi pune probleme mai complicate.

Se notează cu CER (x,y), care are ca semnificație cererea unui subcanal între un component x al părții Sursă de date potențială și componentul y, din partea Receptor de date, cum se arată în fig. 225.

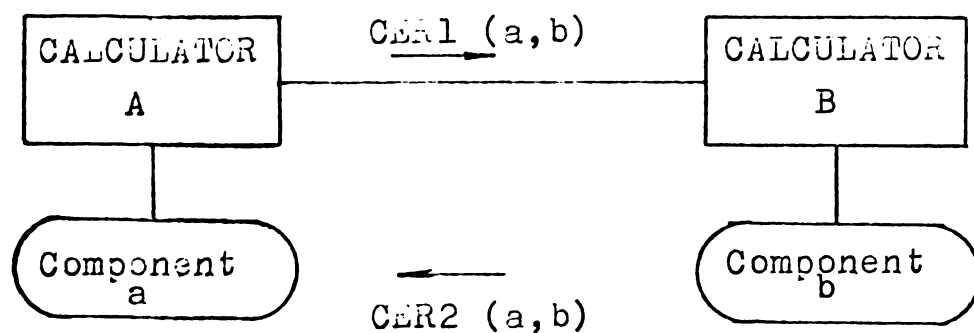


Fig.2. 25.Comanda CER2.

Dacă în același timp, calculatorul A trimite spre calculatorul B comanda CER1 (a,b) și calculatorul B trimite spre calculatorul A comanda CER2 (a,b), este posibil un conflict; de aceea trebuie luate precauțiile necesare pentru a nu risca, fie de a aloca de două ori același terminal, fie de a respinge cele două cereri, cu toată că una din ele putea fi satisfăcută.

Soluția constă în a da prioritate la una din cereri, de exemplu CER1.

Se procedează astfel; înainte de a trimite CER1 se rezervă componentul "a" de la partea Sursă de date (A), iar de la cealaltă parte se crează o tabelă de subcanale Receptor în curs de stabilire.

Partea Receptor de date (B) va asigura un număr provizoriu legăturii cerînd crearea acestuia în comanda CER2.

Dacă CER1 ajunge în B înainte de recepția unui răspuns pozitiv, subcanalul desemnat de comanda CER1 va fi pus în funcțiune.

Acestea conduc la a prevedea pentru comanda CER2, următoarea compoziție ca în fig.2.26.

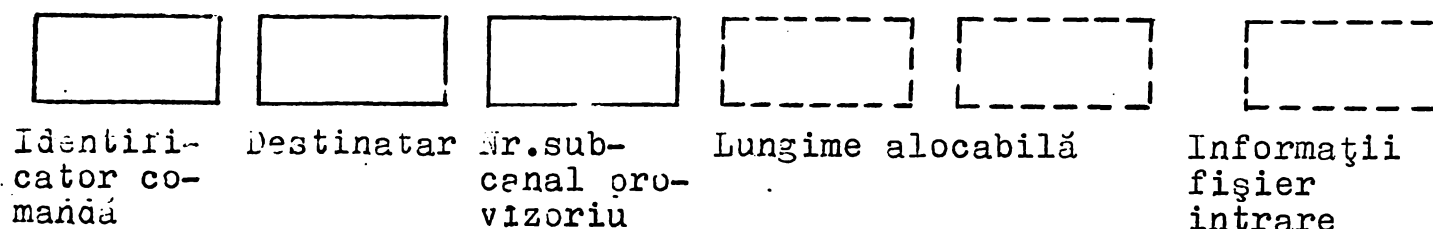


Fig.2.26.Compunerea comenzii CER2.

Blocul "destinatar" se referă la terminalul care servește ca suport de fișier în care se citește de la distanță, lungimea alocată precizează memoria de care se dispune pentru a crea un tampon component, iar alte informații care depind de sistemul de gestiune a fișierelor, vor trebui deasemenea transmise.

Semnificația comenzii: stabilirea unui subcanal.

Comanda ACER2. Această comandă, Acceptare CERere de legătură ACER2, este un răspuns de recepție pozitiv, transmis de viitoarea Sursă de date, dacă cererea CER2 poate fi acceptată, ceea ce implică satisfacerea următoarelor condiții; component existent și disponibil, fișier existent, disponibil, normal deschis, tampon component creat; lungime maximă alocabilă transmisă în CER2, suficientă față de lungimea maximă a articolului.

Se vede că Sursa de date impune, în final numărul subcanalului și că tot aceasta indică mărimea tamponului component de reținut pentru Receptorul de date, ca în fig. 2.27.

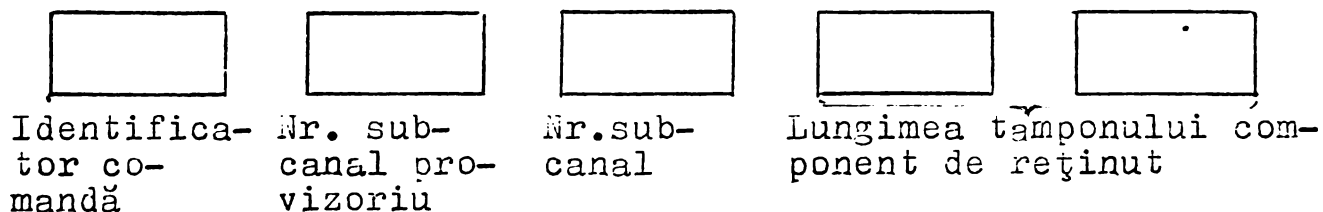


Fig.2.27. Componerea comenzii ACER2.

Semnificația comenzii este; subcanal stabilit.

Comanda NACER2. Această comandă NeAcceptare CERere de legătură NAÇER2, este un răspuns al Sursei de date, dacă comanda CER2 a fost refuzată; se arată în fig.2.28. compoziția comenzii NACER2 și se indică și motivul de respingere.

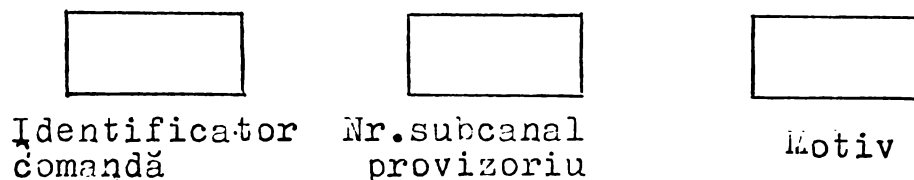


Fig.2.28 Componerea comenzii NACER2.

Semnificația comenzii; subcanalul nestabilit.

#### COMENZILE DE EFECTUARE A SCHIMBURILOR DE DATE IN LINIE

Transmisia datelor în linie pentru un subcanal nu se face decât dacă Receptorul de date se declară pregătit pentru a le primi, adică elementul precedent pentru același subcanal a ajuns la destinație - fișier sau tampon, dacă e vorba de un articol din fișier, fracționat din necesitatea transmisiei.

Datele destinate unui subcanal, transmise în timpul unei emisii/recepții pe linie, constituie un element.

Comenzile de efectuare a schimburilor de date în linie sînt două: ACC și NACC.

Comanda ACC. Avizare de Confirmare Corectă a recepției ACC; informează că partea Receptor de date poate să primească un nou element, fie că dispune de loc în memorie prin eliberarea acesteia, ca urmare a unei scrieri corect efectuată în fișierul Receptorului de date, fie că elementul transmis cere să fie completat și partea Receptor de date așteaptă elementul următor pe o lungime inferioară sau egală cu cea indicată.

Se arată compoziția comenzii ACC în fig. 2.29.

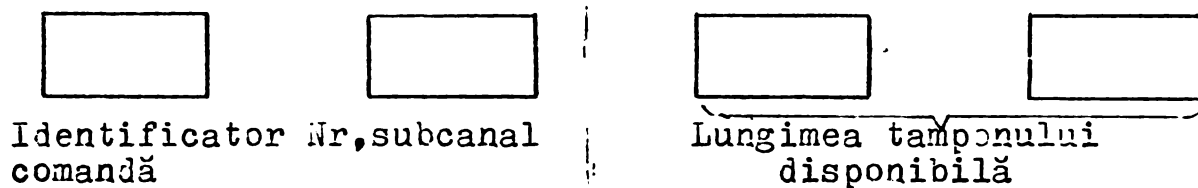


Fig.2.29. Compoziția comenzii ACC.

Semnificația comenzii: element precedent primit corect.

Comanda NACC. NeAvizare Confirmare Corectă a recepției NACC; avertizează Sursa de date că la recepția elementului precedent a existat o anomalie precizînd și cauzele. În funcție de motiv, Sursa de date poate să retransmită elementul respectiv sau să suspende legătura.

Compoziția comenzii NACC se arată în fig.2.30.

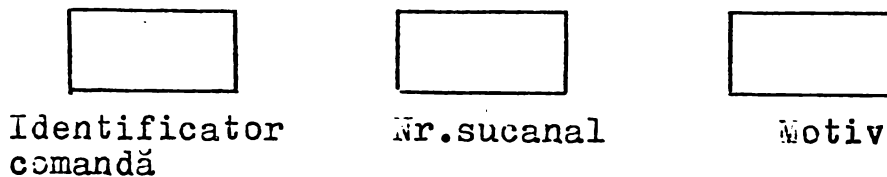


Fig.2.30. Componerea comenzii NACC.

Semnificația comenzii: condiții anormale la recepția elementului precedent.

## COMENZILE DE SUSPENDARE A SCHIMBURILOR DE DATE

Aceste comenzi de suspendare a schimburilor de date în linie sînt de două feluri, după cum se referă la un subcanal de date - ruperea legăturii RPL, sau la un ansamblu de subcanale - închiderea legăturilor INCHI.

Comanda RPL. Această comandă, RuPere Legătură RPL, este transmisă fie de Sursa de date, după o recepție a unei comenzi NACC, cu un motiv care întrerupe legătura - până ireparabilă sau sfîrșit de fișier, fie de Sursa de date sau Receptor de date prin ordin exterior - intervenție operator.

Comanda precizează motivul întreruperii subcanalului pentru a se putea avertiza operatorul printr-un mesaj la consolă.

Compoziția comenzii RPL se arată în fig.2.31.

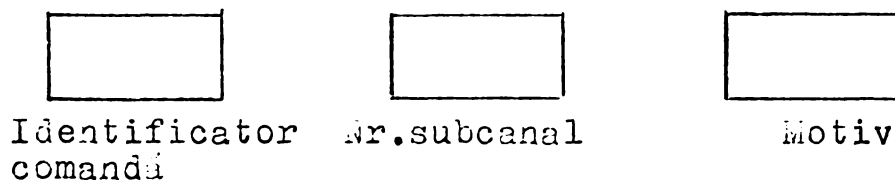


Fig.2.31. Compoziția comenzii RPL.

Semnificația comenzii; ruperea legăturii sau întrerupere subcanal.

Comanda INCHI. Această comandă INCHIdere legături INCHI, este transmisă numai la o comandă din exterior și niciun subcanal nu va mai putea fi stabilit; dacă nu a existat nici o întrerupere de subcanale existente, se așteaptă sfîrșiturile fișierului.

Compoziția comenzii INCHI, este arătată în fig.2.32.

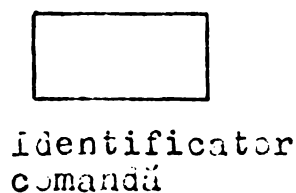
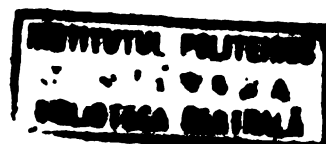


Fig.2.32. Compoziția comenzii INCHI.

Semnificație comenzi: închiderea subcanalelor



## STRUCTURAREA SCHIMBURILOR ÎN LINIE

Un mesaj este constituit dintr-un ansamblu de informații care sînt emise o singură dată; după adăugarea caracterelor conform procedurilor de transmisie se obține un bloc de transmisie. Compoziția mesajului este dată în fig.2.33.

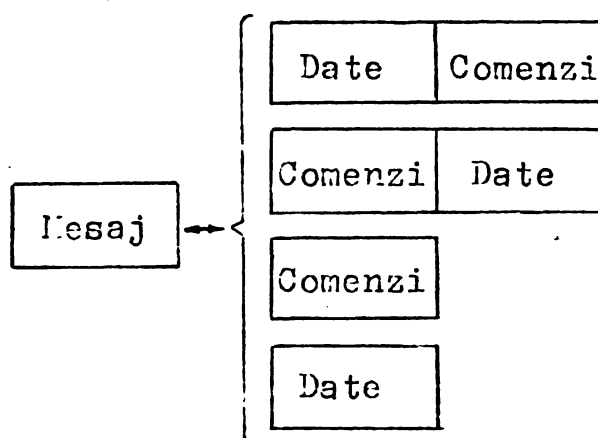


Fig.2.33. Compunerea posibilă a unui mesaj.

Mesajul poate comporta două zone - comenzi și date a căror ordine este indiferentă, fiecare din aceste zone, poate fi reprezentată separat.

Zona de comenzi reprezentată în fig.2.34. începe printr-un identificator COM, urmat de lungimea comenzilor.

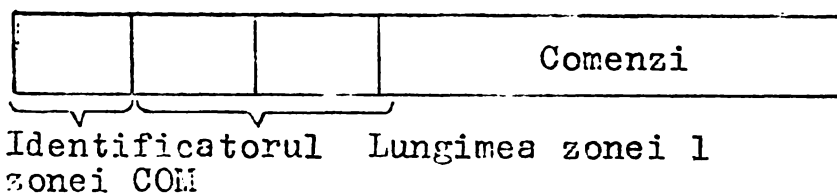


Fig.2.34. Zona de comenzi

Zona datelor reprezentate în fig.2.35. începe cu un identificator de zonă (DAT) urmat de unul sau mai multe elemente.

Fiecare element este constituit dintr-un ansamblu de date destinate unui subcanal și care sînt transmise o singură dată.

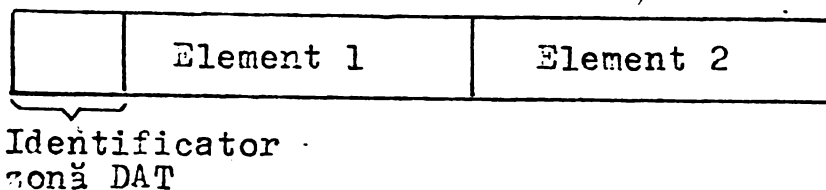


Fig.2.35. Zonă de date

Deci, relațiile între un element și un fișier sînt următoarele; fișierul este divizat într-o suită de elemente pentru transmisie se face după ordinea acestora în fișiere

Un element poate transporta un articol complet, o parte din-

tr-un articol sau mai multe articole complete, fig.2.36, aceste cazuri fiind cunoscute cu ajutorul blocului "structură".

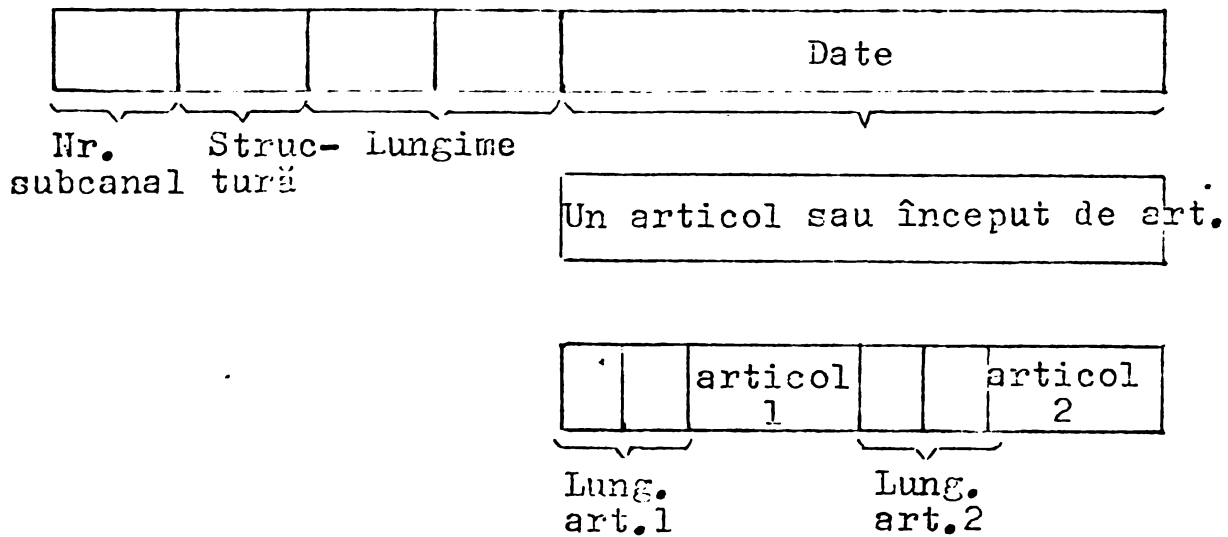


Fig.2.36.. Compunerea unui element.

Cînd un element cuprinde mai multe articole ,fiecare dintre acestea este precedat de lungimea sa.

Această compoziție a zonei de date,permite un "multiplexaj în spațiu" pentru că datele care privesc mai multe subcanale pot fi transmise simultan dar este posibil și un "multiplexaj temporal",adică consultarea fiecărui subcanal pe rînd.

Este necesar un caracter de sfîrșit de mesaj sau o indicație de lungime a zonei de date,dacă se ignorează lungimea mesajului care s-a primit de la linie.

Subcanalul de comunicare între mașinile de scris puștru, nu respectă aceste reguli; el se consideră că este stabilit în permanență și nu există pentru această cerere de stabilire sau răspuns la aceasta.

Se adoptă,astfel, pentru subcanalul mașinii de scris - MS, structura reprezentată în fig.2.37.

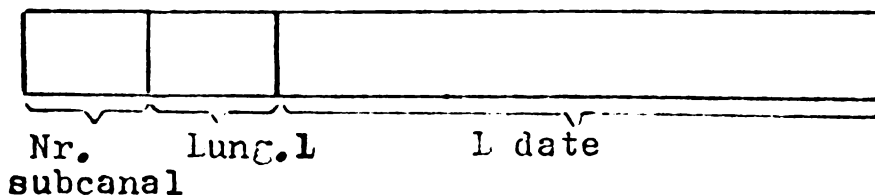


Fig.2.37. Elementul corespondent subcanalului de comunicație între mașinile de scris.



## 2.5. PROCEDURA DE GESTIUNE A SARCINILOR

În această parte a lucrării noastre, am prezentat studii teoretice care cuprind concepte proprii privind elementele teoretice de bază ale unui executiv sau supervisor. Am elaborat acest executiv pentru a satisface o cerință principală a sistemului nostru (constituit din procese paralele sau sarcini multiple) de sincronizare a proceselor, de ordonare a acestora, de alocare a resurselor și altele. Problemele principale asupra cărora mi-am concentrat atenția în cercetările pe care le-am abordat au fost cele mai prioritare și se referă la sincronizarea proceselor din cadrul sistemului nostru și de alocare de memorie pentru tamponurile de teleprelucrare.

Procedura de gestiune a sarcinilor, așa cum am denumit acest executiv de teleprelucrare a fost necesar a fi elaborată, datorită ineficienței și insuficienței serviciilor oferite de sistemul de exploatare standard ale calculatorului; algoritmi greoi și neadaptabili de alocarea memoriei, controlul dificil al execuției și al accesului la memorie, mecanism de sincronizare a proceselor lent și atunci comutarea proceselor se realizează necorespunzător, utilizatorul nu poate interveni în modul de alegere a unui proces pentru a intra în execuție, nefiindu-i accesibile criteriile și prioritățile de alegere, de asemenea, nu poate controla sarcinile, evenimentele, semafoarele și altele.

Am prezentat în continuare unele concepte proprii privind elementele de bază ale unui executiv pentru teleprelucrare, ca sincronizarea dintre procese cu ajutorul semafoarelor și alocarea resurselor de memorie pentru tamponuri.

Studiile noastre teoretice pe aceste probleme fundamentale care stau la baza realizării unui executiv, sînt originale și consider că aduc o contribuție însemnată în acest domeniu; aceasta prezintă posibilități de generalizare și constituie un ghid pentru rezolvarea problemelor de sincronizarea proceselor și alocarea memoriei în cadrul unor sisteme complexe informatice.

În ceea ce privește sincronizarea proceselor, avînd în vedere că nu există pînă în prezent o teorie generală, ci numai soluții particulare, am elaborat pe baza unor cercetări, algoritmi și modele proprii pe care le-am scris într-un limbaj de programare adecvat care corespund situațiilor de sincronizare

a proceselor cu ajutorul semafoarelor, întâlnite în sistemul nostru, dar care pot fi extinse și generalizate. Am acceptat în general, conceptul de sincronizare prin semafor a lui Dijkstra, dar l-am adaptat și dezvoltat pentru cerințele noastre deoarece instrumentul respectiv nu a fost suficient de sistematizat și eficient pentru cazul nostru. Ca urmare, am încercat să găsim o soluție corespunzătoare, corectă, în ceea ce privește sincronizarea proceselor paralele care evoluează în "două spații - timp" diferite și disjuncte cum se întâmplă în sistemul nostru pentru legătura între două calculatoare la distanță; soluția prezentată de Dijkstra se referă la sincronizarea proceselor paralele într-un singur spațiu - timp.

În ceea ce privește alocarea resurselor, având în vedere că pînă în prezent în literatura de specialitate nu a fost prezentată o concepție teoretică generală, efortul nostru l-am îndreptat spre elaborarea unui studiu propriu privind stabilirea unor modele și algoritmi care pot soluționa problemele de alocare a memoriei pentru tampoanele de teleprelucrare necesare sistemului nostru adică a tampoanelor necesare pentru stabilirea legăturii între calculatoare. Algoritmi prezentați sînt eficienți, simpli și reprezintă diverse situații întâlnite în cadrul sistemului nostru.

Studiile noastre teoretice care se referă la elementele de bază ale executivului de teleprelucrare au luat în considerare, cele mai recente lucrări de specialitate din acest domeniu, care ne-au inspirat și ne-au permis acumularea unor cunoștințe valoroase, valorificate practic în cercetările noastre.

De un real folos au fost lucrările lui: Dijkstra, privind sincronizarea proceselor cu ajutorul semafoarelor; Denning, privind arhitectura și elementele de bază ale unui executiv; Gelenbe, privind calculele matematice de alocare a memoriei; Brandwajn, privind unele probleme teoretice de sincronizare a proceselor; Baltac, privind unele aspecte teoretice de alocare a memoriei; Ferrié și Mossière, privind gestiunea proceselor și partajarea resurselor și alții. De asemenea alte lucrări de bază consultate sînt prezentate în bibliografie (notate cu 2) iar cele folosite mai mult au fost marcate cu asterisc.

Ca atare, am putea spune că există o teorie generală privind realizarea executivelor, lucrărilor de specialitate oferindu-ne numai soluții teoretice foarte particulare de multe ori empirice: deasemenea se face simțită lipsa de in-

stramente generale în acest domeniu (firmele de calculatoare au realizări practice dar foarte diversificate și incompatibile).

Avînd în vedere aceste considerații am fost constrîns să procedez la elaborarea unor cercetării proprii în acest domeniu pentru a soluționa problemele ridicate de sincronizarea proceselor aflate în două calculatoare la distanță precum și de alocare de memorie pentru tamponare.

### SINCRONIZAREA PROCESELOR. SEMAFOARE

Ideia de la care am pornit a fost că, în sistemul nostru, procesele să nu evolueze în mod anarhic independent unele de altele ci să existe relații între ele care să stabilească desfășurarea lor în timp; procesele pot astfel influența alte procese sau pe ele însuși pentru a coopera la realizarea unei sarcini comune, aplicația de teleprelucrare a datelor.

Apare deci, problema fundamentală studiată și rezolvată de noi din punct de vedere teoretic în această parte a lucrării de a asigura sistemul nostru cu un mecanism de sincronizare puternic și eficient. Am avut de rezolvat astfel două tipuri de probleme de sincronizare distincte; de excludere mutuală sau bariera de acces la o secțiune critică și de atașare a unui proces printr-un semnal, la un eveniment.

Am ales pentru rezolvarea problemei noastre, de sincronizare între procese, mecanismul semafoarelor ca cel mai puternic și mai corespunzător mijloc de sincronizare. Am constatat că pentru rezolvarea sincronizării proceselor din lucrarea noastră metodele de programare obișnuite sînt insuficiente și prezintă inconveniente (obligă să se introducă secvențe speciale pentru blocajul între procese și să se țină seama de procesul real).

Ca urmare pentru rezolvarea algoritmilor concepuți de noi pentru anumite situații de sincronizarea proceselor cu ajutorul semafoarelor am utilizat o variantă a limbajului Algol (deoarece acest limbaj nu ne permite descrierea proceselor paralele) denumită pseudo Algol, limbaj care completează Algolul introducînd două limitatoare suplimentare "parbegin și parend" astfel că instrucțiunile cuprinse între acestea pot fi executate în paralel.

Mecanismul semaforului. Semaforul înțeles în sensul dat de Dijkstra, este constituit din asocierea unei variabile care ia valori întregi eventual negative și a unui șir de așteptare.

Variabila, comună proceselor care se sincronizază, trebuie să fie inițializată la o valoare care nu este negativă, înainte ca procesele să fie lansate și mai mult, aceasta trebuie declarată într-un spațiu la care procesele au acces. Operațiile de semafoare se fac prin intermediul primitivelor; sînt introduse două primitive de bază, notate cu P și V pe lângă alte două de creare și distrugere semafor. Aceste primitive acționează asupra semaforului (sem) astfel:

P(sem) - diminuează cu 1 valoarea variabilei asociate semaforului sem; dacă valoarea rezultată este pozitivă sau nulă acțiunea primitivei se oprește aici, iar dacă din contră valoarea este negativă, procesul care execută această primitivă se blochează și trece în șirul de așteptare al semaforului sem.

V(sem) - mărește cu 1 valoarea variabilei asociate semaforului sem; dacă valoarea rezultată este negativă sau nulă, un proces din șirul de așteptare este activat, iar dacă valoarea este pozitivă acțiunea primitivei se oprește aici.

Deci, cele două operații primitive întreruptibile, care permit blocarea și activarea proceselor sînt prezentate în continuare:

```
P(sem):  begin
          v(sem): = v(sem) - 1
          if   v(sem) < 0 else
              begin
                stare (p) : = blocat, procesul p este pus în
                               șirul de așteptare s(sem)
              ..
              .
              .
              end
V(sem):  begin
          v(sem): = v(sem) + 1
          if   v(sem) > 0 else
              begin
                ..
                ..
                ..
                stare(p'): = activ procesul p' este scos
                               din șirul de așteptare
                               s(sem).
              .
              .
```

end

end

unde semaforul "sem" are;  $v(\text{sem})$  variabilă întreagă și  $s(\text{sem})$  șirul de așteptare.

Mecanismul semafoarelor permite rezolvarea a două tipuri de probleme - bariera de acces la o resursă și semnalul care permite de a atașa un proces sau mai multe, la un eveniment exterior.

Semafoarele care permit a se realiza aceste două funcțiuni sînt, respectiv semaforul de excludere mutuală și semaforul privat

Ținînd cont de condițiile impuse sistemului nostru de gestionare a legăturii între calculatoare la distanță a trebuit să elaborez o serie de algoritmi de sincronizare a proceselor paralele cu ajutorul semafoarelor. Am plecat evident la rezolvarea problemei de la noțiunea de semafor a lui Dijkstra dar care are o serie de insuficiențe ca tratarea cooperării proceselor numai sub aspect temporal și imposibilitatea activării mai multor procese în același timp etc; această noțiune de semafor a fost adaptată la specificul nostru, modificată corespunzător și dezvoltată pentru a răspunde, corect la sincronizarea proceselor în două spații diferite.

Sincronizarea proceselor cu ajutorul semafoarelor. Prezentăm în continuare probleme privind sincronizarea proceselor cu ajutorul semafoarelor din cadrul sistemului nostru.

Astfel foarte frecvent avem de rezolvat probleme de sincronizare a unui proces cu sosirea unui eveniment; evenimentele sînt simple și compuse.

Evenimente simple: Dacă sincronizarea se realizează cu eveniment simplu și dacă acestea urmează precis să sosească, este necesar ca procesul care îl așteaptă să efectueze o operație P. Pentru aceasta se creează un semafor "eveniment sosit" cu valoarea inițială 0.

Procesul 1 este procesul care așteaptă evenimentul, procesul 2 asigură sosirea evenimentului. Astfel avem:

begin integer

parbegin

proces 1 : begin

•  
•  
•  
•  
•

```

P (eveniment sosit);
prelucrare legată de sosirea evenimentului;
.
.
.
end
proces 2 : begin
.
.
.
V (eveniment sosit)
.
.
.
end;
parend
end"
```

Dacă nu există siguranță că evenimentul se va produce, acest gen de secvență nu este recomandabil.

Se definește atunci un drapel (flag) cărui i se protejează accesul printr-un semafor. Se arată în continuare, un exemplu de prelucrare în funcție de sosirea unui eveniment. Astfel avem;

```
"begin intener flag, acces la flag;
acces la flag: = 1;
flag: = 0;
parbegin
proces 1 : begin
.
.
.
P(acces la flag);
if flag = 1 then
begin
flag : = 0;
prelucrare 1;
end
else prelucrare 2;
V (acces la flag)
.
.
.
"
```

```
      .  
      .  
      .  
      end;  
    proces 2 : begin  
eveniment :  
      P (acces la flag);  
      flag 1 : = 1;  
      V (acces la flag)  
      .  
      .  
      .  
      end  
    parend  
end"
```

Evenimente compuse. In cadrul sincronizării evenimentelor compuse, se disting două modalități; intersecția evenimentelor și reuniunea evenimentelor.

Intersecția evenimentelor se referă la faptul că un eveniment așteptat poate fi o combinație logică de mai multe evenimente. Sincronizarea unui proces cu sosirea unui eveniment (eveniment 1 sosit)  $\cap$  (eveniment 2 sosit) .... , nu prezintă dificultăți particulare.

Se definesc astfel, semafoarele pentru evenimente compuse, cu valori inițiale nule. Apoi este suficient de a efectua o secvență de operații P una după alta pe semafoarele definite. Astfel, avem;

```
"begin integer eveniment 1 sosit, eveniment 2 sosit,  
      eveniment 3 sosit;  
eveniment 1 sosit: = 0;  
eveniment 2 sosit: = 0;  
eveniment 3 sosit: = 0;  
      parbegin  
      proces 1 : begin  
      .  
      .  
      .  
      P (eveniment 1 sosit);  
      P (eveniment 2 sosit);  
      P (eveniment 3 sosit);
```

```
prelucrare;
.
.
.
end;
proces 2 : begin
.
.
V (eveniment 1 sosit);
.
.
end
proces 3 : begin
.
.
V (eveniment 3 sosit);
.
.
end;
parend
end"
```

Reuniunea evenimentelor, se referă la punerea unui proces în așteptarea unui eveniment dintre  $n$ . Problema nu este simplă, pentru că nu se poate iniția o suită de operații  $P$ , deoarece este imposibil să se prevadă care va fi evenimentul sosit.

Se definește un semafor comun cu valoarea inițială 0 și de fiecare dată, când un eveniment dintre evenimentele așteptate sosește, se efectuează o operație  $V$  pe acest semafor. Astfel avem;

```
"begin interer semafor comun;
semafor comun: = 0;
parbegin
proces 1 : begin
.
.
.
P (semafor comun);
prelucrare;
.
.
```



```
    end;  
    proces 2 : begin  
        .  
        .  
        .  
    eveniment 1 : V (semafor comun);  
        .  
        .  
        .  
    end;  
    proces 3 : begin  
        .  
        .  
        .  
    eveniment 2 : V (semafor comun);  
        .  
        .  
        .  
    end;  
parend  
end"
```

Se remarcă utilitatea păstrării identității evenimentelor sosite; aceasta se poate face poziționând drapelele corespunzătoare evenimentelor. Acest mod de a proceda se arată în fig.2.38:

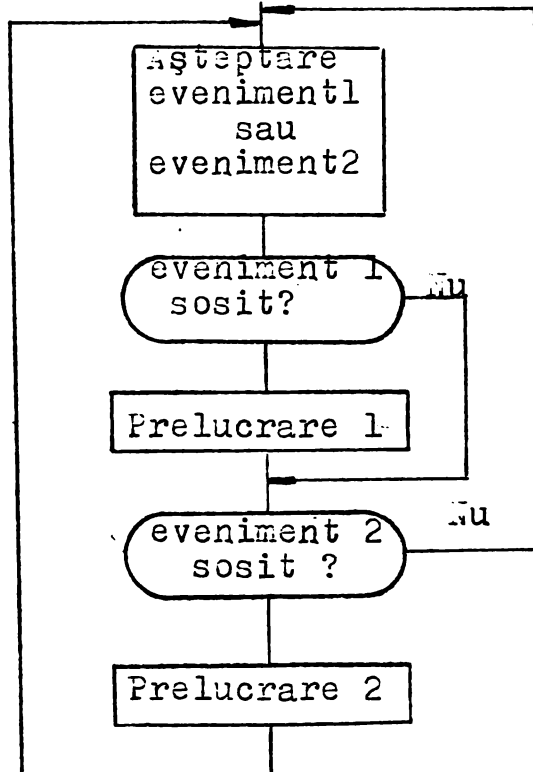


Fig.2.38. Reuniunea evenimentelor

Drapelul "flag 1" corespunde la evenimentul 1 și drapelul "flag 2" corespunde la evenimentul 2, în următorul mod;

flag 1 = 0 eveniment 1 nu a sosit

flag 1 = 1 eveniment 1 a sosit

și același lucru pentru flag 2 și eveniment 2.

Programul devine astfel;

"begin integer semafor comun, flag 1, flag 2;

semafor comun : = 0;

flag 1 : = 0;

flag 2 : = 0;

parbegin

proces 1 : begin

așteptare: P. (semafor comun);

if flag 1 = 1 then

begin

flag 1 : = 0;

prelucrare 1;

end;

if flag 2 = 1 then

begin

flag 2 : = 0;

prelucrare 2;

end;

noto așteptare;

end:

proces 2 : begin

.

.

.

eveniment 1:

flag 1 : = 1;

V (semafor comun);

.

.

.

end;

proces 3 : begin

.

.

.

eveniment 2 :

```
flag 2 := 1;  
V (semafor comun);  
.  
.  
.  
end
```

parend

end".

Programul nu este totuși corect; s-a presupus că evenimentele 1 și 2 nu pot sosi din nou, atât timp cât procesul 1 nu a terminat prelucrările 1 și 2. Ca urmare se poate ajunge la situația că procesul 1 poate termina operația sa P pe semaforul comun fără ca el să fi efectuat o prelucrare oarecare; este suficient pentru aceasta ca cele două procese 2 și 3 să efectueze operațiile V pe semaforul comun aproape simultan.

Soluția fără defectele arătate mai sus este prezentată în continuare. La fiecare condiție realizată se asociază o operație P pe un semafor comun, ceea ce face ca la un moment dat când se trece în așteptare, valoarea semaforului comun să fie egală cu numărul evenimentelor sosite și netratate.

"begin înteger semafor comun, flag 1, flag 2;

semafor comun := 0;

flag 1 := 0;

flag 2 := 0;

parbegin

proces 1 : begin

așteptare

P (semafor comun)

V (semafor comun)

if flag 1 = 1 then

begin

flag 1 := 0

prelucrare 1;

P (semafor comun);

end;

if flag 2 = 1 then

begin

```
        flag 2 := 0;
        prelucrare 2;
        P (semafor comun)
        end
        goto asteptare;
    end;
proces 2 : begin
    .
    .
    .
    end
    parend
end"
```

In continuare se prezintă justificarea algoritmului de așteptare a unui eveniment dintre n.

Se presupune că fiecare eveniment nu se poate produce din nou atît timp cît nu a afectat prelucrarea asociată. Această situație se rezumă astfel;

```
"begin
    integer S, P1, P2, flag 1, flag 2;
    flag 1 := 0;
    flag 2 := 0;
    S := 0;
    P1 := 0;
    P2 := 0;
    parbegin
    T : begin
        P (S);
        V (S);
        if flag 1 = 1 then
            begin
                flag 1 := 0;
                P (S);
                V (P1);
            end;
        if flag 2 = 1 then
            begin
                flag 2 := 0
```

```
        P (S);
        V (P2);
        end
        goto T;
Ev 1 : begin
        if eveniment 1 then
            begin
                flag 1 : = 1;
                V (S);
                P (P1);
            end
            else
                goto Ev 1;
            end
Ev 2 : begin
        if eveniment 2 then
            begin
                flag 2 : = 1;
                V (S);
                P (P2);
            end
            else
                goto Ev 2;
            end;
        parend
end".
```

Se remarcă că, în momentul intrării pentru prima dată în T, sînt posibile patru configurații:

- nici un eveniment nu a sosit; flag 1 și flag 2 = 0, S=0;
- evenimentul 1 a sosit; flag 1=1, flag 2=0, S=1;
- evenimentul 2 a sosit; flag 1=0, flag 2=1, S=1
- evenimentul 1 și evenimentul 2 au sosit; flag 1=1, flag 2=1, S=2.

În primul caz, procesul T se va bloca și va rămîne blocat pînă cînd un proces, Ev 1 sau Ev 2 va efectua V (S).

În al doilea caz, semaforul S avînd valoarea 1, operația P va putea fi terminată în momentul cînd se efectuează operația V, S poate avea valoarea 0 sau 1, dacă între timp evenimentul 2 a sosit.

Luînd prima ipoteză, Sare din nou valoarea 1 după operația V; se întrerupe pe 0 flag 1 și se autorizează o nouă sosire a evenimentului 1; operația P(S) poate să fie trecută și procesul continuă testînd flag 2.

În a doua ipoteză, dacă evenimentul 2 a sosit în timp ce se prelucra evenimentul 1, flag 2 este pe 1 și semaforul S are valoarea 1; a treia operație P va fi deci, de asemenea, trecută și dacă evenimentul 1 nu este produs din nou în timpul prelucrării evenimentului 2, S are valoarea 0 cînd se reîntoarce în T.

Al treilea caz se examinează într-un mod analog.

În al patrulea caz, S are valoarea 2 cînd se intră în T pentru prima oară; se tratează primul eveniment, cînd se efectuează  $V(P_1)$ , S este egal cu 1; rămîne de prelucrat evenimentul 2.

Cînd se revine în T și dacă evenimentul 1 nu a sosit din nou în timpul prelucrării evenimentului 2, S este egal cu 0.

Se vede că dacă flag 1 și flag 2 sînt amundouă la 0 cînd se efectuează instrucțiunea T, operația P pe care acesta o înțelege nu va putea fi terminată și nu se mai parcurge nicio dată bucla fără ca să aibă o activitate de efectuat; se remarcă că, dacă nu este astfel rezolvată, procesul T nu se va opri niciodată, pentru că prima operație P este urmată de o operație V pe același semafor.

#### ALOCAREA RESURSELOR

Alocarea resurselor în special a memoriei centrale, constituie un obiectiv principal al Procedurii de gestiunea sarcinilor, obiectiv pe care l-am realizat cu ajutorul unor algoritmi speciali concepuți în lucrarea noastră.

În general în literatura de specialitate nu există o concepție teoretică suficient de generală în acest sens și de aceea prezentăm în continuare unele concepte proprii pe acest subiect materializate prin algoritmi corespunzători.

Studiul nostru teoretic privind alocarea resurselor (memorie centrală) conținînd algoritmi corespunzători, se referă concret că la interpretarea comenzii de stabilirea unui subcanal (care leagă calculatoarele aflate la distanță), va fi alocat tamponul component afectat aceluia subcanal.

Algoritmul de alocare propus pentru Sistemul nostru este prezentat în continuare.

Fie:

$l_1$  - lungimea maximă a articolului fișierului considerat, presupusă cunoscută;

$l_2$  - lungimea maximă a zonei de date cuprinzând celelalte elemente decât cele ale subcanalului mașină de scris.

Notăm cu:  $l_d$  - spațiul de memorie disponibil;

$l$  - lungimea bază de alocare.

Se disting două cazuri în funcție de faptul că articolul din fișier va fi sau nu decupat pentru transmisia sa în linie.

Algoritmul de alocare în cazul că articolul nu va fi segmentat.

Atunci avem  $l_1 \leq l_2$ , articolul nu va fi segmentat.

Dacă spațiul disponibil  $l_d$  este mai mare decât lungimea maximă a zonei de date  $l_2$ , adică  $l_d > l_2$ , lungimea bază  $l$  va corespunde cu numărul maxim de articole a căror sumă de lungimi rămâne inferioară lungimii maxime a zonei de date; se scrie,

$$l = \left[ l_2 / l_1 \right] * l_1$$

unde,  $[ ]$  = întreg.

Dacă spațiul disponibil  $l_d$  este inferior lungimii maxime a zonei de date  $l_2$  adică,  $l_d \leq l_2$  și rămânând superior unei anumite limite și lungimii articolului, atunci se calculează lungimea bază  $l$  ca corespunzând numărului maxim de articole a căror sumă a lungimilor rămâne inferioară spațiului disponibil.

$$l_1 \leq l_d \leq l_2 \quad \text{și} \quad kl_2 \leq l_d$$

atunci;

$$l = \left[ l_d / l_1 \right] * l_1$$

unde  $kl_2$  reprezintă limita inferioară.

Cînd spațiul disponibil este inferior limitei  $kl_2$ , se consideră că alocarea este imposibilă și subcanalul nu va putea fi stabilit.

Lungimea efectivă alocată va fi superioară lungimii bază  $l_1$ , pentru a ține cont de informațiile auxiliare - număr subcanal, structură, lungime care se regăsesc de asemenea în tamponul corespondent. Se adaugă la lungimea bază o oarecare lungime de rezervă  $l_r$ , care se determină ușor în cazul articolelor de lungime fixă;

$$l_r = 4 + 2 \lceil l/l_1 \rceil,$$

unde;

4 octeți de descriere a elementului și 2 octeți lungime pe articol.

Este mai greu de a se determina pentru articolele de lungime variabilă; se poate lua ca măsură de precauție

$$l_r = 4 + 4 \lceil l/l_1 \rceil$$

Algoritmul de alocare în cazul că articolul este segmentat.

Atunci avem;  $l_1 \cong l_2$ , articolul va trebui să fie segmentat.

Dacă spațiul disponibil este superior lungimii maxime a articolului, lungimea bază trebuie să fie făcută egală cu aceasta din urmă; altfel, cererea va fi respinsă și subcanalul nu va putea fi stabilit. Ca și în cazul precedent, se adaugă o lungime de rezervă  $l_r$ .

Algoritmul de alocare a tamponului component pentru un subcanal este prezentat în fig.2.39.

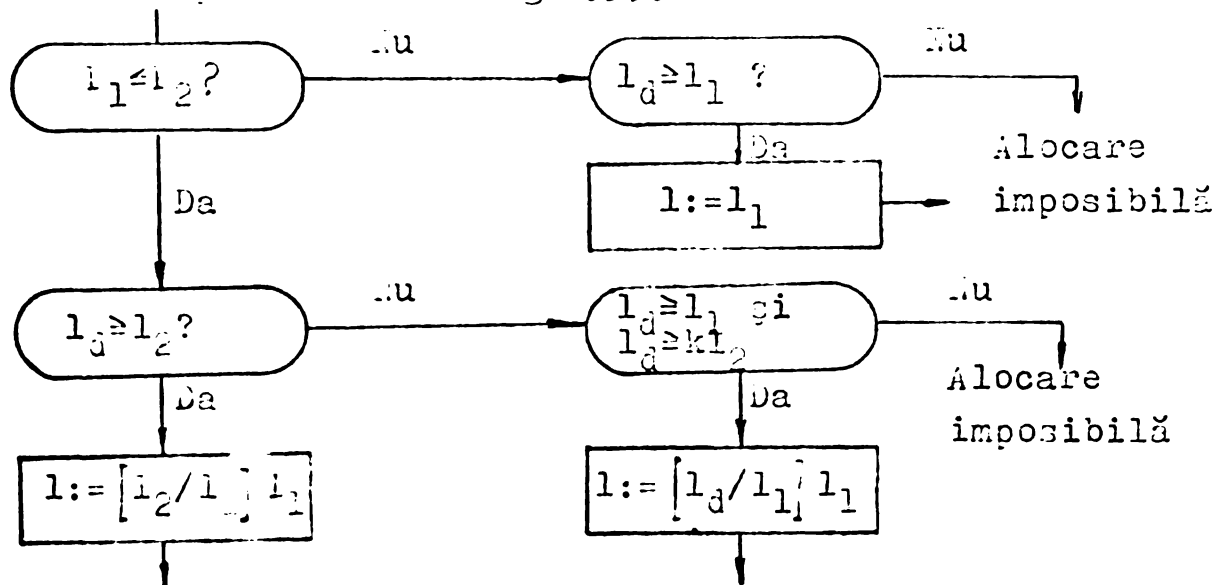


Fig.2.39. Algoritmul de alocare a tamponului component pentru subcanal.



Același algoritm va fi utilizat și pentru legătura la receptorul de date.

Se remarcă că algoritmul respectiv conduce la o subutilizare de memorie prin fragmentare.

Alocarea memoriei și eliberarea sa prin paragrafe.

Există ipoteze de alocare a memoriei și eliberarea sa prin "paragrafe" de mărime fixă.

Astfel, în continuare prezentăm o demonstrație care constituie un mod de rezolvare al acestei probleme.

Se consideră starea de echilibru, în care toate cererile de memorie sînt satisfăcute.

Se numește segment, o cerere de memorie. Se notează cu  $N_0$  - numărul de segmente în memorie fără paragrafizarea, cu  $N$  numărul de segmente în memorie cu paragrafizare și cu  $e = \frac{N}{N_0}$  măsura de eficacitate a paragrafizării.

Fie;

$m$  - mărimea memoriei,

$f$  - mărimea paragrafului,

$q_i$  - probabilitatea ca un gol să aibe mărimea  $i$ ,

$p_i$  - probabilitatea ca un plin să aibe mărimea  $i$ ,

$z_j$  - probabilitatea ca să fie  $j$  goluri.

Probabilitatea  $p$ , ca o cerere să nu găsească un gol de mărimea sa, se scrie:

$$p = \sum_{i=1}^{\infty} p_i \sum_{j=0}^{\infty} z_j (1 - q_i)^j \quad (1)$$

Dacă un segment este de mărimea  $x$ , se alocă  $\left[ \frac{x}{f} \right]$  paragrafe.

Se notează cu  $r$  mărimea medie a unui plin și cu  $s$  mărimea medie a unui segment, rezultă,

$$r = E \left\{ \left[ \frac{x}{f} \right] f \right\} = s + \frac{f}{2} + g \quad (2)$$

Pentru un număr de distribuții de probabilitate uniforme, Erlang;

$g = 0$  și deci,

$$r = s + \frac{f}{2} \quad (3)$$

Se demonstrează că în echilibru se poate scrie;

$$m = N(1 + \frac{p}{2}), \quad \text{din literatura de specialitate.}$$

unde  $m$  este mărimea memoriei.

Rezultă;

$$N = m/r \cdot 1/(1+p/2) \quad \text{și} \quad N_0 = m/r_0 \cdot 1/(1+p_0/2) \quad (4)$$

unde,

$$r_0 = s \left\{ \begin{array}{l} x_j = s; \\ p_0 \approx 1 \text{ din experiență;} \end{array} \right.$$

$$e = N/N_0 = r_0/r \cdot (1+p_0/2)/(1+p/2) = (2+p_0)/(2+p) r_0/r,$$

deci;

$$e = s/r \cdot 3/2+p \quad (5)$$

Se arată că  $e$  poate fi mai mare ca unu, în anumite condiții și că memoria va fi deci mai bine utilizată dacă este divizată în paragrafe;

$$p = \sum_{i=1}^{\infty} p_i \sum_{j=0}^{\infty} z_j (1-q_i)^j \approx \sum_{i=1}^{\infty} p_i \sum_{j=0}^{\infty} z_j (1-jq_i) \quad (6)$$

Admițând că  $q_i \ll 1$  relația (6) se scrie;

$$p = \sum_{i=1}^{\infty} p_i \sum_{j=0}^{\infty} z_j - \sum_{i=1}^{\infty} p_i q_i \sum_{j=0}^{\infty} j z_j$$

și se obține;

$$p = 1 - M \sum_{i=1}^{\infty} p_i q_i \quad (7)$$

$$M = \sum_{j=0}^{\infty} j z_j, \quad \text{unde } M \text{ este deci numărul mediu de goluri.}$$

Presupunând că  $p_i$  urmează o distribuție uniformă -

$p_i = f/2s$ , și că  $f \ll m$ , adică mărimea unui paragraf, este mult mai mică decât cea a memoriei, rezultă din calcule și se vede că dacă  $M$  este destul de mare, cazul obișnuit  $e > 1$ .

### 3. DEFINIȚIA DE PREȚUL DE BAZĂ AL SISTEMULUI

#### 3.1. Definiții

În această parte a lucrării, prezentăm tratarea matematică a sistemului nostru, legată de comportamentul său, pentru evaluarea parametrilor de bază ai sistemului.

Aparatul matematic pe care l-am folosit și care este cel mai adecvat scopului lucrării noastre, respectiv evaluarea parametrilor de bază ai sistemului, constă din teoria modelelor de așteptare legată de importanța tehnicii de calcul probabilistic. Prin aceasta am determinat soluția optimă a problemei încărcării maxime pe care o poate suporta sistemul, astfel ca, pe de o parte, cererile de prelucrare ale utilizatorilor să fie cât mai prompt satisfăcute, iar pe de altă parte, capacitatea sistemului să fie cât mai bine folosită și să permită ca intensitățile maxime de trafic să poată fi prelucrate fără prea mari întârzieri.

Abordarea acestei probleme ne-a condus la construirea și rezolvarea unor modele matematice stochastice care caracterizează în esență comportamentul sistemului nostru și prin a căror analiză am putut determina parametrii de bază și optimele pentru aceștia, ceea ce ne permite trecerea la dimensionarea sistemului alocând a resurselor necesare, în special capacitatea de memorare, capacitatea liniilor de comunicație, viteza unităților de prelucrare și altele.

Caracterul stochastic și existența șirurilor de așteptare în sistemul nostru se datoresc atât mesajelor care apar aleatoriu, ca intrări în sistem, cât și prelucrărilor acestora în resursele sistemului, fenomene despre care putem presupune că sînt descrise de procese stochastice atât de tip exponențial prin forțarea realității, cât și mai corect, de procese cu distribuții generale care, în acest sens, oglindesc realitatea într-o măsură mai mare.

Sistemul nostru ca element de bază care asigură legătura între două sau mai multe calculatoare, de capacitate de memorare în tampoane limitată, din cadrul unei rețele de calculatoare, îl putem reprezenta în ceea ce privește problematica noastră de transmitere și prelucrare a mesajelor de la un calculator la alt calculator, printr-un model matematic cu două sau mai

multe stații în serie și cu șiruri de așteptare de mărime limitată. În acest model, calculatorul este prezentat printr-un sistem de așteptare unde unitatea de prelucrare semnifică stația, iar capacitatea de memorare din tamponașele pentru teleprelucrare semnifică mărimea șirurilor de așteptare.

Prin modelele matematice pe care le-am construit și analizat am încercat să reflectăm cât mai corect structura și comportamentul sistemului din realitate, acordând o importanță deosebită fenomenelor de așteptare datorate transmiterii mesajelor și aglomerării acestora la resursele calculatoarelor.

Fenomenele reale din cadrul rețelelor de calculatoare conduc în general la modele complexe și acestea s-au dovedit dificil de rezolvat prin analize matematice din cauza unor calcule foarte greoaie și chiar imposibile. Ca atare, am tratat în general modele cu anumite simplificări făcute în limitele prelucrării matematice abordabile care, deși dau rezultate aproximative, conduc totuși la estimarea în bune condiții a parametrilor esențiali ai sistemului.

Tratarea matematică a modelelor noastre a avut ca scop în esență evaluarea numărului maxim de mesaje, sau matematic spus de clienți, care pot traversa sistemul nostru, respectiv modelul stochastic corespunzător, în unitatea de timp, adică de a evalua rata maximă de încărcare notată de noi cu  $\rho_{max}$ . Acest parametru, cel mai important care caracterizează modelul, ne permite evaluarea performanțelor acestuia și, în general, a rețelelor de calculatoare; ca urmare, obiectivul rezolvării noastre matematice este determinarea acestui parametru în diverse situații particulare și generale legate de numărul de stații în sistem și tipul distribuțiilor de probabilitate pentru sosiri și servicii.

Tratarea matematică a sistemului nostru și în general a rețelelor de calculatoare a necesitat un efort de cercetare în ceea ce privește alegerea unor modalități de rezolvare a modelelor corespunzătoare. Astfel am fost conduși la alegerea a două modalități de rezolvare, mai avantajoase, care ne oferă rezultate satisfăcătoare și prezintă eficiență practică. Aceste căi de rezolvare se deosebesc prin aparatul matematic folosit, prin eficiența practică și prin aria de cuprindere.

Am abordat inițial o modalitate de tratare matematică într-un stil clasic, adică printr-o analiză matematică detaliată bazată pe un aparat matematic, frecvent utilizat, de al-

gebră și probabilități, deci o tratare matematică cu un pronunțat caracter analitic și totodată teoretic.

Datorită aparatului matematic folosit, formulele găsite pentru determinarea parametrului de bază ne-au condus la rezultate exacte, explicite, dar limitate numai la modele cu un număr mic de stații în serie și cu distribuții exponențiale.

Am considerat că această rezolvare poate constitui o metodă analitică de analiză și calcul deoarece dispune de un aparat matematic omogen și mai puțin complicat, o concepție de modelare bine sistematizată, de posibilitatea de generalizare și de extindere.

Totuși, din cauza aproximării introduse de către modelele exponențiale și a limitărilor aparatului matematic care se complică și nu poate da soluții pentru un sistem cu un număr mare de stații în serie sau cu distribuții generale, cum este în cazul rețelelor de calculatoare, această metodă analitică nu ne satisface decât parțial.

Ca urmare am abordat o altă modalitate de tratare matematică cu caracter aplicativ, mai eficientă și mai realistă, bazată pe tehnici matematice foarte actuale de aproximare, prin care putem rezolva modele mai complexe, care reprezintă mai bine fenomenele reale din cadrul rețelelor de calculatoare. Mai precis, am elaborat noi tehnici de aproximare; am conceput o tehnică de aproximare proprie, "prin echivalență" care permite echivalarea unui model complex cu șiruri de așteptare în serie, printr-un model cu stații independente și cu soluții cunoscute și deasemenea am adaptat la cerințele noastre și am dezvoltat tehnica de aproximare prin procese de difuzie, care permite înlocuirea unui proces stochastic complex, general cu un proces markovian continuu și care are soluții matematice. În funcție de natura și complexitatea modelelor de rezolvat am ales atît una sau cealaltă tehnică, cît și ambele, în cazuri mai complicate, pentru a se completa între ele.

Am considerat că această modalitate de rezolvare poate constitui o metodă prin aproximare, de analiză și calcul, deoarece are o largă aplicabilitate practică, dispune de un grad mare de generalizare și de extindere putînd aborda sisteme cu un număr mare de stații și cu diferite distribuții de probabilitate, prezintă ușurință în rezolvarea concretă a modelelor matematice deși se bazează pe un aparat matematic complicat și

foarte actual.

Cu toate că această metodă conduce la soluții aproximative, rezultatele obținute sînt în general superioare celor găsite prin metoda analitică cu modele exponențiale, modele care pînă în prezent, erau frecvent folosite în domeniul sistemelor cu calculatoare electronice.

Referitor la aceste două metode prezentate, putem concluziona că cea analitică prezintă anumite limitări și restricții față de cea prin aproximare, care, de fapt, reprezintă opțiunea noastră, susținută de ideea de a nu căuta soluții exacte pentru ecuațiile de sistem ale modelului, ceea ce este foarte complicat sau imposibil în anumite cazuri, ci de a accepta soluționări aproximative, dar care să ofere totuși rezultate corecte despre comportarea reală a sistemului.

Putem aprecia că în ultima perioadă de timp, pe plan mondial dezvoltarea sistemelor de teleprelucrare a datelor și în general, a rețelelor de calculatoare a determinat elaborarea și apariția unor lucrări de cercetare cu privire la evaluarea parametrilor de bază ai modelelor și deci a performanțelor rețelelor de calculatoare. Dar, cu toate acestea, performanțele rețelelor rămîn încă insuficient cunoscute și deci în continuare constituie un subiect deschis pentru noi abordări de studii și cercetări.

Cauzele principale care nu au permis ca aceste lucrări să răspundă cerințelor noastre generate de evaluarea în bune condiții a performanțelor atît pentru sistemul nostru cît și pentru rețelele de calculatoare, și totodată care nu au permis ca acestea să <sup>se</sup> poată constitui într-o metodă generală și eficientă de analiză și calcul, sînt următoarele:

- Rețelele de calculatoare sînt complexe în special din punct de vedere al fenomenelor de așteptare, și prin urmare al modelării matematice a acestora. Nu au fost cuprinse, astfel, o serie de caracteristici de bază ale rețelelor de calculatoare cum ar fi, capacitatea limitată a resurselor, blocajul calculatoarelor la supraîncărcare, protocoalele de transmitere a mesajelor și de comunicație între calculatoare, întîrzierea mesajelor în rețea, influența timpilor de retransmitere asupra vitezei de prelucrare și altele.

- Lucrările menționate au uneori un caracter pur matematic cu o abordare teoretică și de natură generală fă-

ră posibilitate de valorificare practică, iar altele acestea cuprind rezolvări simpliste și care se referă la cazuri particulare.

- Aparatul matematic folosit în aceste lucrări, este foarte complicat, nesistematizat, neomogen, diversificat de la autor la autor, utilizează fie metode vechi, rigide care pot oferi soluții numai pentru cazuri de modele simple și particulare cu distribuții de probabilitate constantă sau exponențială, fie metode de aproximare recente dar nematurizate suficient, pentru a oferi soluții satisfăcătoare.

Cu toate acestea, la elaborarea lucrării noastre privind tratarea matematică, am luat în considerare cele mai recente și importante lucrări, studii și cercetări, apărute în domeniu și din care am desprins și reținut atât unele idei interesante legate de concepția de modelare matematică cât și idei privind modalități de rezolvare matematică, (analiza matematică a modelelor, diferite tehnici de aproximare) - în bibliografie (3).

Prezentăm în continuare idei și concepte care ne-au sprijinit în elaborarea metodelor noastre de rezolvare, precum și autorii lor: tratarea teoretică a unui sistem cu stații în tandem și evaluarea teoretică a capacității sistemului cu folosirea unor formule de convergență care se referă la stabilitatea sistemului (D. Hildebrand); determinarea parametrilor modelelor cu șiruri de așteptare în serie, de tip exponențial, cu evaluarea probabilității ca un client care intră în sistem să găsească o stație liberă și, de asemenea, rezolvarea printr-o metodă de aproximare a modelelor cu șiruri de așteptare în serie cu diferite distribuții, cu evidențierea efectelor blocării clienților din cauza capacității limitate a șirurilor (G. Pujolle); prezentarea unor soluții matematice pentru aproximarea unor modele pentru sisteme cu calculatoare (E. Gelenbe); rezolvarea teoretică a unor modele precum și soluțiile pentru modelele clasice de așteptare, tratarea teoretică a modelelor de aproximare (L. Kleinrock); găsirea soluțiilor exacte de tip teoretic pentru modele simple cu 2-3 stații și distribuții constante și exponențiale cu aplicații la multiprogramarea calculatoarelor (B. Avi Itzhak); partea teoretică a aproximării prin difuzie cu soluționarea ecuațiilor de difuzie care stau la baza înlocuirii unui proces stochastic general ca un proces de difuzie de tip continuu markovian (H. Kobayashi); unele aspecte teoretice pri-

vind precizia aproximării prin difuzie aplicată la un sistem cu șiruri de așteptare în serie (M. Reiser) și alți.

Considerăm că tratarea matematică prezentată în această parte a lucrării, constituie o contribuție personală și originală atât din punct de vedere teoretic cât și practic, prin metodele noastre de calculul parametrilor și de evaluare a performanțelor rețelelor de calculatoare. Pentru aceasta a fost necesar un efort susținut de cercetare în vederea elaborării acestor metode care au la bază o concepție cadru proprie deosebită de cea a lucrărilor de specialitate existente, datorită faptului că abordează o problemă mai cuprinzătoare, analizează mai atent fenomenele reale și utilizează un aparat matematic mai adecvat.

Modelele matematice construite și analizate cu ajutorul metodelor noastre au acoperit problema și fenomenele reale ale rețelelor de calculatoare într-o manieră destul de cuprinzătoare.

Astfel, activitățile de bază din cadrul rețelei de calculatoare, respectiv schimburile de mesaje între calculatoare sînt cuprinse în totalitate și reprezentate prin modele cu șiruri de așteptare în serie. O contribuție personală importantă în acest sens a fost aceea de atribuire a unor posibilități de lărgire și dezvoltare a ipotezelor privind generalizarea atât a proceselor de servicii și de sosiri (de la distribuții de probabilitate constante și exponențiale pînă la generale), cât și a numărului de stații în serie (de la două stații pînă la un număr oarecare).

Trăsăturile specifice ale sistemului nostru și a rețelelor de calculatoare în general ca de exemplu capacitatea limitată a tamponelor de memorie, existența unor protocoale de transmitere a mesajelor pentru calculatoare și de confirmarea recepției acestora, întîrzierea mesajelor în rețea și altele, sînt surprinse și reprezentate ca o contribuție deosebită personală în modelele noastre, prin unele caracteristici speciale care introduc fenomene ca: blocarea stațiilor datorită capacității limitate a șirurilor de așteptare, apariția reacției inverse între stații ceea ce duce la lipsa de independență dintre acestea, existența timpilor de retransmisie a clienților și influența acestora asupra încărcării sistemului și altele.

Aparatul matematic constituit de noi și utilizat în aceste metode este cel mai adecvat fiind bine



sistematizat, elastic și actual și oferă posibilități de extindere și generalizare a rezultatelor obținute putând aborda cazuri de sisteme atât complexe, generale cât și particulare.

În lucrare, când a fost cazul, s-au folosit și rezultate din rezolvarea unor modele de așteptare clasice.

Aceste tipuri de modele sînt caracterizate de o serie de componente dintre care fac parte în primul rînd tipurile variabilelor aleatoare care dau timpii de sosire și timpii de serviciu.

Astfel, tipul distribuției timpilor de intersosire sau al distribuției timpului de serviciu pot fi:

- Exponențial, notat cu  $E_1$ ; cu distribuția de tip  $\mu e^{-\mu x}$ ;
- Erlang de ordin  $r$ ,  $E_r$ ;  $r\mu(r\mu)^{r-1} e^{-r\mu x} (r-1)$ ;
- Hiperexponențial de ordin  $R$ ,  $H_R$ ;  $\sum_{i=1}^R \alpha_i \mu_i e^{-\mu_i x}$
- Deterministic,  $D$ ;  $u_0(x-1/\mu)$
- general,  $G$ ; distribuție arbitrară.

Referirile cele mai frecvente le-am făcut la modelele bine cunoscute de tip  $M/M/1$  și  $M/M/1/K$  adică modele cu sosiri Poisson (intersosiri exponențiale) și timpii de serviciu exponențiali, cu un singur serviciu cu disciplină FIFO, primul model de capacitate infinită, iar al doilea de capacitate limitată.

Parametrul esențial ce caracterizează aceste sisteme este intensitatea de trafic (încărcarea sistemului) notată  $\rho$  și exprimată ca  $\rho = \frac{\lambda}{\mu}$ , unde  $\lambda$  este numărul mediu de sosiri în unitatea de timp (rata sosirilor) iar  $\mu$  este numărul mediu de servicii în stație în unitatea de timp (rata serviciului).

În cazurile staționare ale modelelor  $M/M/1$  și  $M/M/1/K$ , adică când probabilitatea  $P_n(t)$  ca în sistem să existe  $n$  clienți la momentul  $t$ , nu depinde de  $t$ , soluțiile sînt:

$$P_n = (1-\rho)\rho^n \quad n \geq 0 \quad \text{pentru } M/M/1$$
$$P_n = \rho^n(1-\rho)/(1-\rho^{K+1}) \quad 0 \leq n \leq K \quad \text{pentru } M/M/1/K.$$

Dintre notațiile pentru termenii probabilistici clasici des utilizați în lucrare, menționăm:

- Media, sau valoarea așteptată a variabilei aleatoare  $x$ , notată cu  $E(x)$ ,
- Variația sau dispersia lui  $x$ , notată  $\sigma_x^2 = \text{Var}(x) = E(x-E(x))^2$
- Coeficientul pătratic de variație a lui  $x$ , notat  $C_x = \sigma_x^2/E^2(x)$

În concluzie, putem afirma că metodele noastre umplu un gol în domeniul tratării matematice a rețelelor de calculatoare, oferind rezultate concrete și satisfăcătoare pentru

o mare gamă de sisteme complexe de teleprelucrare și rețele de calculatoare deoarece au surprins o problemă mai cuprinzătoare în raport cu lucrările de specialitate anterior prezentate, au eliminat unele din deficiențele lor și au preluat constructiv ideile valoroase din acestea; rezultatele noastre pot fi verificate cu ușurință și cu succes în practică.

### 3.2. CONCEPTIA DE MODELARE MATEMATICA

Tipul general de model matematic stochastic care reprezintă legătura între calculatoare, din cadrul unei rețele de calculatoare, respectiv sistemul nostru, pe care îl analizăm și rezolvăm în continuare este prezentat în fig. 3.1. și are următoarele trăsături:

- Structură cu stații în serie - tandem;
- Clienții circulă de la prima stație până la ultima stație, ieșirea din prima stație fiind intrarea pentru cea de-a doua stație și așa mai departe și suportă câte un serviciu în fiecare stație conform principiului primul sosit, primul servit;
- Capacitatea de stocare a stațiilor este limitată, ceea ce înseamnă că există șiruri de așteptare intermediare la fiecare stație cu un număr finit de clienți, mai puțin prima stație despre care se presupune că poate avea un număr oricât de mare de clienți (această stație are rolul unui calculator privilegiat - gazdă, cu capacitate de memorare foarte mare și în care nu au loc pierderi de mesaje - pachete de date).

În momentul când un client termină serviciul și găsește șirul de așteptare plin, al stației următoare, atunci acesta așteaptă, este blocat în stația în care a fost servit, până când se eliberează un loc în șirul de așteptare al stației următoare;

- Reacția inversă care apare între stații datorită blocării cauzate de umplerea șirurilor de așteptare intermediare finite, influențează timpii de serviciu ai stațiilor, timpii de serviciu ai buclei de reacție vor fi reflectați prin creșterea timpului de inutilizare al clientului datorită

blocării. Rezultă astfel, o interdependență a fenomenelor de așteptare care se reflectă prin faptul că, timpii succesivi între intrările în prima stație și la fel, timpii de serviciu ai stațiilor nu vor fi independenți și identic distribuiți (presupunerea ca acești timpii să fie independenți ar fi facilitat mult rezolvarea modelului);

- Retransmisia clientului la stația precedentă apărută ca urmare a fenomenului de reacție conduce la faptul că dacă o stație este plină, serviciul precedent este blocat fie imediat, fie la terminarea serviciului. Există mai multe modalități de retransmisie în funcție de regulile de gestiune a mișcării clienților în sistem (corespondente protocoalelor din cadrul rețelelor de calculatoare) și care se reflectă prin blocarea clientului (imediat sau la terminarea serviciului), prin pierderea acestuia în rețea, prin reemiterea lui de la stația precedentă și prin retransmisie de la stația emitentă (prima stație).

Complexitatea fenomenelor de așteptare din modelul nostru datorată numărului oarecare de stații în serie, cu distribuții frecvent, generale și unor restricții cum ar fi limitările de capacitate și existența unor reguli de mișcare a clienților în sistem face ca tratarea matematică prin scrierea ecuațiilor de echilibru cu aparatul matematic existent să fie foarte greoaie pentru cazurile simple cu ipoteze simplificatoare și chiar imposibile pentru situații mai complexe, reale, cum ar fi în cazul unei legături între calculatoare din cadrul rețelelor. Pentru rezolvarea matematică a modelelor noastre am încercat pe baza unor studii proprii să găsim o soluție la această problemă, respectiv o tehnică nouă de abordare teoretică, posibilă și eficientă.

Ca urmare am procedat la elaborarea unui concept cadru teoretic de modelare care să surprindă fenomenele reale și structura sistemului, concept pe care l-am dezvoltat și materializat cu ajutorul unui aparat matematic adecvat rezultând astfel cele două metode de rezolvare, analitică și prin aproximare.

Conceptul cadru are la bază ca o cerință fundamentală, determinarea parametrului  $\rho_{max}$ , parametru legat de capacitatea sistemului, în condiții de stabilitate. Cu alte cuvinte urmărăm o rată maximă de intrare a clienților în sistem, care poate fi controlată sau gestionată de acesta, astfel ca timpii de așteptare să rămână finiți cu probabilitatea unu.

Această rată maximă a intrărilor în sistem este o măsură a capacității sistemului deoarece media timpilor dintre intrări (media timpilor de ocupare ai primei stații) reprezintă

componenta de serviciu a parametrului de intensitate de trafic  $\rho_{\max}$ .

Raționamentul de bază pe care l-am făcut pentru soluționarea problemei menționate mai sus pleacă de la observațiile că lungimea șirului inițial de așteptare despre care presupunem că are în permanență clienți, crește și descrește în funcție de timpii cât clienți se află în prima stație, adică ocupă stația, (timpii de serviciu plus timpii de blocare). Astfel, pe medie, va exista o creștere a șirului de așteptare dacă timpii dintre sosiri (inter sosirile) sînt mai mici decît timpii dintre intrările clienților în prima stație și va fi o descreștere a șirului de așteptare, dacă timpii dintre sosiri sînt mai mari decît timpii dintre intrările în prima stație.

De asenenea conceptul se bazează în plus, pe o cerință specifică care se referă la capacitatea limitată a stațiilor cu apariția reacțiilor și la existența unor retransmisii supuse regulilor de mișcare a clienților în sistem.

Pentru rezolvarea acestei probleme, care ne-a cauzat dificultăți mari în rezolvarea sistemului, am găsit o soluție proprie care constă în eliminarea efectului reacțiilor dintre stații deci a interdependenței dintre acestea și am procedat la transformarea printr-o tehnică matematică proprie a sistemului de așteptare în serie cu capacitate limitată într-un sistem de așteptare în serie echivalent în care stațiile să fie independente, dar cu procegele stochastice de sosire și servicii modificate corespunzător și astfel să le putem asimila cu modele clasice de așteptare cu o singură stație și a căror soluții sînt bine cunoscute. Această tehnică de rezolvare prin care nu mai ținem seama de reacția de respingere dintre stații ci numai de relația de succesiune dintre ele, constă în esență în modelarea fenomenului de respingere datorat blocării prin introducerea anumitor probabilități de respingere a clienților la fiecare stație și introducerea lor ca factori de influență ai serviciului stației precedente care astfel se modifică corespunzător. Să remarcăm că dacă în rezolvarea aceasta am fi presupus că șirurile de așteptare ale stațiilor ar fi avut capacitate infinită atunci stațiile puteau fi considerate de la început independente și în acest caz rezolvarea modelului nu ar fi ridicat probleme deosebite.

### 3.3. METODA DE CALCUL ANALITICĂ, EXACTĂ.

Metoda de calcul analitică are la bază conceptul cadru de modelare matematică, dezvoltat cu ajutorul unui aparat matematic cu un caracter analitic, de algebră simplă și probabilități. Prin acest aparat matematic am putut aborda numai modele cu șiruri de așteptare în serie, cu un număr mic de stații și de tip exponențial, deci numai presupunerile de cazuri simple ale sistemului nostru, care rămân în limite de prelucrare matematică abordabilă, cu obținerea prin calcul, de soluții exacte și explicite.

Scopul pe care ni l-am propus prin această metodă de calcul a fost determinarea ratei maxime a intrărilor în sistem adică numărul maxim de clienți care trec prin sistem în unitatea de timp, ca o măsură a capacității sistemului.

În studiul pe care l-am elaborat, am plecat de la cele stabilite anterior, în conceptul cadru de modelare, adică de la ideea determinării ratei maxime a intrărilor într-un sistem stabil cu probabilități structurale definite (stații în serie cu presupunerea că timpurile de serviciu și de intersosire sînt independenți).

Am demonstrat matematic că determinarea capacității sistemului constă în evaluarea timpului mediu de ocupare al primei stații. Luînd în considerare variabilele aleatoare care dau momentele principale de stare ale sistemului, în care clienții intră în sistem, trec dintr-un șir de așteptare în stație și invers și derivînd din ele intervalele pentru timpuri în care o stație așteaptă sosirea unui client, fiind liberă, în serviciu, sau blocată, am putut demonstra unele formule pentru evaluarea mediei timpurilor de serviciu plus cei de blocare ai primei stații în funcție de timpurile de serviciu și de așteptare ai stației următoare. Rezultatul principal pe care l-am stabilit este că rata maximă a intrărilor în sistem care caracterizează capacitatea sistemului se poate exprima ca inversul mediei timpurilor de serviciu plus de blocare ai primei stații.

Aceste rezultate teoretice obținute de noi pe cale algebrică nu se pot concretiza efectiv din punctul de vedere al calculului numeric datorită existenței multor necunoscute neexplicitate din sistemul de ecuații. Pentru o rezolvare mai convenabilă a modelelor noastre, studiile pe care le-am întreprins în continuare ne-au condus la soluționări efectiv posibile, explicite prin trecerea la anumite evaluări probabilistice care ne oferă posibilități mai mari de generalizare (dar totuși limitate, numai pentru cazurile de tip exponențial). Astfel, am reușit ca printr-un artificiu

matematic, plecînd de la formula teoretică stabilită referitor la timpii de ocupare ai primei stații, să o explicităm convenabil prin evaluarea probabilității ca în momentul în care un client sosește în sistem să existe o stație liberă după un număr de stații, începînd cu prima, succesiv ocupate.

Aceste rezultate cu un caracter mai aplicativ, le-am putut finaliza printr-o concretizare de ordin numeric pe modele cu număr mic de stații în serie identice din punct de vedere al serviciilor și cu distribuții de tip exponențial pentru timpii de serviciu. Cu toate că în aceste cazuri relativ simplificate calculele au fost nebanale, se poate observa că pentru modelele cu mai multe stații în serie chiar identice și de tip exponențial, calculele devin deosebit de dificile.

Incepem studiul nostru referitor la rezolvarea analitică a modelului prin utilizarea unui artificiu pentru a elimina șirurile de așteptare intermediare. Astfel, considerînd procesul de mișcare al unui client dintr-un loc în următorul loc din șirul de așteptare sau în stație, ca un serviciu cu timp de serviciu nul (Avi Itzhak) putem privi sistemul ca avînd stații consecutive fără șiruri de așteptare intermediare. Deci, în continuare vom considera un model cu  $m$  stații în serie de capacitate 1. În vederea rezolvării matematice definim momentele în care clienții intră în sistem, trec din șirul de așteptare în stație sau trec din stație în șirul de așteptare. Astfel notăm:

$T_n$  - timpul dintre sosirile clienților  $n-1$  și  $n$ ,

$S_{k,n}$  - timpul de serviciu al clientului  $n$  în serviciul  $k$ ,

$T_{k,n}$  - momentul în care clientul  $n$  părăsește serviciul  $k$  și intră în serviciul  $k+1$ ,

$T_{0,n}$  - momentul intrării clientului  $n$  în primul serviciu,

$R_n = T_1 + T_2 + \dots + T_n$  - momentul sosirii clientului  $n$ .

Pentru condițiile inițiale luăm o secvență de variabile aleatoare, independente de  $T_{k,n}$  și care, dacă sistemul este gol,  $T_{0,0} = T_{1,0} = \dots = T_{m,0} = 0$

Evident din semnificațiile variabilelor introduse.

$T_{k,n}$  se pot exprima recursiv prin:

$$T_{0,n} = \max(R_n, T_{1,n-1}), \quad (1)$$

$$T_{k,n} = \max(T_{k-1,n} + S_{k,n}, T_{k+1,n-1}), \quad (k = 1, \dots, m-1), \quad (2)$$

$$T_{m,n} = T_{m-1,n} + S_{m,n} \quad (3)$$

adică, clientul  $n$  intră în prima stație, la cel mai târziu dintre momentul sosirii sale la stație și momentul intrării clientului  $n-1$

în stația a doua; clientul n intră în stația k+1 după ce a intrat și a terminat serviciul în stația k și clientul n-1 a părăsit stația k+1; la stația m, clientul n intră după ce și-a terminat serviciul la stația m-1 iar clientul precedent a părăsit sistemul.

Definim în continuare:

- Timpul de așteptare al clientului n în șirul inițial:

$$W_n = T_{0,n} - R_n, \quad (4)$$

-- Timpul de blocare al clientului n în serviciul k:

$$B_{k,n} = T_{k,n} - T_{k-1,n} - S_{k,n} \quad (5)$$

- Timpul în care serviciul k este liber, în așteptarea clientului n:

$$V_{k,n} = T_{k-1,n} - T_{k,n-1}, \quad (6)$$

și eliminăm din relațiile de mai sus variabilele T (care sînt momente de timp) pentru a lucra numai cu variabilele care reprezintă intervalele de timp.

Astfel, obținem pentru  $W_n$ :

$$W_n = T_{0,n} - R_n = \max(R_n, T_{1,n-1}) - R_n = \max(0, T_{1,n-1} - R_n) = \max(0, T_{1,n-1} - R_{n-1} - T_n) = \max(0, B_{1,n-1} + S_{1,n-1} + T_{0,n-1} - T_{0,n-1} - T_n),$$

deci,

$$W_n = \max(0, W_{n-1} + S_{1,n-1} + B_{1,n-1} - T_n); \quad k = 1, 2, \dots, m-1. \quad (7)$$

Calculînd  $S_{k,n} + B_{k,n}$  obținem:

$$S_{k,n} + B_{k,n} = T_{k,n} - T_{k-1,n} = \max(T_{k-1,n} + S_{k,n}, T_{k+1,n-1}) - \max(T_{k-2,n} + S_{k-1,n}, T_{k,n-1}) = \max(T_{k-1,n} + T_{k,n} - T_{k-1,n}, T_{k+1,n-1}) - \max(T_{k-2,n} + T_{k-1,n} - T_{k-2,n}, T_{k,n-1})$$

și atunci:

$$S_{k,n} + B_{k,n} = \max(S_{k,n}, S_{k+1,n-1} + B_{k+1,n-1} - V_{k,n}), \quad k = 1, 2, \dots, m-1 \quad (8)$$

Căutăm de asemenea și pentru timpii  $V_{k,n}$  o expresie recursivă, dar tot fără intervenția variabilelor  $T_{k,n}$ .

$$V_{1,n} = T_{0,n} - T_{1,n-1} = \max(R_n, T_{1,n-1}) - T_{1,n-1} = \max(0, R_n - T_{1,n-1})$$

deci:

$$V_{1,n} = \max(0, T_n - W_{n-1} - S_{1,n-1} - B_{1,n-1}) \quad (9)$$

$$V_{k,n} = T_{k-1,n} - T_{k,n-1} = \max(T_{k-2,n} + S_{k-1,n}, T_{k,n-1}) - T_{k,n-1} = \max(T_{k-2,n} + S_{k-1,n} - T_{k,n-1}, 0)$$

și atunci:

$$V_{k,n} = \max(0, V_{k-1,n} + S_{k-1,n} - S_{k,n-1} - B_{k,n-1}), \quad k = 2, 3, \dots, m \quad (10)$$

Înlocuind în (8) expresia de la (10) și folosind proprietățile elementare ale operatorului max obținem:

$$\begin{aligned} S_{k,n} + B_{k,n} &= \max(S_{k,n}, S_{k+1,n-1} + B_{k+1,n-1} - V_{k,n}) = \max(S_{k,n}, \\ &\max(S_{k+1,n-1}, S_{k+2,n-2} + B_{k+2,n-2} - V_{k+1,n-1}) - V_{k,n}) = \max(S_{k,n}, \\ &\max(S_{k+1,n-1} - V_{k,n}, S_{k+2,n-2} + B_{k+2,n-2} - V_{k+1,n-1} - V_{k,n})) \end{aligned}$$

deci în final:

$$S_{k,n} + B_{k,n} = \max(S_{k,n}, S_{k+1,n-1} - V_{k,n}, \dots, S_{m,n-m+k} - V_{k,n} - \dots - V_{m-1,n-m+k+1}) \quad (11)$$

unde am folosit că  $B_{m,k} = 0$ , ceea ce corespunde faptului evident că ultima stație nu are timpi de blocare.

Din observarea fenomenelor reale, sîntem conduși la studierea modelelor cu șiruri inițiale de așteptare lungi, deci cu timpi de așteptare foarte mari și prin urmare folosim în continuare ipoteza că timpul în care serviciul la prima stație este liber între oricare doi clienți consecutivi este nul ( $V_{1,k} = 0$ ). Deci prima stație fiind mereu ocupată, înseamnă că pe medie, timpii de intrare în prima stație, deci timpii în care sistemul primește clienți pentru a-i servi sînt dați de timpii de ocupare ai primei stații. Rezultă că timpul mediu de ocupare al primei stații  $E(S_{1,n} + B_{1,n})$  joacă rolul de timp de serviciu al sistemului (timp de "trecere" prin sistem). Se remarcă că  $E(S_{1,n} + B_{1,n})$  nu reprezintă media timpului total de serviciu în sistem pentru un client, nici media timpului petrecut (prin servire sau blocare) de client în sistem, ci reprezintă media timpului în care sistemul poate conduce, gestiona un anumit client în sensul că poate permite apoi intrarea altui client în sistem.

În ceea ce privește analiza variației lungimii șirului inițial de așteptare, observăm că vom avea o creștere dacă media timpilor de intersosire  $E(T_n)$  este mai mică decît media timpilor dintre intrări în sistem  $E(S_{1,n} + B_{1,n})$  și vom avea o descreștere, dacă media timpilor de intersosire  $E(T_n)$  este mai mare decît media timpilor dintre intrări  $E(S_{1,n} + B_{1,n})$ , adică

$$\begin{aligned} E(T_n) &< E(S_{1,n} + B_{1,n}) \text{ și respectiv:} \\ E(T_n) &> E(S_{1,n} + B_{1,n}). \end{aligned} \quad (12)$$

Evident, tendința de creștere sau descreștere a șirului inițial de așteptare semnifică starea de stabilitate, respectiv instabilitate a șirului de așteptare; așadar raportul

$$\lim_{n \rightarrow \infty} \frac{E(S_{1,n} + B_{1,n})}{E(T_n)} \text{ ne indică starea de stabilitate sau in-}$$



stabilitate a sistemului, după cum;  $\lim_{n \rightarrow \infty} \frac{E(S_{1,n} + B_{1,n})}{E(T_n)} < 1$ , respectiv

$$\lim_{n \rightarrow \infty} \frac{E(S_{1,n} + B_{1,n})}{E(T_n)} > 1.$$

În plus, ca urmare a analogiei pe care o putem face între modelul nostru și un model clasic de așteptare cu o stație (putem privi sistemul nostru ca pe un sistem de așteptare format dintr-un flux de intrare, o unitate de servicii multiple și un flux de ieșire) rezultă că parametrul de intensitate a traficului pentru modelul nostru este tocmai raportul expresiilor din relațiile (12) adică la limită:

$$\rho = \lim_{n \rightarrow \infty} \frac{E(S_{1,n} + B_{1,n})}{E(T_n)} \quad (13)$$

În concluzie, folosind o exprimare matematică și notațiile introduse mai înainte se poate spune că, dacă:  $\rho > 1$ , atunci  $\lim_{n \rightarrow \infty} P(W_n \leq t) = 0$  și sistemul de așteptare este instabil, iar dacă:

$\rho < 1$ , atunci există o anumită funcție de repartiție notată cu  $F(t)$ , astfel încât  $\lim_{n \rightarrow \infty} P(W_n \leq t) = F(t)$  și în acest caz sistemul este stabil.

Pentru modele teoretice de așteptare care au unele caracteristici comune cu modelul nostru, Hildebrand demonstrează prin anumite tehnici matematice complicate că unii vectori aleatori (timpul de așteptare, timpul liber al stației) sînt fie procese aleatoare Markov cu probabilități de trecere staționare, fie converg la o limită finită - rezultate care au constituit un sprijin în confirmarea propriilor noastre rezultate.

Rezultă, în finalică numărul:

$$1 / \lim_{n \rightarrow \infty} E(S_{1,n} + B_{1,n}) \quad (14)$$

este rata maximă a intrărilor în sistem care poate fi controlată de un sistem dat, deci aceasta este capacitatea de trecere a sistemului. Evaluarea cantității  $\lim_{n \rightarrow \infty} E(S_{1,n} + B_{1,n})$  fiind foarte importantă, orice metodă de determinare a ei devine o metodă pentru determinarea parametrului  $\rho$ .

Pentru determinarea timpului de ocupare al primei stații, în situația de echilibru, la limita  $n \rightarrow \infty$ , utilizăm formula (11) care devine astfel:  $S_1 + B_1 = \max(S_1, S_2, S_3 - V_2, S_4 - V_2 - V_3, \dots, S_m - V_2 - V_3 - \dots - V_{m-1})$

unde deci am notat cu  $S_k$  variabila aleatoare care dă serviciul în stația  $k$ , iar celelalte variabile aleatoare cu semnificația corespunzătoare.

Rezultatele obținute pentru cazul general arătate mai sus stau la baza determinării capacității pentru anumite cazuri particulare dar care sînt totuși limitate.

Un caz particular pe care l-am soluționat ușor este al unui sistem cu numai două stații  $m = 2$ . Pentru aceasta avem:

$$S_{1,n} + B_{1,n} = \max(S_{1,n}, S_{2,n-1}) \text{ și evident}$$

$$\lim E(S_{1,n} + B_{1,n}) = E(\max(S_{1,n}, S_{2,n-1})). \quad (16)$$

Deci, media timpilor de serviciu și de blocare, suma cărora formează timpul ocupat pentru prima stație, este egală cu media maximului dintre timpii de serviciu ai celor două stații.

Dacă notăm cu  $\lambda$  rata sosirilor în sistem, respectiv numărul mediu de clienți care sosesc în sistem în unitatea de timp, atunci pentru a avea îndeplinită condiția de stabilitate a sistemului, rata  $\lambda$  a sosirilor pe care sistemul poate să le gestioneze va trebui să satisfacă  $\lambda E(S_1 + B_1) < 1$ .

Rezultă că rata maximă a sosirilor este:

$\lambda_{\max} = 1/E(S_1 + B_1)$  și deci, în final, parametrul căutat, intensitatea de trafic maximă este:

$$\rho_{\max} = \lambda / E(\max(S_1, S_2)) \quad (17)$$

Un alt caz particular studiat de noi este al unui sistem în care timpul de serviciu al unei stații  $k$  este întotdeauna cel mai mare  $S_{k,n} \cong S_{i,n}$ ,  $i \neq k$ , cu probabilitatea 1; se vede atunci că  $P(V_{2,n} = \dots = V_{k,n-k+2} = 0) = 1$  în echilibru și că:

$$\lim E(S_{1,n} + B_{1,n}) = E(S_{k,n}). \quad (18)$$

Adică, media timpului de ocupare a primei stații este egală cu media timpului de serviciu al stației cu cel mai mare timp de serviciu. Deci, luînd în considerare rata sosirilor în sistem,  $\lambda$ , atunci rezultă că:

$$\rho_{\max} = \lambda / E(S_k) \quad (19)$$

În continuare, față de aceste rezolvări banale, imediate, am abordat cazul modelelor cu un număr mic de stații și cu distribuție de tip exponențial pentru timpii de serviciu ai stațiilor. Pentru a evalua timpul de ocupare al primei stații am plecat de la rezultatele obținute pentru cazul general prezentat anterior, formula (15), și am încercat să construim un cadru de calcul care să ne permită eliminarea timpilor de așteptare necunoscute ai stațiilor.

Astfel luînd în considerare formula (15) punem  $p_2 = \text{prob}(S_3 > V_2)$ ,  $p_3 = \text{prob}(S_4 > V_2 + V_3)$ ,  $p_{m-1} = \text{prob}(S_k > V_2 + \dots + V_{m-1})$ . și recurgem în continuare la unele evaluări

probabilistice. Prin urmare, interpretând probabilitățile  $p_2, p_3, \dots, p_{m-1}$ , să observăm că;  $p_2$  este probabilitatea ca în momentul în care un client intră în prima stație, stația a doua să fie ocupată;  $p_3$  este probabilitatea ca în momentul în care un client intră în prima stație, stația a treia să fie ocupată, ș.a.m.d.

Rezultă că timpul de ocupare al unui client în prima stație, adică din momentul în care intră în serviciu și pînă cînd părăsește stația,  $S_1+B_1$ , poate fi evaluat în funcție de probabilitățile  $p_k$  în felul următor:

- Dacă prima stație este ocupată în momentul în care a doua se eliberează atunci, timpul de ocupare în prima stație,  $S_1+B_1$  este egal cu  $\max(S_1, S_2)$ , iar probabilitatea să avem aceasta este  $(1-p_2)$ .

- Dacă primele două stații sînt ocupate în momentul în care a treia stație se eliberează atunci, timpul de ocupare  $S_1+B_1$  este egal cu  $\max(S_1, S_2, S_3)$  și aceasta se întîmplă cu probabilitatea  $p_2(1-p_3)$ .

- Continuînd, deducem că dacă toate stațiile dinaintea stației  $k$  sînt ocupate în momentul în care aceasta se eliberează, rezultă că  $S_1+B_1 = \max(S_1, S_2, \dots, S_k)$  și aceasta se întîmplă cu probabilitatea  $p_2 \cdot p_3 \cdot \dots \cdot p_{k-1} (1-p_k)$ .

Notăm variabila aleatoare  $\max(S_1, S_2, \dots, S_k)$  cu  $M_k$  și obținem în final descompunerea expresiei  $S_1+B_1$  ținînd cont și de faptul că legea exponențială nu are memorie; "

$$S_1+B_1 = M_2(1-p_2) + M_3 p_2(1-p_3) + \dots + M_{m-1} p_2 p_3 \dots p_{m-2} (1-p_{m-1}) + M_m p_2 p_3 \dots p_{m-1}, \quad (20)$$

Produsul  $p_2 p_3 \dots (1-p_k)$  reprezintă probabilitatea ca un client intrînd în primul șir de așteptare să aibă înaintea lui  $k$  stații ocupate și stația  $(k+1)$  să fie liberă.

Egalitatea (20) de mai sus trebuie înțeleasă în sensul că dacă toate stațiile sînt ocupate pînă la stația  $(k+1)$  care este liberă, atunci clientul va rămîne în prima stație un timp egal cu  $\max(S_1, S_2, \dots, S_k)$ .

În continuare, trecem la exemplificarea rezultatelor obținute cu formula (20). Ca un exemplu particular calculăm  $\rho_{\max}$  pentru cazul a trei stații identice cu serviciul exponențial. Din relațiile (15) și (20) pentru  $m = 3$  obținem:

$$S_1+B_1 = \max(S_1, S_2, S_3 - V_2) = \max(S_1, S_2, S_3) p_2 + \max(S_1, S_2) \cdot (1-p_2) \quad (21)$$

și apoi:

$$(1-p_2) = p_2 \text{prob}(S_1 > \max(S_2, S_3) + S_3) + (1-p_2) \text{prob}(S_1 > S_2 + S_3) \quad (22)$$

Avem deci de calculat funcțiile de repartiție ale variabilelor aleatoare  $(S_2+S_3-S_1)$  și  $(\max(S_2, S_3)+S_3-S_1)$ . Notînd cu  $e_u$  densitatea de repartiție a unei variabile aleatoare de tip exponențial negativ, adică  $e_u(x) = \mu e^{-\mu x}$ ,  $\mu > 0$ ,  $0 < x < \infty$  și cu  $\Theta$  semnul de convoluție, adică de compunere a două funcții de repartiție corespunzătoare a două variabile aleatoare independente și ținînd seama de faptul că  $S_1, S_2, S_3$  sînt variabile aleatoare independente putem scrie:

$$\begin{aligned} \text{prob}(S_1 > S_2 + S_3) &= (e_{\mu_1} \Theta e_{\mu_2} \Theta e_{\mu_3}) (\{(\alpha_1, \alpha_2, \alpha_3) \mid \alpha_1 > \alpha_2 + \alpha_3\}) = \\ &= \mu_1 \mu_2 \mu_3 \iiint_{\substack{(\alpha_1, \alpha_2, \alpha_3) \in \mathbb{R}^3 \\ \alpha_2 + \alpha_3 < \alpha_1}} e^{-\mu_1 \alpha_1} e^{-\mu_2 \alpha_2} e^{-\mu_3 \alpha_3} dx_1 dx_2 dx_3 \\ \text{prob}(S_1 > \max((S_2, S_3) + S_3)) &= (e_{\mu_1} \Theta e_{\mu_2} \Theta e_{\mu_3}) (\{(\alpha_1, \alpha_2, \alpha_3) \mid \\ \max(\alpha_2, \alpha_3) + \alpha_3 < \alpha_1\}) &= (e_{\mu_1} \Theta e_{\mu_2} \Theta e_{\mu_3}) (\{(\alpha_1, \alpha_2, \alpha_3) \mid \\ \alpha_2 \leq \alpha_3\}) &+ (e_{\mu_1} \Theta e_{\mu_2} \Theta e_{\mu_3}) (\{(\alpha_1, \alpha_2, \alpha_3) \mid 2\alpha_3 < \alpha_1 \text{ și } \alpha_2 \leq \alpha_3\}) = \\ &= \mu_1 \mu_2 \mu_3 \iiint_{\substack{(\alpha_1, \alpha_2, \alpha_3) \in \mathbb{R}^3 \\ \max(\alpha_2, \alpha_3) + \alpha_3 < \alpha_1}} e^{-\mu_1 \alpha_1} e^{-\mu_2 \alpha_2} e^{-\mu_3 \alpha_3} dx_1 dx_2 dx_3 \end{aligned}$$

După calcularea integralelor de mai sus, înlocuind în (22) obținem:

$$1 - p_2 = p_2 \left( \frac{\mu_2}{\mu_1 + \mu_2} - \frac{\mu_1 + \mu_2}{2\mu_1 + \mu_2} \right) \frac{\mu_3}{\mu_1 + \mu_3} + (1 - p_2) \frac{\mu_2 \mu_3}{(\mu_1 + \mu_2)(\mu_1 + \mu_2)} \quad (23)$$

Deoarece am presupus stațiile identice, punem  $\mu_1 = \mu_2 = \mu_3$  în (23) și după simplificări găsim  $p_2 = \frac{2}{11}$ .

Pentru a putea calcula  $\rho_{\max}$ , avem nevoie de  $E(S_1 + B_1)$  și deci, după cum se vede în (21), de  $p_2$ , de  $E(\max(S_1, S_2))$  și de  $E(\max(S_1, S_2, S_3))$ .

Dacă  $s_k$  este densitatea de probabilitate a variabilei  $S_k$ , adică  $s_k(x) = \mu e^{-\mu x}$  iar  $F_k$  este distribuția de probabilitate a lui  $S_k$ , atunci  $E(R_m)$  este:

$$\begin{aligned} E(R_m) &= E(\max(S_1, S_2, \dots, S_m)) = \int_0^{\infty} x \sum_{k=1}^m (s_k(x) \prod_{j \neq k} F_j(x)) dx = \\ &= \frac{1}{\mu} \left( 1 + \frac{1}{2} + \dots + \frac{1}{m} \right). \end{aligned}$$

Atunci înlocuim în (21) și obținem:

$$\begin{aligned} E(S_1 + B_1) &= E(R_3) p_2 + E(R_2) (1 - p_2) = \frac{1}{\mu} \left( \left( 1 + \frac{1}{2} + \frac{1}{3} \right) \cdot \frac{2}{11} + \left( 1 + \frac{1}{2} \right) \cdot \frac{9}{11} \right) = \\ &= \frac{1}{\mu} \frac{30}{22} \end{aligned}$$

adică  $\rho_{\max} = \frac{22}{39}$

Neexistând nici o formulă recursivă pentru deducerea lui  $\rho_{\max}$  pentru 4 și 5 stații în serie, calculele trebuie să se facă de la capăt și sînt foarte complicate. Menționăm că din calculele pentru 4 stații, se obține  $\rho_{\max} = 0,515$  iar pentru 5 stații  $\rho_{\max} = 0,436$  (G. Fajolle).

Se observă că încărcarea maximă a sistemului descrește pe măsură ce crește numărul stațiilor în serie - fenomen normal și ușor de intuit, dar este important de știut că ea nu scade sub o valoare finită nenulă.

### 3.4. METODE DE CALCUL PRIN APROXIMARE

Am abordat rezolvarea prin tehnici matematice de aproximare, a modelelor noastre, care are în vedere restricțiile și limitările metodei analitice propusă anterior; am elaborat o metodă de calcul denumită în lucrare, metoda prin aproximare, care este o modalitate eficientă și practică de tratare matematică a modelelor complexe pentru determinarea parametrilor acestora și deci a performanțelor rețelelor de calculatoare. Această metodă constituie în ansamblul ei o contribuție personală cu caracter teoretic și practic în domeniul tratării matematice a sistemelor de teleprelucrarea datelor și rețelelor de calculatoare.

Considerăm că utilizarea teoriei așteptării pentru rezolvarea modelelor complexe reprezentînd rețelele de calculatoare, a crescut enorm prin introducerea acestor tehnici de aproximare matematică, deoarece putem lua în calcul orice formă de distribuție de probabilitate pentru procesele stochastice (de sosire și de serviciu) iar soluțiile obținute sînt suficient de bune comparativ cu modelele de tip exponențial folosite pînă în prezent în domeniul calculatoarelor și care forțau în măsură foarte mare realitatea.

Tehnicile de aproximare elaborate în această lucrare, prin procese de difuzie și prin echivalență sînt cele mai adecvate pentru satisfacerea cerințelor rețelelor de calculatoare. Teoria proceselor de difuzie este aplicată în cazul nostru datorită analogiei fenomenelor de curgere care au loc în interiorul acestor sisteme (cum este în cadrul rețelelor transmiterea mesajelor de la un calculator la altul) ceea ce ne-a permis să procedăm la înlocuirea proceselor stochastice complicate cu procese cu soluții matematice simple de tip continuu

Markov.

Teoria echivalenței am utilizat-o ca o modalitate de rezolvare a unor fenomene complexe din cadrul modelelor noastre ceea ce ne-a permis eliminarea reacției între stații prin echivalarea modelului inițial cu un model cu stații independente de tip exponențial, cu soluții simple.

Prin urmare, am dezvoltat o serie de cercetări, mai mult teoretice, plecând de la unele lucrări matematice recente introducând aceste teorii moderne pur matematice la condițiile noastre concrete, elaborând în acest fel tehnici proprii de aproximare prin procese de difuzie și prin echivalențe care s-au constituit în părți componente ale metodei noastre de calcul prin aproximare; aceste două tehnici au fost, în final, utilizate direct în cadrul unui studiu pentru tratarea matematică a sistemului nostru și extinderea la rețele de calculatoare în condiții de respectare mai fidelă a realității.

Ca urmare, metoda noastră prin aproximare constă din trei mari părți componente:

Un prim studiu privind tehnica de aproximare prin procese de difuzie, care constă în înlocuirea unui proces stochastic complex discret, cu unul simplu de tip continuu Markov; are un caracter mai mult teoretic, iar meritul nostru constă în faptul că am adaptat și dezvoltat ideile teoretice de matematică pură din acest domeniu (Kobayashi privind aproximarea prin difuzie a sistemelor cu șiruri de așteptare în serie și Reiser privind precizia aproximării prin difuzie la sisteme de așteptare) într-o sinteză teoretică proprie conformă cu cerințele noastre, ale rețelelor de calculatoare.

Al doilea studiu privind tehnica de aproximare prin echivalență are un caracter mai aplicativ și pornește de la unele idei ale lui Pujolle. Această tehnică constă în echivalarea modelelor cu fenomene de reacție prin modele mai simple de tip exponențial cu soluții cunoscute.

Cele două tehnici de aproximare cuprinse în studiile noastre pot fi folosite separat sau împreună, ele completându-se de multe ori în cazul sistemelor complexe.

În final, în partea a treia a metodei noastre prin aproximare, făcând apel la cele două părți componente ale sale procese de difuzie și echivalență, este aplicată concret la rezolvarea, în condiții de respectare a realității, a sistemului nostru și a rețelelor de calculatoare.

Metodele de calcul prin aproximare au la

bază evident conceptul cadru de modelare de la care am plecat și pe care l-am dezvoltat prin anumite tehnici matematice de aproximare, putând astfel trata matematic, modelele noastre de așteptare mai complexe (cele mai apropiate de realitate), cu distribuții de tip general și cu un număr oarecare de stații; încercăm să eliminăm și acum efectele reacției între stații, iar procesele cu distribuții generale să le înlocuim cu procese cu distribuții exponențiale, mai simple ce se pot calcula.

Metode de aproximare prin procese de difuzie

Prezentăm în această parte a metodei de aproximare un studiu privind tehnica de aproximare prin difuzie. Am pornit de la analiza unui model cu un singur șir de așteptare cu un singur serviciu, pe care l-am considerat ca element de bază al unui sistem de așteptare complex cu șiruri de așteptare în serie. Considerăm modelul nostru de tip general, adică un sistem de așteptare de forma GI/G/1, cu distribuția timpilor de interesare, de tip general, aceștia fiind independenți între ei, cu distribuția timpului de serviciu de tip general și care dispune de un singur serviciu.

Procesul stochastic discret care în cazul nostru este reprezentat prin fluctuația de lungime a șirului de așteptare, este înlocuit cu un proces continuu Markov, numit și proces de difuzie; distribuția de probabilitate a procesului continuu este descrisă printr-o ecuație de difuzie în care intervine distribuția timpilor de serviciu din modelul inițial care se aproximează și pe care o rezolvăm în anumite condiții de limită.

Notăm cu :

$Q(t)$  - lungimea șirului de așteptare,

$\Delta Q(t)$  - variația lungimii șirului de așteptare între  $t$  și  $\Delta t$ ,

$\lambda$  - rata sosirilor,

$\sigma_a^2$  - varianța sau dispersia timpilor de interesare,

$\mu$  - rata serviciului sau inversul mediei timpului de serviciu,

$\sigma_s^2$  - varianța sau dispersia timpilor de serviciu,

Pentru  $\Delta t$  suficient de mare,  $\Delta Q(t)$  va fi aproximativ normal distribuit și atunci avem, media:

$$E [Q(t)] = (\lambda - \mu)\Delta t = \beta \Delta t, \text{ unde } \beta = \lambda - \mu$$

și varianța:

$$\text{Var. } |Q(t)| = (C_a \lambda + C_s \mu) \Delta t = \alpha \Delta t, \text{ unde } \alpha = C_a \lambda + C_s \mu \quad (2)$$

unde am notat cu:

$C_a$  - co eficientul pătratic de variație pentru timpul de intersosire, și care are următoarea formulă

$$C_a = \text{Var}(t_a) \lambda^2, \quad t_a \text{ fiind timpul de intersosire}$$

$C_s$  - co eficientul pătratic de variație pentru timpul de serviciu

$$C_s = \text{Var}(t_s) \mu^2, \quad \text{unde } t_s \text{ fiind timpul de serviciu.}$$

Presupunerea că șirul de așteptare nu este niciodată gol, și cu celelalte observații anterioare, ne conduc la aproximarea procesului discret  $Q(t)$  printr-un proces continuu  $x(t)$  cu variații  $dx(t)$  distribuite normal cu media  $\beta dt$  și variația  $\alpha dt$ ; procesul  $x(t)$  fiind definit de următoarea ecuație diferențială stochastică prezentată de Kobayashi:

$$dx(t) = \beta dt + z(t)(\alpha dt)^{1/2} \quad (3)$$

unde:

$z(t)$  este un proces gaussian alb cu o medie 0 și o varianță egală cu unitatea.

Dacă nu impunem condiția la limită  $x(t) \geq 0$  atunci procesul  $x(t)$  este un proces Wiener sau, o mișcare browniană care are o densitate de probabilitate  $p(x_0, x; t)$  care satisface ecuația de difuzie a lui Kolmogorov;

$$\frac{\partial p(x_0, x; t)}{\partial t} = \frac{\alpha}{2} \cdot \frac{\partial^2 p(x_0, x; t)}{\partial x^2} - \beta \frac{\partial p(x_0, x; t)}{\partial x} \quad (4)$$

unde  $x_0$  este valoarea inițială și

$$p(x_0, x; t) dx = P \{ x \leq x(t) \leq x+dx | x(0) = x_0 \}$$

Rezolvăm această ecuație de difuzie cu condiția la limită  $x(t) \geq 0$  deoarece este o condiție naturală, lungimea șirului de așteptare nefiind niciodată negativă. Pentru aceasta, în cazul staționar punem la zero derivata în raport cu timpul în ecuația (4) și atunci cerința evidentă;

$$p(x_0, x; \infty) dx = 1$$

ne conduce la principala condiție de stabilitate cunoscută  $\beta < 0$  sau  $\lambda < \mu$ ; cu aceasta obținem soluția în stare staționară a ecuației (4);

$$p(x) = p(x_0, x; \infty) = (2|\beta|/\alpha) \exp(-2|\beta|x/\alpha) = 2(1-\rho)/(C_s + C_a \rho) \exp(-2(1-\rho)x/(C_s + C_a \rho)) \quad (5)$$

unde:

$$\rho = \lambda/\mu < 1$$



Soluția staționară a procesului de difuzie care a înlocuit procesul discret, a fost descrisă cu ajutorul funcției de densitate de probabilitate (5) care este de tip exponențial.

În continuare, ne întoarcem la procesul inițial discret pentru care interpretăm formula (5) ca o distribuție geometrică a șirului de așteptare de mărime variabilă. Obținem o soluție aproximativă pentru distribuția lungimii șirului de așteptare (procesul discret inițial), definind un vector de probabilitate  $\tilde{p}(n)$  prin integrarea lui  $p(x)$  într-un interval unitate,  $n \leq x < n+1$ :

$$\tilde{p}(n) = \int_n^{n+1} p(x) dx = (1-\tilde{\rho})\tilde{\rho}^n, \quad n = 0, 1, 2, \dots \quad (6)$$

unde;

$$\tilde{\rho} = \exp(-2(1-\rho)/(C_s + C_a \rho)). \quad (7)$$

Forma distribuției din (6) ne amintește de cunoscuta formulă de distribuție pentru lungimea șirului de așteptare pentru modelul exponențial M/M/1, cu  $C_a = C_s = 1$ ,

$$p(n) = (1-\rho)\rho^n, \quad n = 0, 1, 2, \dots$$

Vedem că  $\tilde{p}(n)$  din (6) este o foarte bună aproximație pentru  $p(n)$ , deoarece parametrul  $\tilde{\rho}$  din (7) este apropiat de  $\rho$  când  $C_s = C_a = 1$ .

Exactitatea soluției aproximative depinde așa cum am arătat de; aproximarea lui  $Q(t)$  printr-un proces  $x(t)$  continuu Markov, utilizarea condiției la limită în  $x = 0$  și discretizarea funcției de densitate  $p(x)$  într-un vector de probabilitate  $\tilde{p}(n)$ .

Pentru modele cu distribuții generale, cu  $C_a$  și/sau  $C_s$  apreciabil diferiți de unitate, cu alegerea unor condiții la limită deosebite, rezultatul obținut din formula (6) este:

$$\tilde{p}(n) = \begin{cases} 1-\rho, & n=0 \\ \rho(1-\tilde{\rho})\tilde{\rho}^{n-1}, & n \geq 1 \end{cases} \quad (8)$$

Distribuția lungimii șirului de așteptare rezultă că este geometrică pentru  $n \geq 1$ .

Considerăm astfel că aproximațiile prin difuzie pentru procesele cu distribuții generale conduc la reproducerea unui proces discret complex, printr-un alt proces continuu Markov pentru care soluțiile sînt cunoscute.

Tehnica aproximării prin difuzie, așa cum am arătat poate fi utilizată eficient pentru rezolvarea sistemelor cu șiruri de așteptare în serie, cu distribuții de tip general în special în rețelelor deschise care se bazează pe presupunerea că fiecare serviciu poate fi tratat separat (sistemul nostru fiind un exemplu de sistem deschis cu șiruri de așteptare în serie).

Prin această tehnică am ajuns la formulele clasice ale sistemelor cu șiruri de așteptare, evident cu parametri modificați corespunzător.

Metode de aproximare prin echivalență Așa cum am văzut, într-un sistem de așteptare cu stații în serie de capacitate limitată apare fenomenul de blocare a clienților care nu pot trece în următorul șir de așteptare. Conform legilor de mișcare a clienților în sistem, pe care le-am descris în conceptul cadru de modelare, clienții blocați revin la serviciul precedent pe care-l suportă de mai multe ori pînă cînd încercarea lor de trecere în șirul următor are succes. Ca urmare a unor cercetări proprii, noi am soluționat această interdependență a fenomenelor de așteptare dintre stații printr-o tehnică care separă efectul de reacție produs de respingerea clienților, tehnică matematică pe care am numit-o "aproximarea prin echivalență" și care constă în înlocuirea unor modele de așteptare complexe cu fenomene de reacție prin modele cu șiruri de așteptare în serie cu stații independente și cu soluții cunoscute, în mod curent, de tip exponențial. Mai precis, procesele stochastice și serviciile și sosirile suportă modificări corespunzătoare; pentru servicii aceste modificări s-au obținut prin operații matematice de convoluție, care semnifică repetarea serviciului de cîte ori a fost respins clientul.

Am abordat la început un caz mai simplu, cu numai două stații și după ce l-am aproximat prin echivalență, i-am determinat intensitatea maximă de trafic sau încărcarea maximă de trafic  $\rho_{max}$ . În continuare, am trecut la un model mai complex cu  $m$  stații în serie de tip exponențial, pe care l-am putut apoi finaliza calculînd încărcarea maximă a sistemului, făcînd în plus ipoteza că stațiile sînt identice.

Această tehnică de aproximare prin echivalență prezentată printr-un studiu cu caracter teoretic, dar și cu un pronunțat caracter aplicativ este foarte eficientă fiind o componentă de bază a metodei noastre de aproximare; ea constituie totodată o contribuție originală proprie în domeniul tratării matematice a sistemelor de teleprelucrare și a rețelelor de calculatoare.

Am analizat pentru început, cazul modelului cu două stații în serie cu timpi de intersosire și de serviciu de tip exponențial de parametri  $\lambda_1$  și respectiv  $\mu_1, \mu_2$ , cu capaci-

tatea limitată  $M_2$ , la cea de a doua stație. Blocajul se instaurază la prima stație când șirul de așteptare din stația a doua este plin, adică conține  $M_2$  clienți; capacitatea primului șir de așteptare așa cum am presupus pînă acum este  $M_1 = \infty$  iar la al doilea șir de așteptare capacitatea este  $M_2$ .

Pentru rezolvare, studiem modelul ca un sistem format din două stații în serie cu reacție ca în fig. 3.3. Fenomenul de reacție constă în faptul că dacă un client termină serviciul la prima stație și găsește al doilea șir de așteptare plin, atunci el reîncepe serviciul său de la prima stație și aceasta pînă cînd va găsi un loc în stația a doua limitată la  $M_2$  clienți

Deci, cînd a doua stație este plină, clienții care vin sînt respinși, ceea ce dă naștere la o probabilitate de respingere  $p$ . Aproximarea datorită acestei tehnici de echivalență constă în faptul că presupunem  $p$  ca o probabilitate Bernoulli, simplă, căci în realitate această probabilitate depinde de starea sistemului. Rezultă că pentru studierea primului șir de așteptare putem lua o stație echivalentă fără reacție, cu timpul de serviciu dat de densitatea de probabilitate:

$$f(t) = (1-p)\mu_1 e^{-\mu_1 t} + (1-p) \sum_{n=1}^{\infty} p^n E_{n+1} \quad (9)$$

unde;  $\mu_1$  este rata serviciului stației inițiale,  $E_{n+1}$  este distribuția Erlang de ordinul  $n+1$  rezultată prin convoluția de  $n+1$  a distribuției exponențiale de parametru  $\mu_1$ ,  $E_{n+1} p^n (1-p)$  este un termen care reprezintă faptul că clientul efectuează de  $n+1$  ori serviciul cu rata  $\mu_1$  înainte de a putea intra în stația următoare și aceasta se produce cu probabilitatea  $(1-p)p^n$ .

Obținem în continuare prin calcule simple, că noul tip de serviciu este de asemenea exponențial;

$$\begin{aligned} f(t) &= \mu_1 (1-p) (e^{-\mu_1 t} + \sum_{n=1}^{\infty} \frac{(p^n \mu_1^n (n+1)^{n+1} t^n}{(n+1)!} \cdot e^{-\mu_1 (n+1)t}) = \\ &= \mu_1 (1-p) e^{-\mu_1 (1-p)t} \end{aligned} \quad (10)$$

Serviciul echivalent urmează deci, o lege exponențială de rată  $\hat{\mu}_1 = \mu_1 (1-p)$  și deoarece intrările formează un flux Poisson cu rata  $\lambda$  rezultă că prima stație echivalentă este asimilată cu o stație clasică de tip M/M/1.

Notînd cu  $P_i(n)$  probabilitatea ca să existe  $n$  clienți în stația  $i$ ,  $i = 1, 2$ , avem din formulele cunoscute pentru modelul M/M/1:

$$P_1(n) = \hat{\rho}_1^n (1 - \hat{\rho}_1), \text{ unde: } \hat{\rho}_1 = \lambda / \hat{\mu}_1. \quad (11)$$

Pentru studierea celei de a doua stații trebuie să cunoaștem în primul rînd, distribuția intersosirilor, despre care putem arăta că este tot exponențială, p fiind presupusă Bernoulli în modelul nostru echivalent; atunci această stație o vom înlocui deci printr-o stație de tip  $M/M/1/M_2$ . Cum probabilitatea ca un client să fie respins este egală cu probabilitatea ca a doua stație să fie plină, adică  $p = P_2(M_2)$  avem:

$$\begin{cases} p = \hat{\rho}_2^{M_2} (1 - \hat{\rho}_2) / (1 - \hat{\rho}_2^{M_2+1}) \text{ și} \\ \hat{\rho}_2 = \hat{\lambda}_2 / \hat{\mu}_2 = \lambda / \mu_2 (1 - p) \end{cases} \quad (12)$$

Din acest sistem determinăm p și  $\hat{\rho}_2$  implicit și obținem:

$$p = P_2(n) = \hat{\rho}_2^n (1 - \hat{\rho}_2) / (1 - \hat{\rho}_2^{M_2+1}) \quad (13)$$

În ceea ce privește încărcarea maximă pe care o poate suporta sistemul, determinarea căreia constituie scopul lucrării noastre, să remarcăm că în cazul modelului cu două stații pe care-l studiem, încărcarea maximă se obține cînd rata sosirilor la stația a doua este egală cu rata de serviciu a acesteia, adică  $\hat{\lambda}_2 = \hat{\mu}_2$ .

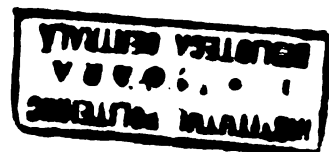
După cum am văzut:

$$p_2 = \hat{\rho}_2^{M_2} \frac{1 - \hat{\rho}_2}{1 - \hat{\rho}_2^{M_2+1}} \text{ deci,}$$

$$p_2 = \lim_{\hat{\lambda}_2 \rightarrow \hat{\mu}_2} \frac{\hat{\rho}_2^{M_2} (1 - \hat{\rho}_2)}{1 - \hat{\rho}_2^{M_2+1}} = \frac{1}{M_2+1}$$

Observînd că  $1 - p_2$  este probabilitatea ca un client servit de prima stație să intre în a doua imediat, înseamnă că încărcarea maximă pe care o poate gestiona sistemul în acest caz particular de 2 stații în serie este tocmai  $1 - p_2$ ; așadar rezultă:

$$\hat{\rho}_{\max} = \frac{M_2}{M_2+1} \quad (14)$$



Tehnica noastră prin echivalență poate fi generalizată pentru studierea comportamentului unui număr oarecare de stații în serie. Modelul pentru cazul general cu  $m$  șiruri de așteptare în serie este prezentat în fig. 3.1. Acest model general îl putem înlocui printr-un model reprezentat de o suită de șiruri de așteptare cu reacție ca în fig. 3.4

Studiind comportamentul unei stații  $i$ , vedem că aceasta se poate înlocui cu o stație cu șir de așteptare, echivalentă, cu următoarele caracteristici:

- rata serviciului ;  $\hat{\mu}_i = \mu_i(1-p_{i+1})$ ;
- rata sosirilor ;  $\hat{\lambda}_i = \lambda/(1-p_i)$

unde;  $p_i$  este probabilitatea de respingere la stația  $i$ ,  $\lambda$  este rata sosirilor în sistem, iar  $\mu_i$  este rata serviciului din stația  $i$ , din modelul inițial cu distribuții exponențiale.

Distribuțiile intersosirilor în stații le-am presupus că sînt bine approximate prin legi exponențiale de rată  $\hat{\lambda}_i$ ,  $i=2,3,\dots,m-1$ , pentru că serviciile au legi exponențiale și că reacțiile nu fac în realitate decît să mărească șirul de așteptare (dacă mărimea acestuia este infinită, aceste distribuții sînt exact exponențiale).

Din cele arătate anterior în cazul sistemului cu două stații pe care l-am studiat, rezultă că putem studia stațiile în mod independent unele de altele. Legătura între ele se face la nivelul calculului de probabilitate de neacceptare a unui client în stația următoare, probabilitate de tip Bernoulli, notată cu  $p_i$ ,  $i=2,3,\dots,m$ . Prin urmare vom examina fiecare stație ca avînd intersosiri care urmează un proces exponențial și un serviciu echivalent cu cel inițial, deasemenea exponențial.

Deci, vom rezolva următorul sistem pentru stația  $i$ ;

$$\begin{cases} \hat{\lambda}_i = \lambda/(1-p_i) \text{ și} \\ p_i = \hat{\rho}_i^{M_i} (1-\hat{\rho}_i)/(1-\hat{\rho}_i^{M_i+1}), \hat{\rho}_i = \hat{\lambda}_i/\hat{\mu}_i \text{ pentru } i=1,2,\dots,m \end{cases} \quad (15)$$

Deoarece  $\hat{\mu}_i$  depinde de  $p_{i+1}$ , stațiile nu sînt independente, dar dacă începem studiul cu stația ultimă,  $m$ , obținem  $\hat{\lambda}_m$  și  $p_m$  pentru că  $p_{m+1}=0$  și din aproape în aproape se obține cuplul de soluții  $(\hat{\lambda}_i, p_i)$  pentru  $i=m, m-1,\dots,2,1$ .

Condiția de echilibru este realizată dacă pentru toți  $i$ ,  $\hat{\lambda}_i \leq \hat{\mu}_i$ .

În acest caz obținem atunci pentru fiecare stație  $i$ , probabilitatea ca să aibă  $n_i$  clienți:

$$P_i(n_i) = \hat{\rho}_i^{n_i} (1-\hat{\rho}_i) / (1-\hat{\rho}_i^{M_i+1}) \quad \text{unde} \quad (16)$$

$$\hat{\rho}_i = \lambda / [u_i(1-p_i)(1-p_{i+1})]$$

Pentru a determina mărimea fluxului maxim de intrare în sistem  $\lambda_{\max}$ , observăm că acest flux va deveni maxim în momentul în care va exista o stație  $i$  cu flux maxim  $\hat{\rho}_i = \hat{\lambda}_i / \hat{\mu}_i = 1$ , iar pentru celelalte stații  $j \neq i$  avem  $\hat{\lambda}_j \leq \hat{\mu}_j$ .

Folosind această idee, este posibil să determinăm încărcarea maximă  $\hat{\rho}_{\max}$ , ce poate fi gestionată de sistem, în cazul în care toate stațiile sînt identice și au capacitatea 1 (presupunere pe care am făcut-o și la rezolvarea analitică), adică  $\hat{\mu}_i = \mu$  și  $M_i = 1, i = 1, 2, \dots, m$ . În această situație sistemul (15) devine:

$$\hat{\lambda}_i = \lambda / (1-p_i)$$

$$p_i = \hat{\rho}_i / (1+\hat{\rho}_i)$$

$$\hat{\rho}_i = \hat{\lambda}_i / \mu(1-p_{i+1}) \quad \text{unde } i = 1, 2, \dots, m \quad (17)$$

și dacă punem  $\rho = \lambda / \mu$ , obținem de mai sus:

$$p_i = \hat{\rho}_i / (1+\hat{\rho}_i) = (\hat{\lambda}_i / (\mu(1-p_{i+1}))) / (1 + \hat{\lambda}_i / (\mu(1-p_{i+1}))) =$$

$$= \hat{\lambda}_i \mu(1-p_{i+1}) / (\mu(1-p_{i+1}) + \hat{\lambda}_i) =$$

$$= \lambda / ((1-p_i)(\mu(1-p_{i+1}) + \lambda / (1-p_i))) = \lambda(1-p_i) / ((1-p_i)(\mu(1-p_{i+1}) + \lambda))$$

$$\text{sau } 1/p_i = (1-p_i)(1-p_{i+1}) / \rho + 1$$

$$\text{și } (1-p_i)/p_i = (1-p_i)(1-p_{i+1}) / \rho$$

$$\text{de unde rezultă: } p_i = \rho / (1-p_{i+1}), \quad i = 1, 2, \dots, m \quad (18)$$

Tot prin înlocuiri succesive, avem

$$\hat{\rho}_i = \lambda / (1-p_i) \mu(1-p_{i+1}) = \rho / ((1-\hat{\rho}_i / (1+\hat{\rho}_i)) \cdot (1-\hat{\rho}_{i+1} / (1+\hat{\rho}_{i+1}))) =$$

$$= (1+\hat{\rho}_i) \rho (1+\hat{\rho}_{i+1}) \quad \text{adică}$$

$$\hat{\rho}_i - \hat{\rho}_i \rho (1+\hat{\rho}_{i+1}) = \rho (1+\hat{\rho}_{i+1}), \quad \text{deci}$$

$$\hat{\rho}_i = \rho (1+\hat{\rho}_{i+1}) / (1-\rho(1+\hat{\rho}_{i+1})), \quad i = 1, 2, \dots, m \quad (19)$$

Din relațiile recursive (18) și (19) pentru  $i = m$ , rezultă  $p_m = \rho$  și  $\hat{\rho}_m = \rho / (1-\rho)$  deoarece  $p_{m+1} = 0$ .

Folosind ideea de mai sus pentru determinarea încărcării maxime, putem folosi următoarea tehnică de calcul prin care se poate găsi încărcarea maximă a primei stații,  $\hat{\rho}_{\max}$ , care, așa cum am observat, va fi tot una cu încărcarea maximă

$\rho_{\max}$  a sistemului cu stații în serie.

Astfel, avînd că  $\hat{\rho}_m = \frac{\rho}{1-\rho}$  și folosind recursiv formula (19) de  $m-1$  ori pentru  $i = m-1, m-2, \dots, 2, 1$ , putem ajunge în final la determinarea în funcție de  $\rho$  a tuturor încărcărilor  $\hat{\rho}_i$ , inclusiv a lui  $\hat{\rho}_1$ , deci a încărcării sistemului.

Metodă combinată; difuzie și aproximare . . . . . Am efectuat în continuare, legat de tratarea matematică a sistemului, un studiu privind modelul matematic care reprezintă suficient de fidel sistemul nostru din realitate cu extindere la o rețea de calculatoare. Am utilizat în acest scop metoda noastră prin aproximare pe care am găsit-o cea mai adecvată nevoilor noastre.

Tratarea matematică cît mai corectă a sistemului nostru în această manieră este o rezolvare originală și constituie o contribuție personală în domeniul teleprelucrării datelor; acest studiu oferă totodată o metodologie eficientă de lucru privind concepția de realizare a rețelelor de calculatoare

Se remarcă în analiza care urmează, că am considerat existența unui tip specific de protocol, ceea ce face ca modelul studiat pentru sistemul nostru să se apropie de situația reală. Protocolul este definit ca un set de reguli și convenții care asigură schimburile de informații între două sau mai multe calculatoare în cadrul unei rețele de calculatoare și apare ca urmare a blocării mesajelor în anumite puncte din cadrul rețelei respective, cînd datorită limitării de capacitate în tamponurile de memorie o transmisie nu are atunci succes din lipsă de spațiu de memorare. Stabilitatea sistemului nostru ca și performanțele sale sînt influențate de protocolul ales.

În analiza care urmează am considerat un singur tip de protocol, corespunzător căruia după transmiterea unui mesaj sau pachet se efectuează o retransmitere de la stația precedentă cu confirmare pozitivă sau negativă, protocol de tip "transmite și așteaptă" (send and wait) care este folosit în realitate în cadrul sistemului nostru. Acesta este cazul de exemplu al procedurii "Transmisie Mod Mesaj - TMM" adoptată în lucrarea noastră și specifică calculatoarelor Felix C-256.

Rezolvarea matematică a sistemului nostru în aceste condiții constă în delaborarea și analiza modelului cu un număr oarecare de șiruri de așteptare în serie, cu procese care au distribuții de tip general, pentru determinarea parametrului de bază, încărcarea maximă a sistemului și care e-

evident implică o rată maximă de intrare în sistem.

Problema majoră care se pune pentru rezolvarea modelului nostru este menținerea fluxului de date de la intrare în limite compatibile cu mărimea resurselor disponibile. Putem afirma deci că rata intrărilor (cantitatea sau numărul intrărilor pe unitate de timp) pentru care toate mesajele pot fi prelucrate, este egală cu capacitatea de trafic a sistemului și determină stabilitatea legăturii între calculatoare sau a rețelei respective.

Abordarea noastră prezintă o mare generalitate prin faptul că lucrăm cu stații de tip  $M/G/1/M_1$  și  $G/G/1/M_1$  (legi exponențiale și respectiv generale pentru timpii de intersosire, legi generale pentru timpii de serviciu, iar capacitatea șirului de așteptare este limitată la  $M_1$  clienți).

Modelele pe care le studiem reprezentate în fig.3.1. sînt formate dintr-un sistem de  $m$  stații în serie, stația  $i, i=1,2,\dots,m$  avînd capacitatea  $M_i, i=2,\dots,m$  și  $M_1=\infty$ , iar clienții trecînd consecutiv prin fiecare stație pentru a fi serviți după disciplina FIFO.

Instrumentele matematice principale care ne permit studiul modelelor cu stații în serie, cu legi generale de serviciu și de intersosire îl constituie teoria aproximării prin echivalență și prin procese de difuzie.

Efectul de respingere a clienților dintre stații (reacția) l-am înlăturat prin echivalarea fiecărei stații inițiale cu o stație cu timpii de serviciu modificați corespunzător cu probabilitatea de respingere la stația următoare și am presupus că probabilitatea de respingere la stația  $i, p_i, i=2,\dots,m$  este egală cu probabilitatea de a avea  $M_i$  clienți în stația  $i$ .

Ca și în cazul anterior studiat la tehnica de echivalență stațiile echivalente  $M/G/1/M_1$  sau  $G/G/1/M_1$  vor avea următorii parametrii:

$$\lambda_i = \lambda / (1 - p_i) \quad (20)$$

Aceasta înseamnă că numărul de clienți care trece de la stația  $i-1$  la  $i$  este suma noilor transmisii și a numărului de clienți retransmiși din stația  $i-1$ .

unde  $\lambda$  este rata procesului de sosiri în sistemul inițial, de tip Poisson și respectiv, general.

- Distribuția de probabilitate a timpului de serviciu la stația  $i$ , (aceasta înseamnă că noul timp de serviciu este timpul total de rămînere a unui client în stație), este:

$$h_i(x) = (1 - p_{i+1}) \bigoplus_{k=1}^{\infty} p_{i+1}^{k-1} f_i^{*k}(x) \quad (21)$$

unde  $\oplus$  reprezintă suma prin convoluție, iar  $f_i(x)$  este distribu-



ția timpului de serviciu la stația inițială (timpii de serviciu considerându-se independenți). Aceasta corespunde la faptul că cu probabilitatea  $(1-p_{i+1})p_{i+1}^{k-1}$  clientul este servit de k ori,

- Rata serviciului la stația i va fi:

$$\hat{\mu}_i = 1/E(h_i(x)) = \mu_i(1-p_{i+1}) \quad (22)$$

unde  $\mu_i = 1/E(f_i(x))$ .

Folosind în continuare tehnica aproximărilor prin procese de difuzie, modelele de așteptare de tip  $M/G/1/M_i$  sau  $G/G/1/M_i$  de mai sus le putem rezolva calculând probabilitatea  $P_i(n)$ , de a avea n clienți în stația i și evident probabilitățile  $P_i(M_i)$ , ca șirul de așteptare al stației i să fie plin. După aceea folosind că probabilitatea de respingere este egală cu probabilitatea ca stația să fie plină, adică  $p_i = P_i(M_i)$  căutăm să ajungem la un sistem cu necunoscutele  $\lambda_i$  și  $p_i$  din care să putem determina încărcarea stațiilor  $\hat{\rho}_i = \lambda_i/\hat{\mu}_i$  și apoi rata maximă de intrare  $\lambda_{\max}$ . Încărcarea maximă în sistem este  $\rho_{\max} = 1$  și se produce exact când este asigurată stabilitatea totală a șirului de așteptare la prima stație, și deoarece  $\rho_{\max} = \lambda_{\max}/\hat{\mu}_1$  rezultă  $\lambda_{\max} = \hat{\mu}_1$ . O formulare echivalentă este,  $\lambda_{\max} = \mu_1(1-p_2)$  sau  $\lambda_2 = \mu_1$ .

Pentru rezolvarea modelului nostru în cazul că stațiile sînt de tip  $M/G/1/M_i$  vom determina probabilitatea  $P_i(n)$  relativ la o stație de acest fel pe baza rezultatelor privind stațiile de tip  $M/G/1$ . Pentru aceasta notăm cu  $P(n)$  probabilitatea de a avea n clienți într-o stație de tip  $M/G/1$  iar pentru această stație folosim notațiile de pînă acum.

Particularizînd formula demonstrată anterior la aproximarea prin difuzie pentru stații de tip  $G/G/1$ , respectiv formula (8), am procedat la rezolvarea modelelor în care am presupus că stațiile sînt de tip  $M/G/1$  unde sosirile formează flux Poisson și atunci coeficientul pătratic de variație al timpilor de intersosire la stația i,  $C_{a,i} = 1$ .

$$P(n) = \begin{cases} 1 - \hat{\rho}_i & \text{dacă } n=0 \\ \hat{\rho}_i(1-\tilde{\rho}_i)\tilde{\rho}_i^{n-1} & \text{dacă } n \geq 1 \end{cases} \quad (23)$$

unde  $\hat{\rho}_i = \lambda_i/\hat{\mu}_i$ ,  $\tilde{\rho}_i = \exp(2(\lambda_i - \hat{\mu}_i)/(\lambda_i + C_{s,i}\hat{\mu}_i))$ ,

semnul  $\wedge$  semnifică aproximarea prin echivalență,

iar semnul  $\sim$  semnifică în plus și aproximarea prin difuzie,

unde  $C_{s,i}$  este coeficientul pătratic de variație al timpilor de serviciu la stația i.

Se știe că în regim staționar, pentru stația  $M/G/1/M_i$  probabilitatea  $P_i(n)$  se poate exprima în funcție de

probabilitatea  $P(n)$  relativă la stația  $M/G/1$  în felul următor:

$$P_i(n) = P(n)/(P(0)+\dots+P(i)) \quad (24)$$

Inlocuind în formula (24) pentru diferite valori ale lui  $n$  din formula (23) obținem:

$$\begin{aligned} P_i(n) &= \hat{f}_i(1-\tilde{f}_i)\tilde{f}_i^{n-1}/((1-\hat{f}_i)+\hat{f}_i(1-\tilde{f}_i)+\dots+\hat{f}_i(1-\tilde{f}_i)\tilde{f}_i^{i-1}) = \\ &= \hat{f}_i(1-\tilde{f}_i)\tilde{f}_i^{n-1}/((1-\hat{f}_i)+\hat{f}_i(1-\tilde{f}_i)(1+\tilde{f}_i+\dots+\tilde{f}_i^{i-1})) = \\ &= \hat{f}_i(1-\tilde{f}_i)\tilde{f}_i^{n-1}/((1-\hat{f}_i)+\hat{f}_i(1-\tilde{f}_i^n)) = \hat{f}_i(1-\tilde{f}_i)\tilde{f}_i^{n-1}/(1-\hat{f}_i\tilde{f}_i^i). \end{aligned}$$

În particular probabilitatea ca un client să fie respins de stația  $i$  următoare, al cărei șir de așteptare de capacitate  $M_i$  este plin, este:

$$p_i = P_i(M_i) = \hat{f}_i(1-\tilde{f}_i)\tilde{f}_i^{M_i-1}/(1-\hat{f}_i\tilde{f}_i^{M_i}).$$

Astfel, comportamentul stației  $i$  este determinat de sistemul de relații:

$$\begin{cases} \lambda_i = \lambda/(1-p_i) \\ p_i = \hat{f}_i(1-\tilde{f}_i)\tilde{f}_i^{M_i-1}/(1-\hat{f}_i\tilde{f}_i^{M_i}) \end{cases} \text{ unde } i = 2, 3, \dots, m \quad (25)$$

în care necunoscutele sînt  $\lambda_i$  și  $p_i$

Un sistem asemănător am găsit în cazul exponențial și stații  $M/M/1/M_i$  și acum rezolvarea este analogă. Astfel, deoarece  $p_{m+1}=0$  (clienți care ies din ultima stație nu sînt respinși la ieșirea din sistem) putem obține succesiv cuplurile de necunoscute  $\lambda_i, p_i$  pentru  $i=m, m-1, \dots, 2$  în funcție de  $\lambda$  dat. Rata maximă de intrare  $\lambda_{\max}$  se obține cînd  $\lambda_{\max} = \hat{\mu}_1$ , sau echivalent,  $\hat{f}_{\max} = \hat{f}_{1, \max} = 1$ ; aceasta se obține prin ajustări corespunzătoare ale lui  $\lambda$  astfel încît să avem  $\lambda = \mu_1(1-p_2)$ .

În continuare am abordat modelul în cazul cîi stațiile sînt de tip  $G/G/1/M_i$ . Rezolvarea în cazul cel mai general ( $G/G/1/M_i$ ) necesită și acum determinarea probabilităților  $P_i(n)$ . Pentru aceasta, deoarece aceste cazuri nu se pot rezolva cu o formulă analogă cu (5) am folosit în schimb un rezultat ajutător din literatura de specialitate obținut tot pe calea aproximării prin procese de difuzie; acest rezultat (Gelenbe) se poate stabili asimilînd reacția dintre stațiile  $n$  și  $n+1$  cu o rețea închisă cu două stații și făcînd apel la proprietățile unor procese stochastice speciale, numite procese cu reveniri instantanee.

Astfel, probabilitatea ca un client să fie respins de stația  $i$  al cărei șir de așteptare este plin, este:

$$p_i = P_i(M_i) = \hat{f}_i(1-\hat{f}_i)/(\exp(-\gamma_i(M_i-1))-\hat{f}_i^2) \quad (26)$$

unde,  $\gamma_i = 2(\lambda_i - \hat{\mu}_i)/(\lambda_i C_{a,i} + \hat{\mu}_i C_{s,i})$

notațiile fiind cele de pînă acum.

Așadar, sistemul de necunoscute  $\lambda_i$  și  $p_i$  care trebuie rezolvat este:

$$\begin{cases} p_i = \hat{p}_i(1-\hat{p}_i)/(\exp(-\mu_i(M_i-1))-\hat{p}_i^2) \text{ unde } i=2,3,\dots,m & (27) \\ \lambda_i = \lambda/(1-p_i) \end{cases}$$

Rezolvarea sistemului se face considerînd  $p_{m+1}=0$  și calculînd pas cu pas, începînd cu  $i=m$  pînă la  $i=2$ . In expresiile astfel aflate ale lui  $p_i$  și  $\lambda_i$  ajustăm parametrul  $\lambda$  dat, de așa manieră ca să obținem  $\lambda = \mu_1^i(1-p_2)$ , relație care exprimă că încărcarea primei stații și deci a întregului sistem, este maximă, valoarea astfel ajustată a lui  $\lambda$  reprezentînd rata maximă de intrare, adică  $\lambda_{\max}$  a sistemului.

Vom calcula în continuare pentru cîteva cazuri de distribuții particulare rata maximă de intrare  $\lambda_{\max}$  a unui sistem format din două stații în serie, prima de capacitate nelimitată, iar a doua cu capacitatea  $M$ , fig.3.3. Așa cum am arătat mai sus rata maximă de intrare se obține pentru  $\lambda_2 = \mu_1$ , relație echivalentă cu  $\lambda = \mu_1(1-p_2)$ .

Avem deci de rezolvat următorul sistem:

$$\begin{cases} \lambda_2 = \lambda_{\max}/(1-p_2) \\ p_2 = \hat{p}_2(1-\hat{p}_2)/\exp(-\mu_2(M-1))-\hat{p}_2^2 \end{cases} \quad (28)$$

Deoarece  $p_3 = 0$  deducem că  $\hat{\mu}_2 = \mu_2$  și înlocuind în formula de definiție a lui  $\hat{p}_2$  găsim că  $\hat{p}_2 = \lambda_2/\mu_2$ . Din aceasta și din relația  $\lambda_2 = \mu_1$  rezultă că  $\hat{p}_2 = \mu_1/\mu_2$ .

Expresia lui  $p_2$  devine:

$$p_2 = \frac{\mu_1/\mu_2(1-\mu_1/\mu_2)}{(\exp(-2(\mu_1-\mu_2)/(\mu_1 C_{s,1} + \mu_2 C_{s,2}))^{M-1} - (\mu_1/\mu_2)^2)} \quad (29)$$

unde s-a folosit că  $C_{s,1} = C_{a,2}$ , relație prezentată anterior la studiul proceselor de difuzie.

Inlocuind pe  $p_2$  în prima ecuație a sistemului de mai sus și ținînd cont că  $\lambda_2 = \mu_1$ , rezultă  $\lambda_{\max}$ .

In cazul în care rata serviciilor în cele două stații este aceeași, adică  $\mu_1 = \mu_2 = \mu$  se obține pentru rata maximă de intrare;

$$\lambda_{\max} = \mu(C_{s,1} + C_{s,2} + 2(M-1))/(2(C_{s,1} + C_{s,2} + M-1)) \quad (30)$$

In situațiile particulare de serviciu cu distribuție Erlang, hiperexponențială sau exponențială negativă, se obțin respectiv rezultatele:

- Dacă distribuția timpilor de serviciu este Erlang de ordin  $r$

cu rata  $\mu=1$  și  $C_{s,1} = C_{s,2} = 1/r$  rezultă:

$$\lambda_{\max} = (r(M-1)+1)/(r(M-1)+2). \quad (31)$$

Pentru  $r = 1$ , obținem  $\lambda_{\max} = M/(M+1)$ , iar pentru  $r = \infty$ , adică pentru timpii de serviciu constanți obținem  $\lambda_{\max} = 1$ .

- Dacă distribuția timpilor de serviciu este hiperexponențială cu  $C_{s,1} = C_{s,2} = r$ , rezultă:

$$\lambda_{\max} = (r+M-1)/(2r+M-1)$$

Pentru  $r=1$ , sîntem în cazul exponențial și obținem  $\lambda_{\max} = M/(M+1)$ , iar pentru  $r = \infty$  obținem cazul cel mai nefavorabil, adică  $\lambda_{\max} = 1/2$ .

Ca o extensie a studiului pentru stații de tip  $G/G/1/M_i$  vom stabili cîteva idei pentru cazul cînd se ia în considerare și timpul de supraîncărcare. Intr-adevăr, cînd un pachet sau mesaj găsește șirul de așteptare de la stația următoare plin înainte de lansarea confirmării negative de primire, se scurge un timp pe care l-am denumit timp de supraîncărcare și de care e bine, pentru creșterea preciziei, să se țină seama. Pînă în prezent am considerat că acest timp, ca și timpul de retransmisie este înglobat în timpul de serviciu.

Ca atare, să presupunem acest timp de supraîncărcare neglijabil și de valoare constantă  $d$ , așa cum este ilustrat în fig.3.5. Folosind aceleași notații ca și pînă acum, precum și procedeul de echivalare a stațiilor, putem afirma că stațiile echivalente de tip  $G/G/1/M_i$ ,  $i=2,3,\dots,m$  vor avea ca legi de serviciu, distribuții de tipul:

$$h_i(x) = \sum_{k=0}^{\infty} (1-p_{i+1})^k p_{i+1}^k \varepsilon_m(x),$$

unde  $\varepsilon_m(x) = \begin{cases} \Theta_n \\ f_i(x-d) \Theta f_i(x) \end{cases}$ .

și unde  $f_i(x)$  este distribuția timpilor de serviciu din sistemul inițial.

Presupunînd că stațiile inițiale au distribuțiile serviciilor exponențiale, adică  $f_i(x) = \mu_i e^{-\mu_i x}$ , stațiile echivalente nu vor mai avea din cauza acestui timp, distribuții de servicii de tip exponențial; această presupunere ușurează însă calculul mediei, dispersiei și coeficienților pătratici de variație, calcul care ne este necesar în continuare la rezolvarea sistemului  $G/G/1/M_i$  prin procese de difuzie.

Oricum, cea mai eficientă tehnică de abordare a acestor stații cu timpii de serviciu generali este metoda de aproximare prin difuzie, așa cum am mai subliniat în capitolul respectiv.

Să mai remarcăm că rezolvarea se poate ușura aproximând distribuțiile timpilor de interesare sau interplecare de la stații ca fiind exponențiale cu rata  $(d+1/\mu_n)^{-1}$ . Această presupunere nu este departe de realitate și efectul ei este că ajungem cu studiul în cazul stațiilor de tip  $M/G/1/M_1$ , a căror rezolvare este totuși mai ușoară decât a celor de tip  $G/G/1/M_1$ . Și într-un caz și în celălalt, rezolvările pot fi conduse pînă la capăt urmînd aceleași tehnici ca și în situația neconsiderării timpului de supraîncărcare  $d$ , atîta doar că de această dată calculele vor fi mai complexe.

Studiile și rezolvările noastre au fost făcute pe modele matematice care reprezintă o rețea de calculatoare mai precis o legătură generală, tip, între calculatoare dintr-o rețea oricît de complicată, în care calculatoarele au capacități limitate, cu excepția primului care l-am presupus cu capacitate infinită jucînd rolul de calculator privilegiat.

Am lucrat, întotdeauna sub presupunerea că timpii de serviciu ai stațiilor sînt independenți și că șirul de așteptare inițial nu e vid și atunci rata maximă de intrare este identică cu rata plecărilor din rețea (situația de stabilitate a sistemului).

Validarea și acuratețea aproximărilor noastre pot fi verificate și măsurate pe o legătură reală între calculatoare sau pot fi verificate prin diverse procedee de simulare.

De asemenea rezultatele obținute în lucrarea noastră pentru cazurile generale și cele particulare utilizînd în special tehnicile de aproximare sînt compatibile atît ca precizie cît și ca formă cu alte rezultate recent întîlnite în literatura de specialitate.

Referitor la scopul atins în această parte finală aplicativă a lucrării noastre, determinarea ratei maxime de intrare pentru ca intensitatea de trafic să fie optimă ( $\rho_{\max} = 1$ ), menționăm că rezultatele obținute de noi și particularizate pentru diverse tipuri de distribuții se pretează la analize comparative utile. Dintre concluziile generale ce se pot desprinde din analiza rezultatelor, să distingem că cea mai remarcabilă constă în faptul că rata maximă de intrare este foarte sensibilă la distribuția timpului de transmisie în sistem pe care am asimilat-o cu distribuția lungimii pachetelor de mesaje.

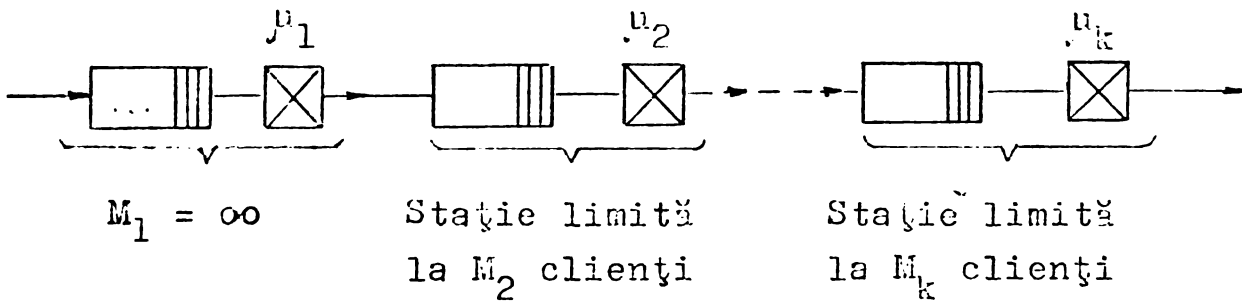


Fig.3.1. Sistem de așteptare cu  $k$  stații în serie, prima de capacitate infinită iar celelalte de capacitate  $M_n$ ,  $n = 2, \dots, k$ .

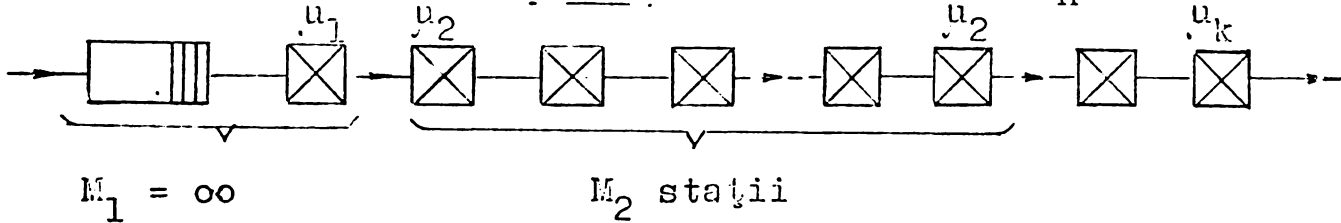


Fig.3.2. Sistem de așteptare cu  $k$  stații în serie prima de capacitate infinită iar celelalte de capacitate 1.

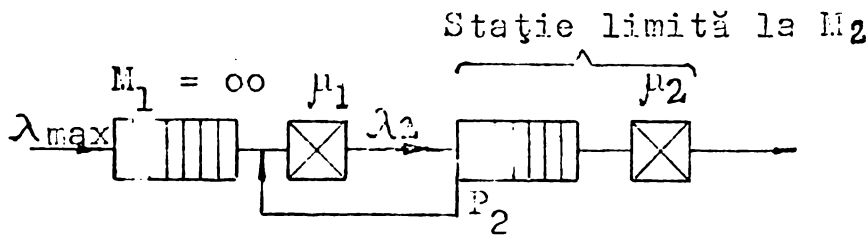


Fig.3.3. Sistem cu reacție cu două stații în serie, prima de capacitate infinită iar cea de a doua de capacitate  $M_2$ .

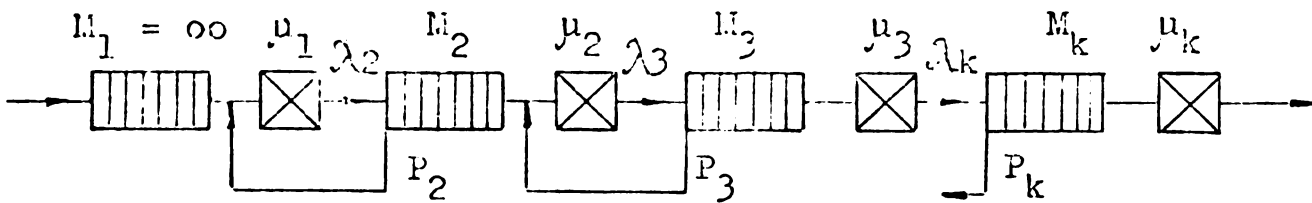


Fig.3.4. Sistem de așteptare cu reacție, cu stații în serie, prima cu capacitate infinită iar celelalte cu capacități finite  $M_n$ ,  $n = 2, \dots, k$ .

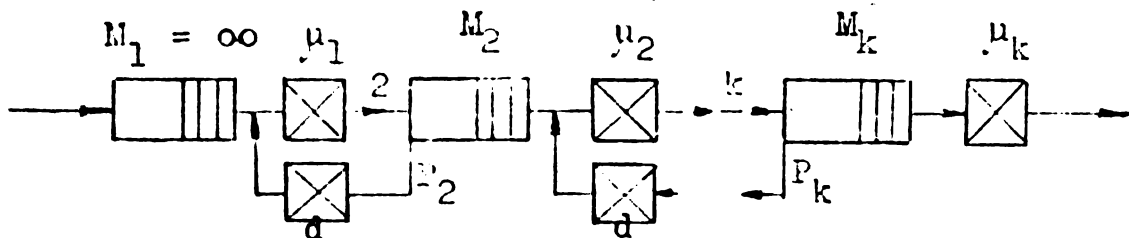


Fig.3.5. Sistem de așteptare cu reacție, cu stații în serie, prima stație cu capacitate infinită, iar celelalte cu capacități finite și în care intervine o întârziere de prelucrare  $d$ .

CAP.4. SISTEMUL DE EXPLOATARE PENTRU LEGATURA (UC-UC)  
INTRE DOUA CALCULATOARE DE TIP FELIX C-256.  
PRODUSUL PROGRAM.

4.1. INTRODUCERE

In această parte a lucrării, am conceput și am realizat practic Sistemul de exploatare pentru legătura între calculatoare (UC-UC), printr-un produs program aplicat la calculatorul electronic de tip Felix C-256; produsul program este extins și la o rețea de calculatoare.

Realizarea produsului program a necesitat parcurgerea unor faze ca; elaborarea concepției aplicative, proiectarea tehnică, construirea de programe, implementarea și experimentarea.

Am pornit de la soluțiile teoretice prezentate anterior pentru un sistem ipotetic pe care însă nu le-am preluat integral, fără modificări, deoarece pînă în prezent nu se dispune de sisteme de calcul cu procesoare virtuale; ca urmare am adaptat și dezvoltat soluțiile teoretice la cerințele concrete ale calculatorului Felix C-256.

Această realizare aplicativă, finalizată printr-un produs program, are un caracter original și constituie o contribuție personală în ceea ce privește software-ul de bază de teleprelucrare.

Datorită calităților, performanțelor, gradului mare de generalizare și ariei de cuprindere, produsul program elaborat se situează la un nivel superior și poate face parte din categoria sistemelor de exploatare și ca atare a Sistemului de exploatare al calculatorului Felix.

Totodată această realizare constituie o metodologie de bază privind concepția aplicativă, proiectarea tehnică, construirea programelor și implementarea unor astfel de produse de teleprelucrare.

Pînă în prezent în țară în acest domeniu, al teleprelucrării datelor, nu a fost realizat practic un astfel de sistem, de mare anvergură și materializat printr-un produs program generalizabil; lucrările elaborate pe această temă pînă în prezent, au avut mai mult un caracter particular și s-au bazat pe software-ul de tele-

prelucrare al calculatorului Felix. În străinătate au existat și există preocupări deosebite pentru realizarea unor produse program pentru sisteme de teleprelucrare și pentru rețele de calculatoare (ARPA, CYCLADES și altele). Totuși aceste realizări au caracter mai puțin generalizabil determinat de lipsa unei concepții generale de sistem și de faptul că în general sistemele respective au un obiectiv particular, cu un anumit specific.

Literatura de specialitate nu este suficient de bogată în ceea ce privește realizările practice; cu toate acestea am consultat o serie de lucrări din domeniu prezentate în bibliografie (4), care mi-au sugerat idei interesante.

Am abordat în această parte a lucrării o serie de probleme teoretice și practice de mare importanță pe care le-am rezolvat corespunzător.

Astfel, Sistemul de exploatare pentru legătura UC-UC între două calculatoare Felix C-256, respectiv produsul program corespunzător, este elaborat pe baza unei concepții aplicative proprii.

Cu unele modele matematice proprii, am realizat funcțiile importante de teleprelucrarea datelor care reprezintă procesul respectiv de controlare, organizare, structurare și transmitere a informațiilor în interiorul calculatoarelor în perifericele locale și în linii de transmisie. Am elaborat un executiv sau supervisor care asigură controlul și gestionarea sarcinilor pe tot parcursul existenței lor. Am creat o procedură care cuprinde comenzi pentru stabilirea legăturilor între calculatoare și efectuarea în bune condiții a schimburilor de date. Am utilizat facilitățile calculatorului din punct de vedere hardware și software precum și tehnici moderne din domeniul informaticii ca; simultaneitatea prelucrărilor cu intrări/ieșiri, multiprogramarea, așteptarea multiplă a evenimentelor, conexiunea fizică directă între memoriile calculatoarelor, utilizarea funcțiilor sistemului de gestiunea fișierelor și a transmisiilor (SGF și SGT), limbajul ASSIRIS, sincronizarea sarcinilor cu ajutorul semafoarelor, crearea și derularea sarcinilor în paralel și altele. Am utilizat unele tehnici și metode moderne de programare pentru a realiza un produs program modular și structurat cu posibilitatea ca părți componente să fie elaborate și să funcționeze separat, să poată suporta modificări, adaptări și extensii prin parametrizare.

Astfel, sistemul nostru a fost adaptat la cerințele calculatorului Felix și a fost organizat și structurat în trei mari părți



componente: Procedura de exploatare, Sistemul de gestiunea sarcinilor și Comenzile.

Procedura de exploatare pentru legătura UC-UC, este constituită dintr-un ansamblu de sarcini cooperante realizate cu ajutorul unor modele matematice; se situează logic și fizic dar pe o treaptă superioară, între nivelul fișierelor și nivelul liniei de transmisie. Realizează teleprelucrarea datelor prin organizarea și gestionarea informațiilor care circulă prin subcanalul stabilit în acest sens între calculatoare și controlează accesul la fișiere prin Sistemul de gestiunea fișierelor (SGF) și la linia de transmisie, prin Sistemul de gestiunea transmisiilor (SGT). Se compune din trei părți: Comunicația care asigură legătura între utilizator (operator) și Sistemul de exploatare pentru legătura UC-UC; Emisia, care asigură formarea, pregătirea și transmiterea în linie a mesajelor; Recepția, care asigură primirea mesajelor din linie, interpretarea lor, execută comenzile și scrie în fișiere articolele corespunzătoare subcanalelor.

Sistemul de gestiunea sarcinilor realizează coordonarea sarcinilor, asigurând activarea și dezactivarea unei sarcini, lansarea multitaskingului, introducerea sau scoaterea unui cuvânt în șirul de așteptare al unei sarcini, punerea unei sarcini în așteptarea unui eveniment dintre  $n$  evenimente, etc.

Comenzile necesare legăturii UC-UC, realizează punerea de acord și avertizarea reciprocă a calculatoarelor în privința condițiilor de funcționare și se referă la stabilirea unui subcanal, controlarea schimburilor de date în linie și suspendarea acestora.

Acești comenzi, le sînt asociate date, la transmiterea în linie; aceasta constituie o caracteristică proprie acestui Sistem de exploatare, deosebit de importantă pentru o funcționare eficientă.

În figura 4.1. se arată schema de principiu a sistemului de exploatare pentru legătura UC-UC, cu componentele sale de bază.

În această parte a lucrării, am acordat o atenție deosebită posibilităților calculatorului Felix C-256.

Referitor la aceasta, față de existența unor facilități software și hardware, acest calculator prezintă și o serie de restricții în special legate de Sistemul de gestiunea fișierelor și Sistemul de gestiunea transmisiilor.

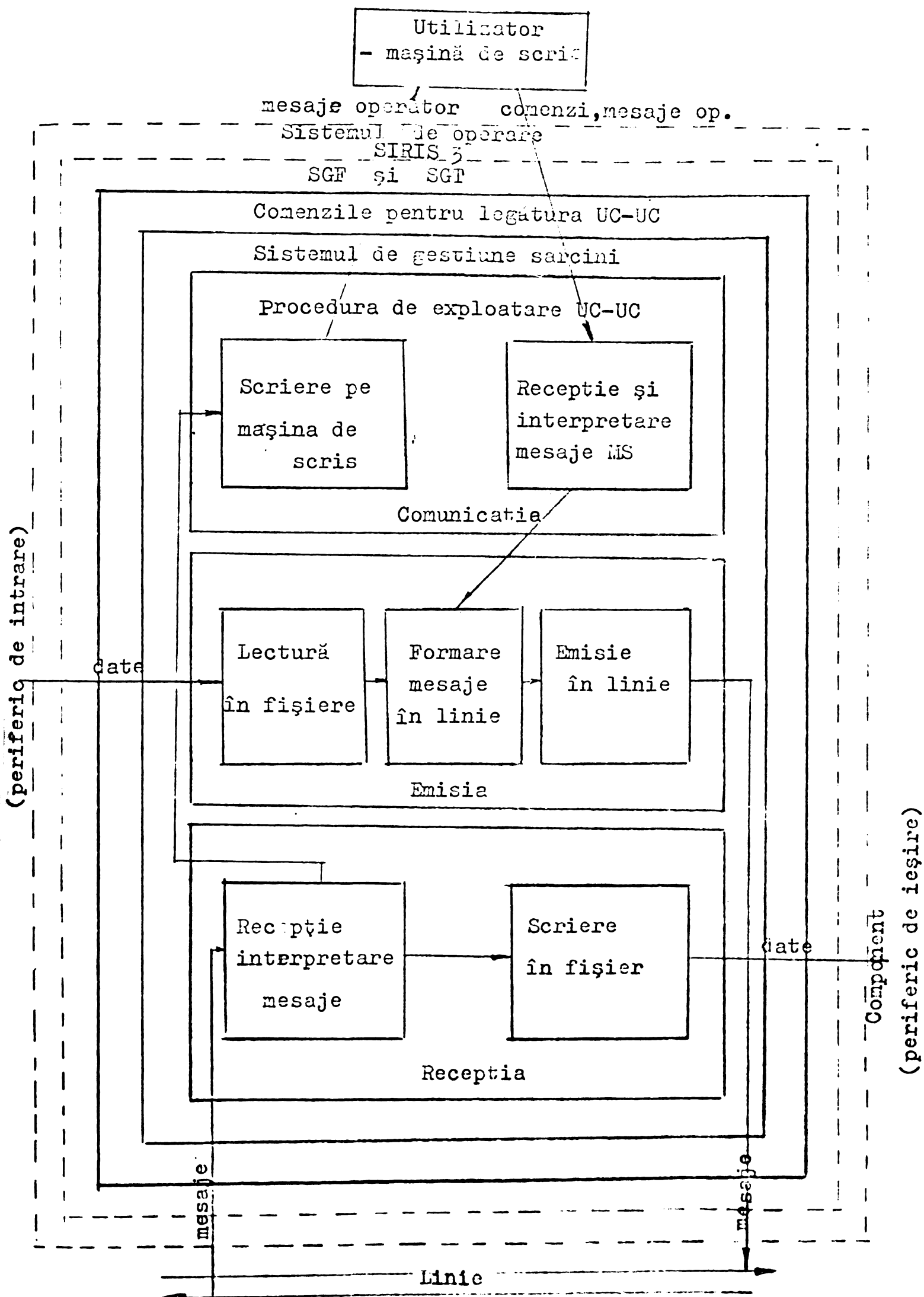


Fig.4.1 Sistemul de exploatare UC-UC - schemă de principiu

Pentru realizarea Sistemului am fost nevoit să trec la eliminarea acestor restricții prin găsirea unor soluții proprii.

Succint, se poate în această privință arăta că:

- În legătură cu Sistemul de gestiunea fișierelor, este indicată utilizarea gestiunii automate a tamponelor pentru a economisi memorie și timp; inconvenientul constă însă în faptul că această gestiune automată exclude orice simultaneitate de prelucrări și intrări/ieșiri. Soluția găsită constă în exploatarea fișierului în organizare nedefinită nivel fizic, sau pentru fișierele secvențiale prin gestiunea programată a tamponelor.

- În legătură cu Sistemul de gestiunea transmisiilor, care este de altfel software-ul de bază elementar de teletransmisie și care coordonează schimburile de blocuri de informații în linie, cu toate facilitățile de care dispune nu este total satisfăcător pentru cerințele Sistemului nostru.

Apare o dificultate la înlănțuirea datelor la transmisia acestora în linie, deoarece înlănțuirea explicită privește zonele tampon ale unui pool gestionat automat iar prin definiția pool-ului, toate zonele sînt de aceeași lungime. Soluția în această situație constă în înlănțuirea datelor prin plasarea acestora în mod convenabil în începutul pool-ului cu condiția ca să nu se efectueze un control în această zonă.

O altă dificultate apare la solicitarea SGT-ului respectiv a multitasking-ului său de a gestiona sarcini, deoarece sarcina în SGT este legată obligatoriu de linie iar la o linie corespunde o singură sarcină și care are o prioritate fixă.

Aceste dificultăți, de altfel majore ne-au condus la elaborarea unui Sistem de gestiunea sarcinilor propriu.

Am acordat o atenție deosebită construirii programelor, implementării și experimentării produsului program atît în ansamblu, cît și pe componentele sale.

Această parte din lucrare este prezentată în detaliu în anexele: 1 - program sursă și 2, 3, 4, experimentări.

Produsul program corespunzător Sistemului de exploatare pentru legătura UC-UC a fost realizat practic prin programe de calcul în limbajul ASSIRIS într-o structură modulară pentru cele trei mari componente și care au fost implementate, testate și experimentate cu rezultate corespunzătoare.

Produsul program rezultat este operațional atât pe componente cât și în ansamblu și constituie astfel un software de bază de teleprelucrare efectiv pentru calculatorul Felix C-256.

În cazul nostru produsul program se referă la aplicația de teleconversia datelor care realizează citirea unui fișier de intrare pe cartele la calculatorul Sursă de date și scrierea datelor într-un fișier de ieșire la imprimantă la calculatorul Receptor de date. Legătura între calculatoare în experimentul nostru este asigurată numai printr-un subcanal; teleconversia simultană a mai multor fișiere poate fi ușor realizată de către sistemul nostru prin extinderea acestuia.

Concret, acest produs program este format dintr-un ansamblu de programe, subprograme și tabele; tabelele se generează de fiecare utilizator în funcție de cerințele acestuia cu ajutorul unor parametri (număr de subcanale, lungime tampon, etc.).

În cadrul experimentului făcut în această lucrare, în final, au fost implementate și experimentate cu succes toate funcțiile principale ale produsului program. Astfel, am realizat:

- Transmisia de mesaje între mașinile de scris de pupitru ale celor două calculatoare;

- Transmisia de comenzi de stabilirea sau de închiderea legăturii între calculatoare și evident, comenzile interne care sînt generate;

- Transmisia de date între calculatoare, de la citirea datelor din fișierul primului calculator pînă la scrierea datelor în celălalt calculator.

Acest experiment a fost realizat pe un singur calculator, în două partiții (fiecare partiție simulînd un calculator separat) cu o legătură fizică între ele, cu ajutorul cuploarelor corespunzătoare. Rezultatele obținute sînt evident identice cu cele din cazul real, cu două calculatoare legate direct între memoriile centrale. O preocupare deosebită în această parte a lucrării am acordat extinderii sistemului de exploatare la o rețea de calculatoare.

Sistemul nostru ia în considerare o legătură bipunct (între două calculatoare). După cum se știe, putem concepe utilizarea în comun a sistemelor de calcul prin interconectarea calculatoarelor într-o rețea. Ca urmare, cu unele mici modificări, Sistemul nostru de exploatare corespunde cerințelor privind extinderea sa la o rețea de calculatoare. Pentru aceasta am considerat rețeaua de calculatoare ca un ansamblu de legături bipunct adică de subcanale (care trebuie bine definite pentru identificare).

#### 4.2. PROCEDURA DE EXPLOATARE PENTRU LEGATURA

##### UC-UC LA CALCULATOARE FELIX C-256

Procedura de exploatare pentru legătura UC-UC a două calculatoare Felix C-256 componentă importantă a Sistemului de exploatare pentru legătura UC-UC am conceput-o pentru a fi realizată și finalizată practic printr-un produs program care poate face parte și extinde software-ul de bază de teleprelucrare pentru calculatorul Felix C-256.

Această componentă constituie, de la concepție și pînă la realizare practică, o contribuție personală originală; astfel, problemele fundamentale din domeniul teleprelucrării ca: conceperea și elaborarea modelelor matematice care asigură teleprelucrarea, derularea în paralel a sarcinilor, sincronizarea sarcinilor, punerea în așteptare multiplă a acestora și altele au stat permanent în atenție pentru a fi rezolvate corespunzător.

Conceptul teoretic prezentat anterior în lucrare nu a fost transpus integral în practică fără modificări, deoarece sistemul de calcul Felix C-256 nu dispune de posibilitățile unui sistem cu procesoare virtuale (care permite derularea mai multor procese în paralel, sincronizarea lor, alocarea dinamică a memoriei, etc.).

Ca urmare s-au folosit facilitățile sistemului de calcul Felix C-256 ca; simultaneitatea intrări/ieșiri și prelucrări, multiprogramarea, așteptarea multiplă, etc. cu ajutorul cărora s-au creat instrumente pentru realizarea Procedurii de exploatare.

S-a urmărit pe cît posibil ca la fiecare proces prezentat anterior în conceptul teoretic să-i corespundă cîte o sarcină.

Sarcina este un program, o procedură care îndeplinește o funcție precizată și care prezintă o anumită autonomie fiind definită prin adresa primei instrucțiuni.

Sarcinile au fost create în principal pentru a permite realizarea legăturii între calculatoare.

Produsul program corespunzător Procedurii de exploatare a fost realizat modular și parametrizat creînd o interfață software între sarcini, bine precizată și delimitată și păstrînd în parame-

tri valorile susceptibile de modificat în sarcini, în special în cele de lectură și scriere, ceea ce va permite numai modificarea unei declarații în începutul secțiunii (sarcinii) fără a se interveni în corpul programului.

La conceperea, proiectarea și implementarea sa, prezentată în continuare în acest capitol, Procedura de exploatare a păstrat structura funcțională stabilită în partea teoretică a lucrării; astfel că produsul program corespunzător se descompune în trei mari părți, fiecare parte înglobând un număr de sarcini specifice:

- Comunicația cu utilizatorul (operatorul).

. Recepția și interpretarea mesajelor de la mașina de scris pupitru - DEMS.

. Ieșirea mesajelor pe mașina de scris - pupitru (în acest caz sarcina nu a fost elaborată separat funcțiunea respectivă fiind preluată de alte sarcini) - SCREMS.

- Emisia

. Lectura în fișiere - LECTC.

. Formarea mesajelor de transmis în linie - FLL.

. Emisia mesajelor în linie - EML.

- Recepția

. Recepția și interpretarea mesajelor din linie - REC.

. Scrierea în fișiere - SCRIMPR.

Sarcinile Procedurii de exploatare așa cum au fost concepute și proiectate vor fi descrise în continuare.

## RECEPȚIA ȘI INTERPRETAREA MESAJELOR DE LA MAȘINA DE SCRIS-PUPITRU

Comunicația ce Sistemul de exploatare pentru legătura UC - UC se realizează cu ajutorul mesajelor emise de la mașina

de scris-pupitru.

Principala funcțiune în cadrul Recepției și interpretării mesajelor este realizată de sarcina Decodificare Mesaje Mașină de Scris - DMMS.

Pentru recepția mesajelor de la mașina de scris -pupitru se dispune de macroinstrucțiunea SXR care permite atunci când evenimentul se produce, trecerea la o secvență de retur de excepție.

Acesta conține în esență o punere la zero a Blocului Eveniment Recepția Mesajului mașină de scris -BERM care va semnala sarcinii Decodificare Mesaje Mașină de Scris-DMMS că a avut loc o intervenție a operatorului; se returnează apoi la secvența întreruptă și se validează returul de excepție cu ajutorul macroinstrucțiunii VXR.

După terminarea secvenței întrerupte se dă controlul sarcinii Decodificare Mesaje Mașină de Scris-DMMS cu prioritatea cea mai ridicată care interpretează mesajele de la mașina de scris-pupitru cu ajutorul limbajului de comandă.

Sarcina-DMMS începe prin testarea primului caracter al mesajului primit; dacă este ":" se poziționează pe eveniment sosit Blocul Eveniment Mașină de Scris -BEMS care semnalează sarcinii de Formare a Mesajelor de transmis în Linie -FML că trebuie luat în atenție un element pentru subcanalul mașină de scris, iar dacă primul caracter este "x" se trece la interpretarea comenzii.

Sarcina dispune în acest din urmă caz de un tabel în care sînt aranjate numele comenzilor admise; dacă o comandă a fost recunoscută atunci se branșează la secvența corespunzătoare.

Dacă primul caracter nu este ":" sau "x" sau în caz că sintaxa comenzii nu e conformă cu cea specificată se emite un mesaj de eroare.

Acest mod de a proceda permite extinderea limbajului de comandă sau modificarea parametrilor unei comenzi fără a reorganiza în ansamblu sarcina DMMS; este suficient să se adauge

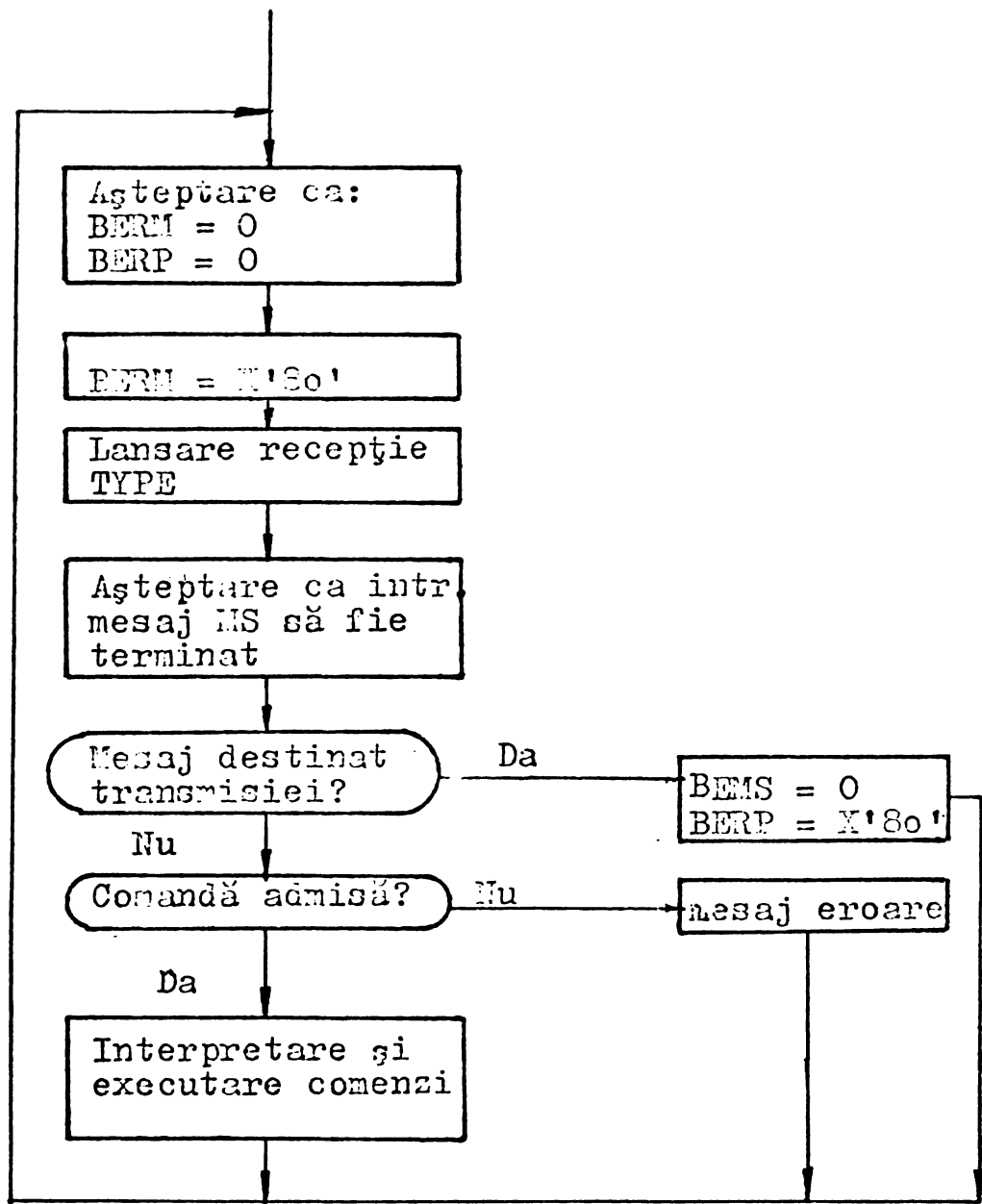


Fig.4.2. Algoritmul sarcinii Decodificare Mesaje Mașină de Scris - DMS.



o intrare în tabelul numelor de comenzi și a secvenței de interpretare a comenzii.

Dacă se dorește a se reîncepe procesul, utilizatorul va trebui să apese din nou pe butonul "întrerupere pupitru".

Algoritmul sarcinii Decodificarea Mesajelor de la Mașina de Scris -DMMS este arătat în figura: 4.2.

Sarcina DMMS comunică cu alte sarcini prin intermediul blocurilor eveniment care indică anumite stări:

- BERP, Bloc Eveniment Recepție Permisă, tamponul de recepția mesajelor mașină de scris pupitru este liber sau ocupat;
- BERM, Bloc Eveniment Recepție Mesaj mașină de scris, a existat sau nu o intervenție a operatorului;

#### LECTURA ÎN FISIERE

Sarcina de LECTură în fișierul de Cartele-LECTC, are ca obiectiv formarea unui element în tamponul component. Pentru realizarea practică s-a considerat ca sursă de date la intrare fișierul pe cartele. În momentul activării sarcina găsește în șirul său de așteptare-SALECTC informații care îi sînt necesare ca în figura: 4.3.

adresa de început a tamponului	
nr. subcanal	lung. tampon

Fig.4.3. Imaginea șirului de așteptare SALECTC al sarcinii LECTC la activarea acesteia.

În urma umplerii tamponului component, subcanalul corespunzător este plasat în SIRul subcanalelor Pregătite -SIRP; se pune în același timp pe zero Blocul Eveniment Nou Sub-Canal-BENSC.

Organizarea șirului SIRP este arătată în fig:4.4.

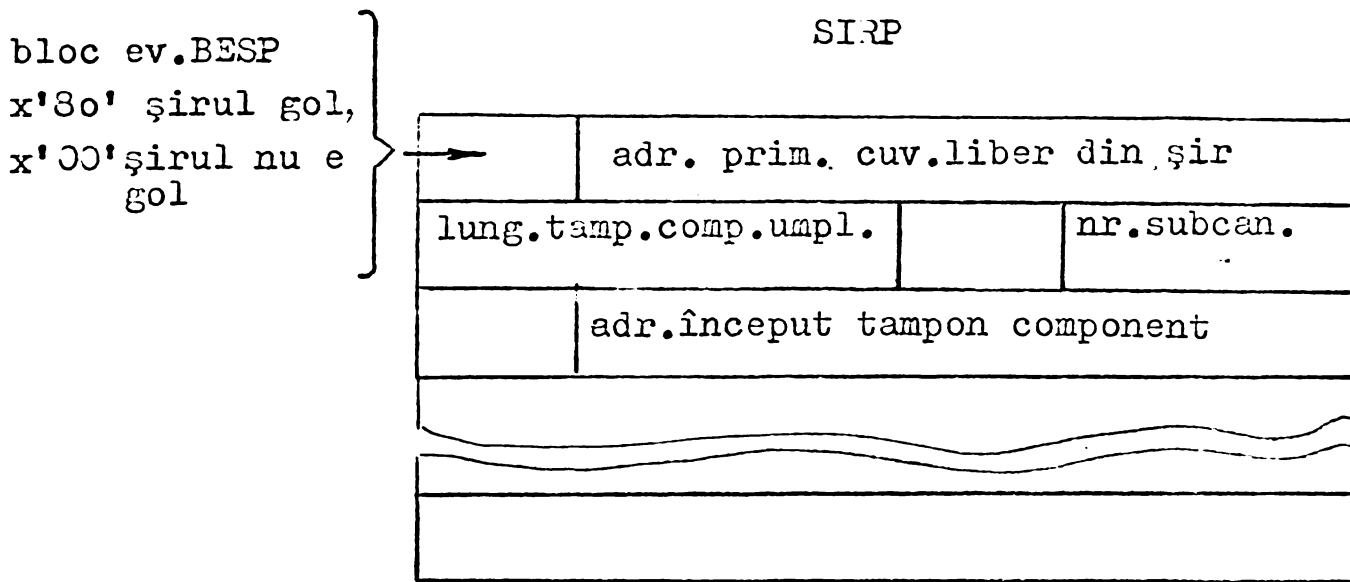


Fig.4.4..SIRul de Subcanale Pregătite - SIRP

Fiecare intrare în șirul SIRP comporta două cuvinte, primul cuvânt indică lungimea în număr de octeți a elementului și numărul de subcanal, al doilea precizează adresa de început a elementului.

Blocul Eveniment al șirului Subcanale Pregătire - BESP este asociat la condiția SIRP gol. Dacă un sfârșit de fișier a fost detectat, i se păstrează imaginea în tabela Sfârșit de FISiere - SFIS plasând în tabelă numărul de subcanal corespunzător ca în figura: 4.5.

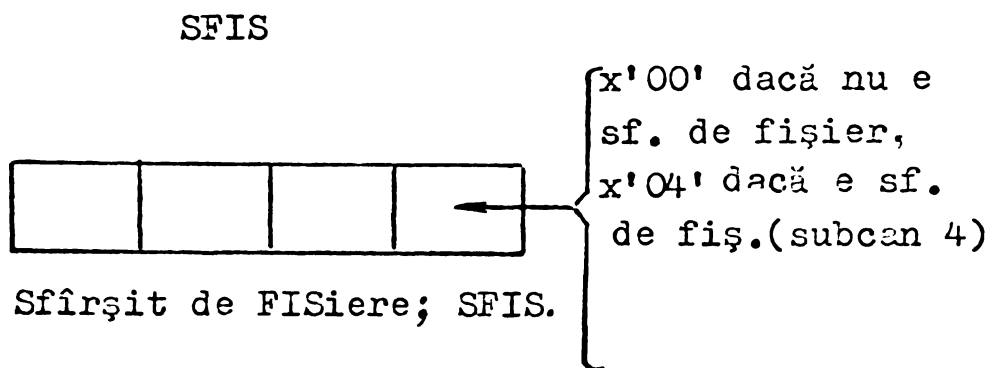


Fig.4.5. Tabela de Sfârșit de FISiere; SFIS.

Imaginea tamponului component ca urmare a executiei sarcinii LECTC este arătată în figura : 4.6.

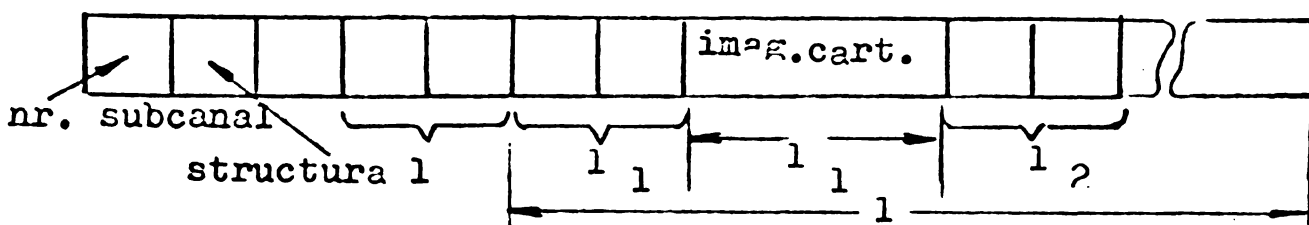


Fig.4.6. Imaginea tamponului component ca urmare a executiei sarcinii LECTC

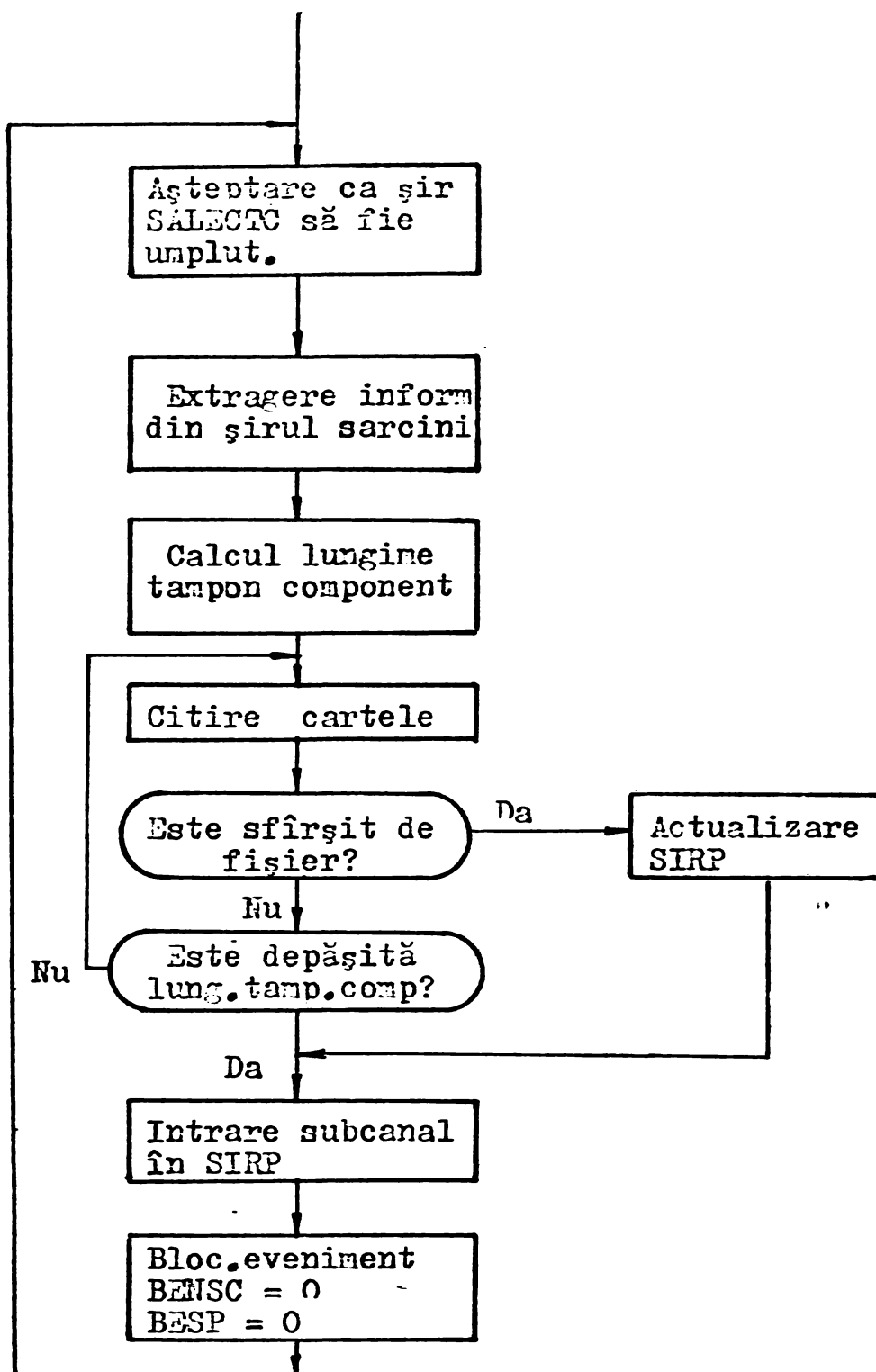


Fig.4.7. Algoritmul sarcinii LECTură în fișierul Cartele - LECTC.

Algoritmul sarcinii LECTC este prezentat în figura:4.7.

Sarcina de lectură în fișiere -LECTC comunică cu alte sarcini prin intermediul următoarelor blocuri de evenimente care indică :

- SALCTC. Sirul de Așteptare al sarcinii LECTC;
- BENSC. Bloc Eveniment Nou Sub-Canal, nou subcanal a intrat sau nu în șirul SIRP.
- BESP. Bloc Eveniment Sir subcanale Pregătite, există sau nu subcanale în SIRP.

#### FORMAREA MESAJELOR DE TRANSMIS IN LINIE

Sarcina Formarea Mesajului în linie -FML prepară înlănțuirea datelor și comenzilor pentru transmisia acestora în linie.

Datele și comenzile sînt înregistrate în Tabela de legături de înLANȚuire - TLANT care este dată în fig. 4.8 .

#### TLANT

adresa primului cuvînt liber
0
adresa de început zonă COMTRAN
lungime comenzi
0
adresa început zonă mesaje
lungime mesaje operator
nr. subcanal SURSA de activat
adr. început zonă în lanț
lungimea zonă înlănțuit

Fig.4.8. Tabela de legături de înLANȚuire -TLANT

Intrarea în TLANT comportă trei cuvinte: primul cuvînt conține numărul subcanalului, al doilea cuvînt conține adresa de început a zonei de înlănțuit și al treilea cuvînt conține lungimea zonei. Primele șase cuvinte în TLANT sînt fixe, corespunzînd zonei de comenzi și zonei de mesaje operator.

Sarcina începe execuția sa prin așteptarea ca un eveniment dintre următoarele evenimente să sosească:

- Existența unor comenzi în zona COMenzilor de TRANsmis - COMTRAN adică blocul eveniment BECA=0;
- Existența unui element pentru subcanalul mașină de scris, blocul eveniment BEMS=0;
- Aviz de Confirmare Corectă- ACC a fost primită după ultima execuție a sarcinii, adică blocul eveniment ACCREC=0;
- Un nou subcanal a intrat în șirul de subcanale sursă pregătit, BENSC=0;
- Șirul SIRP nu este gol, adică BESP=0;

Acest ultim eveniment nu este luat în considerație dacă subcanalul ales se găsește în SIRP sau nu a existat o comandă ACC pentru subcanalele pregătite.

Se continuă prin testarea prezenței comenzilor de transmis și a elementului subcanalului mașină de scris ; datele de în-lănțuit sînt eventual înscrise în TLANT.

Se examinează apoi șirul SIRP și tabela cu comenzile ACC - RECeptionate sau generate - ACCREC din fig.4.9.încercînd a găsi un subcanal pregătit pentru transmisie.

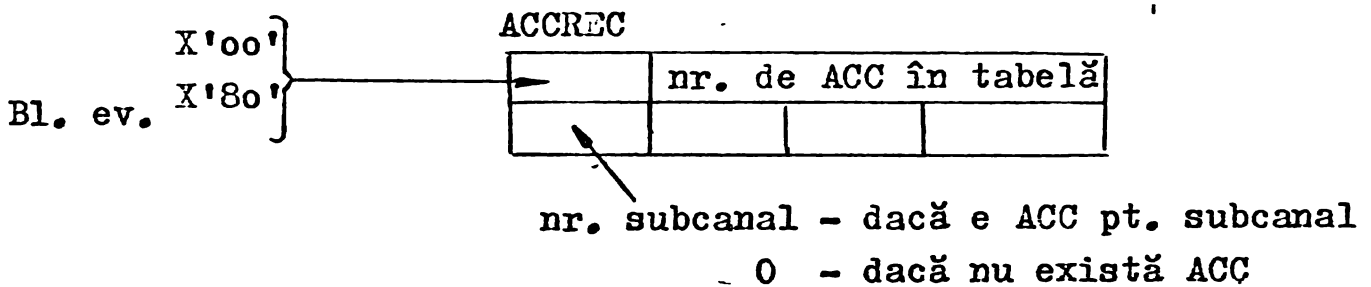


Fig.4.9 .Tabela de comenzi ACC-RECeptionate sau generate - ACCREC.

Se admite ca un element poate să fie: un început, o urmare sau sfârșit de articol, un articol complet sau mai multe articole.

Cînd un subcanal a părăsit șirul SIRP numărul său este înscris în partea subcanale activate, din TLANT, ceea ce înseamnă că sarcina de lectură a fișierului va fi relansată, atunci cînd tamponul component va redeveni disponibil.

Se remarcă că zona comenzilor de transmis și tampoanele mașinii de scris MS - component și recepție se găsesc indisponibile pînă la sfârșitul transiterii în linie a mesajului.

Dacă se dorește să se accelereze funcționarea, trebuie prevăzut două zone pentru a forma comenzile de transmis care vor lucra în basculă; cînd un singur subcanal sursă este activ, i se va asocia două tampoane component.

Sarcina de Emisie în linie - EML poate fi lansată datorită punerii la 0 a Blocului Eveniment Mesaj Format-BEMF.

Blocurile evenimente care servesc la sincronizarea sarcinii FLI cu alte sarcini, sînt următoarele și indică:

ACCREC, Bloc eveniment și tabel, un ACC a fost sau nu primit sau generat;

BENSC, Bloc Eveniment Nou SubCanal, este sau nu un nou subcanal în șirul SIRP;

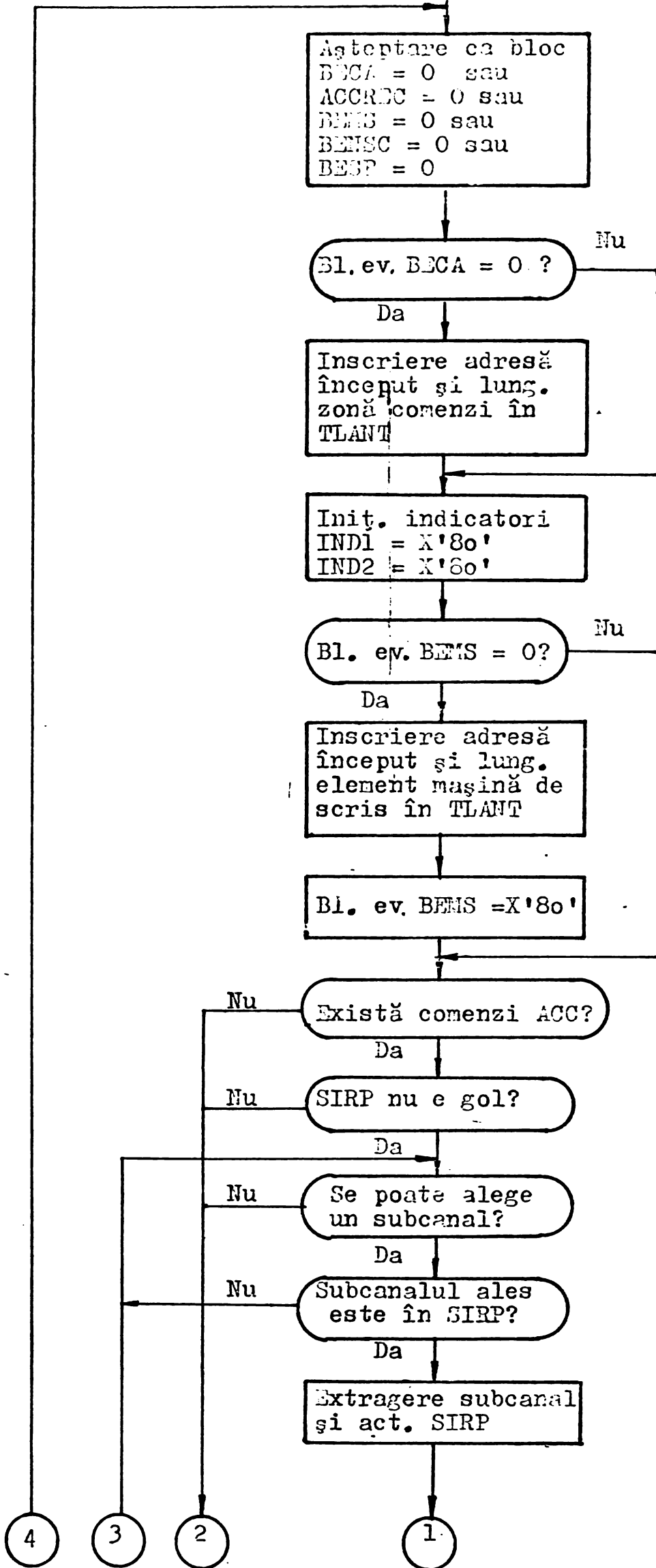
BESP, Bloc Eveniment Subcanale Pregătire - șirul SIRP este sau nu este gol;

BECA, Bloc Eveniment Comenzi Amplasate în zona COMTRAN, există sau nu comenzi de transmis;

BEMS, Bloc Eveniment Mașină de Scris, există sau nu un element pentru subcanalul MS;

BEMF, Bloc Eveniment Mesaj Format, mesajul este sau nu format pentru transmisie;

În fig.4.10. se reprezintă algoritmul sarcinii Formarea Mesajului de emisie în linie - EML.



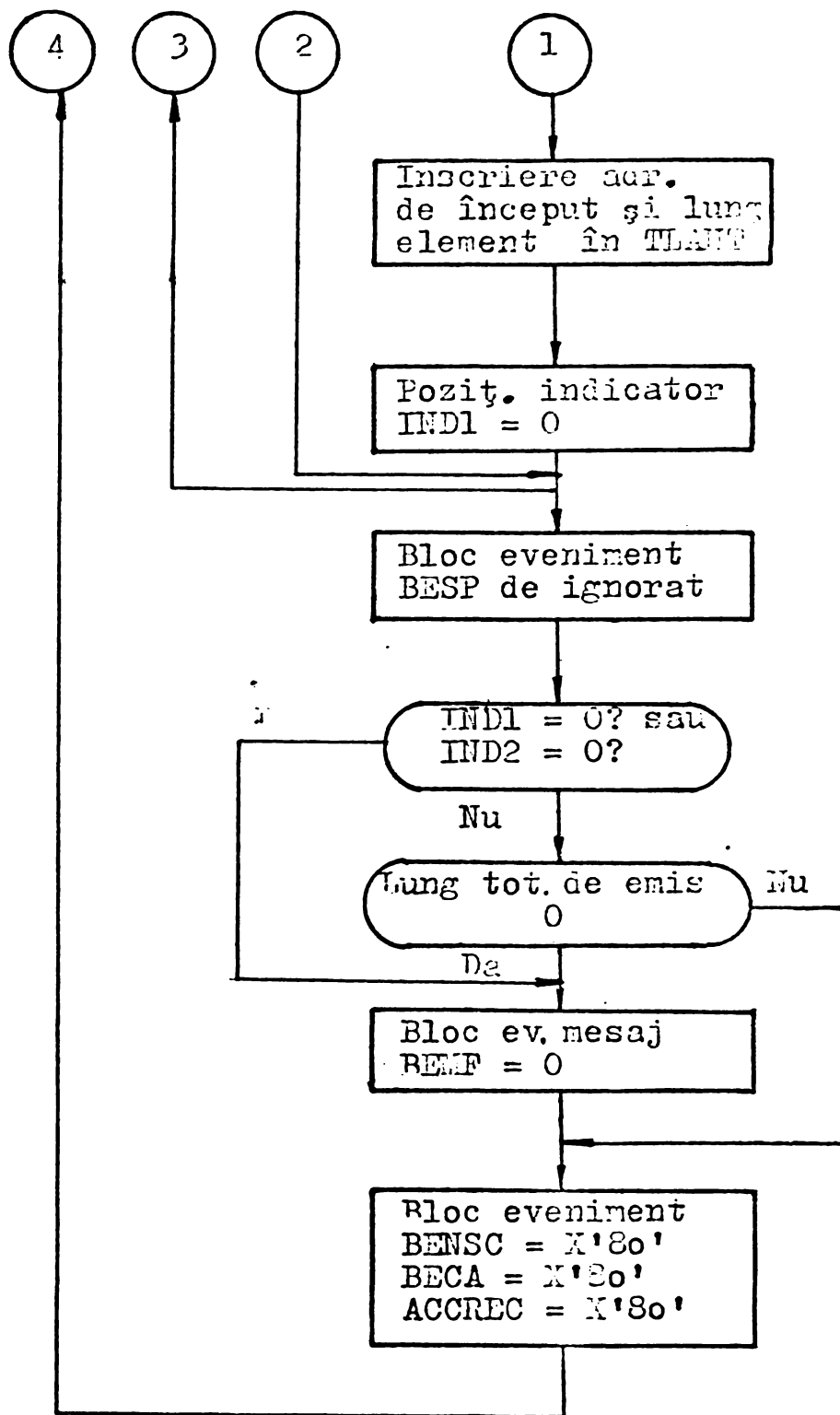


Fig.4.10 Algoritmul sarcinii de Formare Mesaje de emisie în Linie - FML.



## EMISIA MESAJELOR IN LINIE

Sarcina Emisiei Mesaje în Linie-EML este activată când Blocul Eveniment Mesaj Format (BEMF) este la 0. Se examinează tabela de legături de înlănțuire TLANT; părțile componente ale mesajului sînt transferate în tamponul de emisie în linie TAEL și astfel se activează subcanalul autorizînd o nouă intrare de mesaj de la mașina de scris punînd pe 0 Blocul Eveniment Recepție Permisă-BERP și repositionînd BEMF pe X'80'.

Tabela TLANT este actualizată după extragerea subcanalelor care vor forma mesajul de transmis.

Se indică deasemenea unele sarcini de umplere a tamponului component - lectura în fișier, în Tabela de SubCanale Active - TSCAS.

Emisia efectivă pe linia de telecomunicație se realizează cu ajutorul macroinstrucțiunilor Sistemului de gestiune a transmisiilor -SGT.

Pentru teletransmisia datelor s-a folosit procedura pentru legătura între două calculatoare TMM - UC2.

S-a realizat o legătură alternată (half-duplex) folosind o cale a cuplorului CTQM.

Algoritmul sarcinii de Emisia Mesajelor în Linie-EML este arătată în fig. 4.11.

## RECEPTIA SI INTERPRETAREA MESAJELOR DIN LINIE

RECEPTIA și interpretarea mesajelor din linie -REC are ca obiective; primirea mesajelor din linie, analiza și interpretarea acestora.

Recepția mesajelor din linie se realizează cu ajutorul Sistemului de gestiunea transmisiilor -SGT. Se lansează un polling în linie și se execută o așteptare.

Mesajul recepționat din linie este compus ca în fig. 4.12.

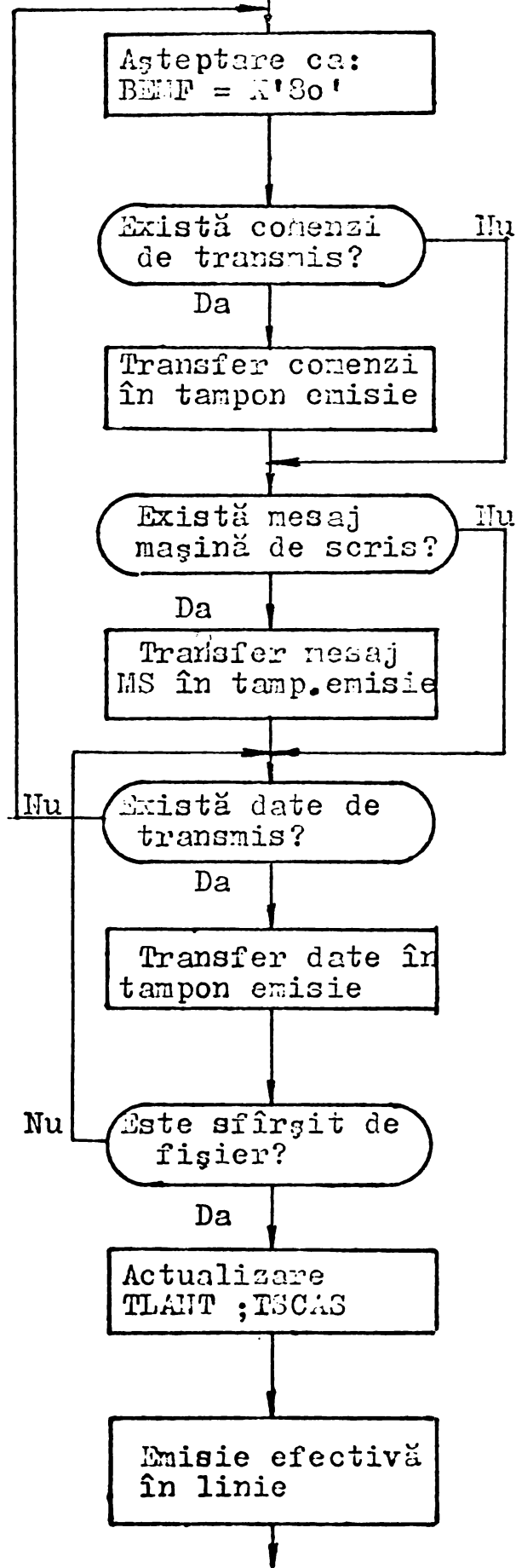


Fig.4.11 Algoritmul sarcinii Emisie Mesaje în Linie - EML.

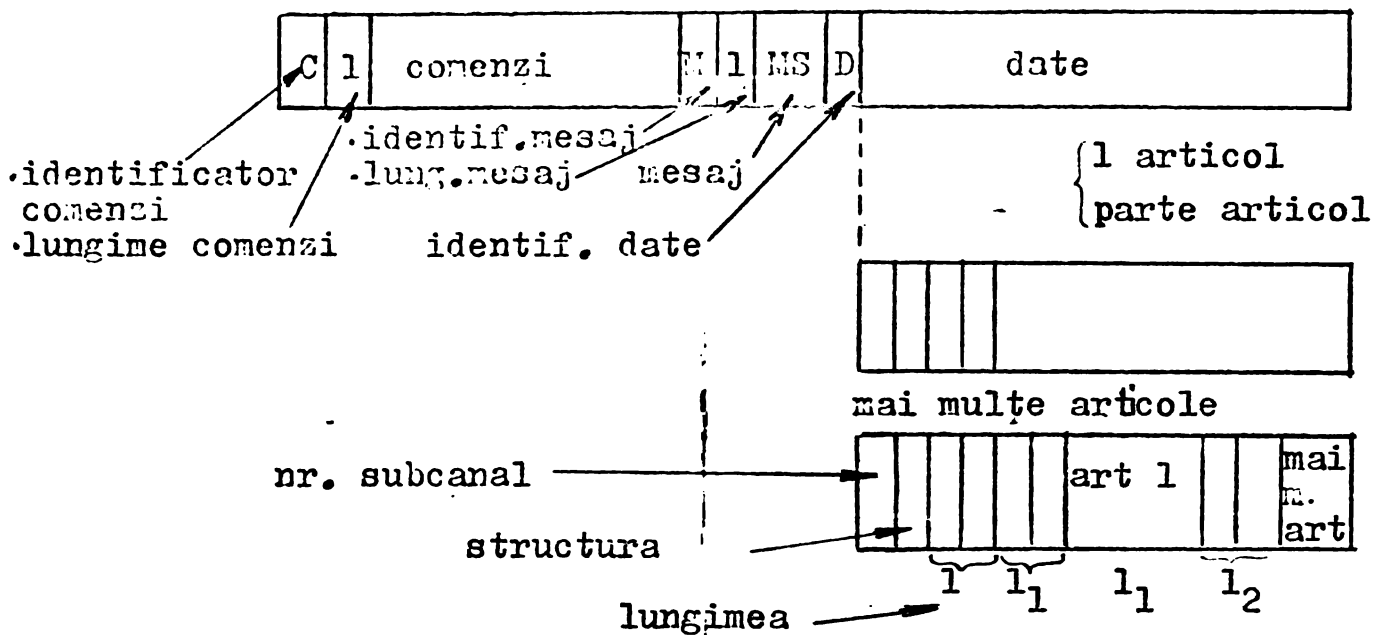


Fig.4.12. Mesajul recepționat din linie

Cînd un mesaj a fost primit Sarcina de Recepție din Linie-REC începe execuția sa prin examinarea primului octet din mesaj; dacă acesta este C-Comenzi se interpretează conținutul zonei de comenzi, iar dacă este D - Dată se semnalizează că este partea de date a mesajului. În funcție de valoarea octetului structură se transferă partea utilă a elementului, în tamponul component al subcanalului receptor.

Dacă tamponul component conține un articol complet sau mai multe articole, se continuă prin a se căuta în Tabela SubCanalelor Active Receptor-TSCAR numele sarcinii de golire a tamponului corespunzător - scriere în fișier. Fiecare din aceste sarcini dispune de un șir de așteptare unde se plasează adresa de început a elementului după octetul structură în tamponul de recepție linie TARL și numărul de subcanal ca în fig. 4.13.

Nr. subcanal	adresa elementului
--------------	--------------------

Fig.4.13. Imaginea șirului de așteptare a sarcinii de ieșire în momentul activării.

Elementul subcanalului mașina de scris -pupitru este tratat aparte. Se recunoaște prin numărul său de subcanal care îi este rezervat și este transferat într-un Tampon Mașină Scris Ieșire -TMSI; adresa acestui tampon este apoi plasată în șirul de așteptare al sarcinii de ieșire a mesajului pe mașina de scris, SCRiere pe Mașina de Scris -SCRMS.

Modul în care se transferă elementul în tamponul component variază în funcție de octetul de structură și se arată în figura 4.14.

- Un început de articol, un articol complet sau mai multe articole; acestea se transferă începând cu octetul de structură.

- O urmare de articol sau un sfârșit de articol; acesta se transferă începând cu partea articol.

-- Lungimea totală a articolelor este reconstituită în tamponul component

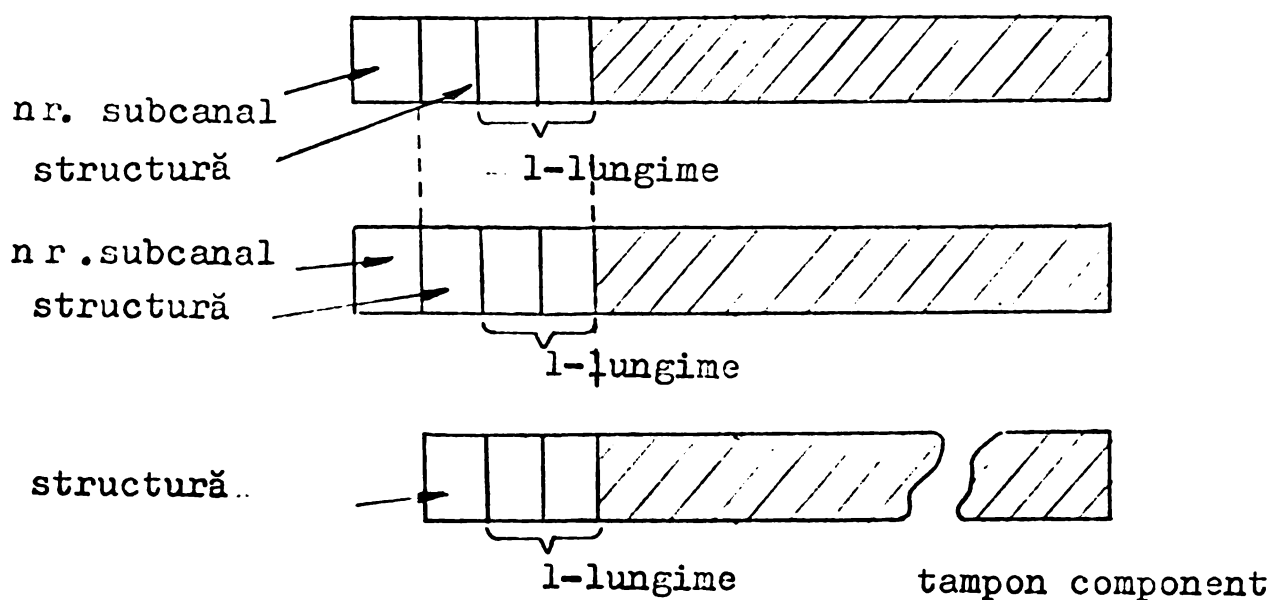


Fig.4.14. Reconstituirea la recepție a articolelor în tamponul component.

Funcționarea sarcinii de Recepție și Interpretare mesaje din linie este arătată în fig. 4.15.

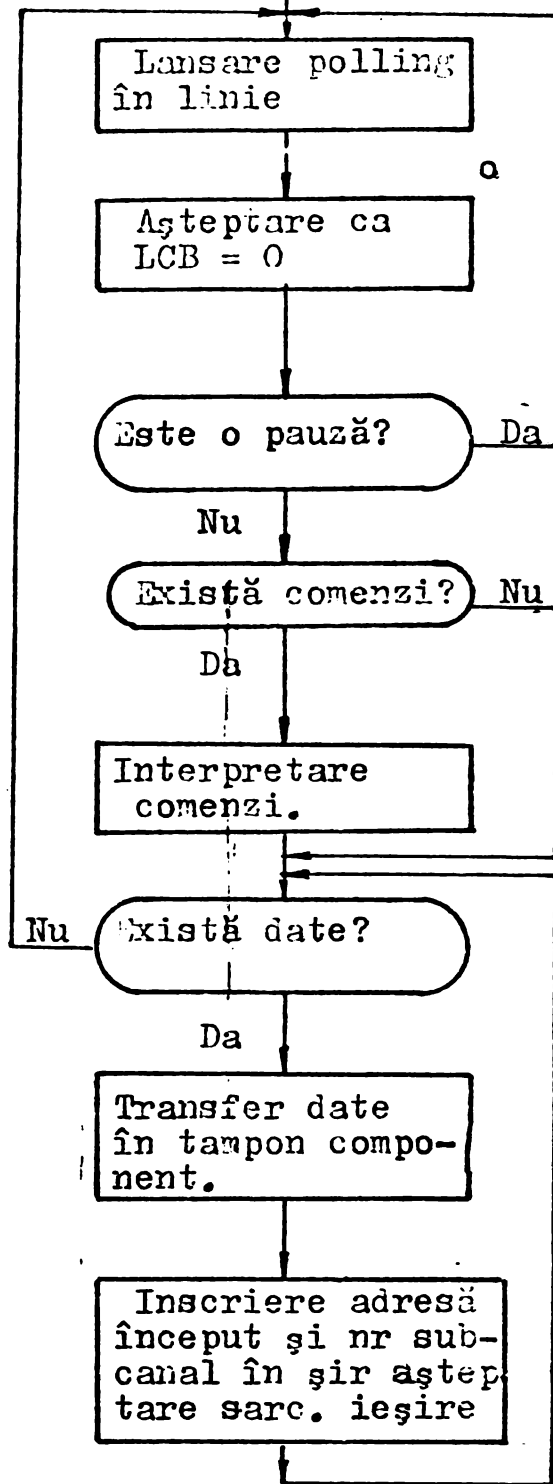


Fig.4.15.Algoritmul sarcinii RECeptie mesaje din linie - REC.

Sarcina se servește ca mijloc de sincronizare mutuală de șirurile de așteptare ale sarcinilor de ieșire și de unele blocuri eveniment.

### SCRIEREA IN FIȘIERE

Sarcina de ieșire, în acest caz SCRiere la IMPrimantă - SCRIMPR, deoarece s-a ales component receptor - imprimanta găsește la activare în șirul său de așteptare, adresa octetului structură în tamponul component și numărul subcanalului de care aparține elementul.

Nu se activează o sarcină de ieșire decât dacă articolul a fost reconstituit și deci octetul său de structură nu poate avea decât valoarea, un articol sau mai multe articole.

Utilizarea rezonabilă corespunde la articole de lungimi inferioare capacității imprimantei care este de 132 caractere.

Algoritmul sarcinii SCRiere pe IMPrimantă (SCRIMPR) este prezentată în fig.4.17.

Interliniile și salturile de pagină sînt comandate printr-un octet de salt S plasat înaintea primului caracter de imprimat, în fig. 4.16.

octet	linie de imprimare
salt	

Fig.4.16.Ieșirea pe imprimantă

Dacă totul s-ar desfășura normal, sarcina de ieșire prepară și generează o Avizare de confirmare corectă - ACC pentru subcanalul respectiv. Acest ACC este introdus în zona COMenzilor de TRANsmis- COMTRAN.

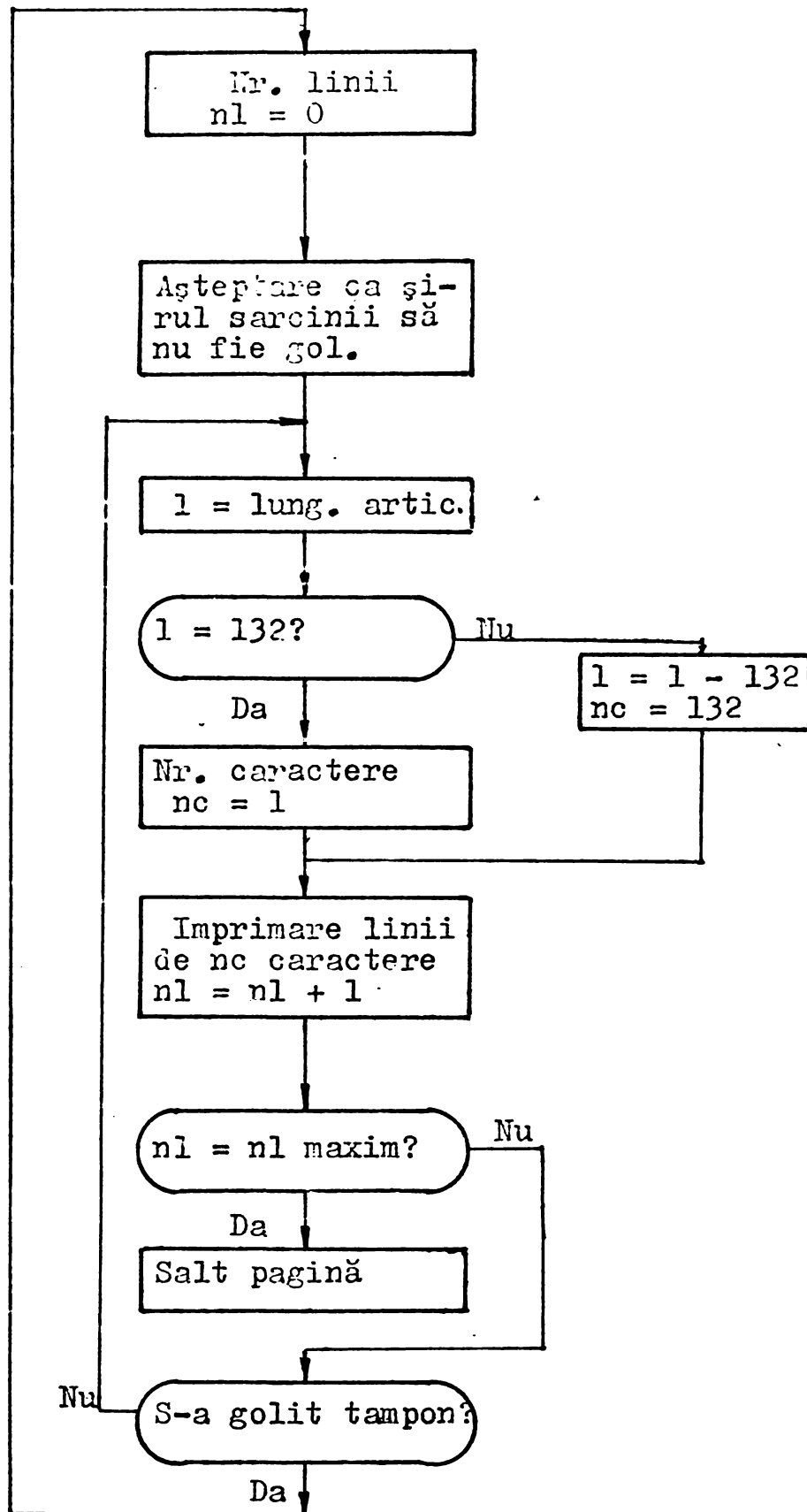


Fig.4.17. Algoritmul sarcinii de SCRiere pe IMPRimantă - SCRIMPR.

#### 4.3. SISTEMUL DE GESTIUNE SARCINI

Sistemul de Gestiune Sarcini, componentă importantă a Sistemului de exploatare pentru legătura UC-UC, este conceput pentru a fi realizat și finalizat printr-un produs program care face parte, independent sau împreună cu celelalte produse program elaborate în această lucrare, din Sistemul de exploatare al calculatorului Felix.

Are funcția principală de gestionare și controlare a derulării mai multor sarcini în paralel; pentru aceasta utilizează posibilitatea calculatorului de simultaneitate a operațiilor de prelucrare cu cele de intrare/ieșire locale și de teletransmisie precum și tehnicile de sincronizarea sarcinilor prin punerea în așteptare a acestora pînă la sosirea unor evenimente exterioare sau interioare.

Ca urmare, putem astfel asigura cu acest Sistem de gestiunea sarcinilor, posibilitatea unei legături UC-UC între calculatoare și de asemenea permite o exploatare optimă a resurselor de timp și echipament.

Sistemul de gestiunea sarcinilor constituie o contribuție personală importantă, cu caracter original, în ceea ce privește concepția, proiectarea tehnică, implementarea și punerea în funcțiune a unui executiv pentru teleprelucrare ce poate face parte, datorită calităților sale, din clasa sistemelor de operare. Oferă posibilități de extindere și generalizare și în alte domenii prin adaptări și modificări largi astfel sfera software-ului de bază al calculatorului Felix sau al altor familii de calculatoare.

Am considerat necesară și oportună realizarea unui Sistem de gestiunea sarcinilor, pentru a satisface în totalitate cerințele unei legături UC-UC în condițiile calculatorului Felix C-256 care nu dispune de un sistem pentru coordonarea și gestionarea unui ansamblu de sarcini (programe) care să poată să se deruleze în paralel. Astfel, multi-tasking-ul SGT-ului nu este corespunzător pentru acestea prezentînd inconveniente privind descompunerea în sarcini și gestionarea lor. Aceasta, deoarece o sarcină în SGT este legată obligatoriu de o linie și unei linii îi corespunde numai o sarcină; de asemenea sarcinile sînt create cu o prioritate fixă care nu poate fi schimbată. Anterior au fost prezentate restricții esențiale pentru lucrarea noastră, datorate SGT și SGF. Sistemul de gestiunea sarcinilor a fost conceput să lucreze cu sarcini diferite, de altă natură și aceasta pentru a realiza funcțiile complexe ale legăturii între calculatoare.



Am avut în vedere că Sistemul de gestiunea sarcinilor în funcționarea sa, să gestioneze sarcinile care se găsesc în stări diferite ca:

- starea de activare, când sarcina este luată în evidență de SGS;
- starea de așteptare, când sarcina așteaptă un eveniment intern sau extern, adică un sfârșit de intrări/ieșiri clasice sau de teletransmisie;
- starea de pregătire, când sarcina poate fi lansată în execuție;
- starea de lansare, când sarcina preia controlul și trece la execuția instrucțiunilor.

Sistemul de gestiune sarcini lansează o sarcină care păstrează controlul pînă cînd este pusă în așteptare de către un eveniment extern (intrare/ieșire clasică sau de teletransmisie); aceasta permite regulatorului (element de bază al SGS) de a lansa o altă sarcină pregătită cu cea mai mare prioritate. În caz că toate sarcinile vor fi în așteptare, controlul se trece monitorului pînă la producerea unor evenimente așteptate, cînd monitorul va ceda din nou controlul regulatorului, care va relansa sarcina pregătită, cea mai prioritară.

## FUNCTIUNILE SISTEMULUI DE GESTIUNE SARCINI

Sistemul de gestiune sarcini l-am conceput și prezentat pentru a realiza un ansamblu de funcțiuni cu ajutorul unor subprograme.

Aceste funcțiuni sau subprograme a căror scheme funcționale se bazează pe algoritmi proprii sînt prezentate în continuare.

Activarea unei sarcini. Lansarea în execuție a unei sarcini necesită în prealabil activarea sa. Activarea se obține printr-un apel la subProgramul de ACTIVARE sarcini - PACTIV -, al Sistemului de gestiune sarcini care nu lansează execuția sarcini, dar crează intrarea acesteia în tabelele Sistemului de gestiune sarcini.

În timpul activării, se asigură o prioritate sarcini ( $0 \leq P \leq 255$ ) unde 0 este prioritatea maximă și 255 prioritatea minimă. În caz că mai multe sarcini au aceeași prioritate, fiind aranjate în ordinea activării, prima sarcină activată este cea mai prioritară.

Prioritățile nu sînt obligatoriu statice, ele pot evolua în cursul vieții unei sarcini.

La activarea sarcinii se precizează de asemenea, adresa și rului de așteptare a sarcinii sau se indică că nu există șir de așteptare.

Apelul la PACTIV se poate găsi oriunde în program și în particular în corpul unei sarcini (activarea unei sarcini prin alta).

Nu se pot crea două sarcini care să poarte același nume.

Lansare multitasking. Lansarea efectivă în execuție a sarcinii activate, în starea pregătită de lansare și cu cea mai mare prioritate se efectuează printr-un apel la subProgramul de INCepere Procès - PINCP al Sistemului de gestiune sarcini, sau regulatorul SGS.

Lansarea multitasking-ului are ca rol căutarea și lansarea în execuție a unei sarcini activate, pregătite cu cea mai mare prioritate la instrucțiunea care se găsește la adresa indicată de numele sarcinii; cînd o sarcină va fi pusă în așteptare sau dezactivată se va încerca lansarea altei sarcini pregătite de lansare, fie de la adresa primei instrucțiuni a sarcinii în cazul în care sarcina nu a mai fost lansată, fie de la adresa instrucțiunii imediat următoare aceleia în care sarcina a fost întreruptă (pusă în așteptare).

Punerea în așteptare a unei sarcini. Punerea unei sarcini în așteptarea unui eveniment dintre mai multe evenimente se obține printr-un apel la subProgramul de Punere în AȘTEptare SARCini - PASTESA din Sistemul de gestiune sarcini.

Se disting trei tipuri de evenimente:

- sfîrșit intrare/ieșire clasică, eveniment SGF;
- sfîrșit intrare/ieșire transmisie, eveniment SGT;
- sfîrșit intrare/ieșire, cerut de macroinstrucțiunea TYPE a Monitorului.

Punerea în așteptare a unei sarcini determină predarea controlului de către sarcina pusă în așteptare, altei sarcini; la sosirea evenimentelor așteptate, aceasta devine din nou pregătită pentru lansare.

Introducerea unui cuvînt în şirul de aşteptare al unei sarcini. La activarea unei sarcini există posibilitatea asocierii la aceasta a unui şir de aşteptare. În acest şir de aşteptare alte sarcini pot plasa informaţii de care are nevoie la executia sa, existînd astfel posibilitatea de comunicaţie între sarcini.

Sirurile de aşteptare sînt definite şi gestionate de algoritmul F.I.F.O. (First In First Out) - "primul venit, primul servit" din cadrul Sistemului de operare al calculatorului.

Pentru a introduce un cuvînt în şirul de aşteptare al unei sarcini se face apel la subProgramul de Introducere Cuvînt în Sirul de Aşteptare al sarcinii - PICSA al Sistemului de gestiune sarcini.

Extragerea unui cuvînt din şirul de aşteptare al unei sarcini. Sirul de aşteptare fiind definit şi gestionat de F.I.F.O., o sarcină nu poate cere decît scoaterea cuvîntului cel mai vechi din şir.

Extragerea unui cuvînt de aşteptare al unei sarcini este obţinută prin apelul la subProgramul de Extragere Cuvînt din Sirul de Aşteptare al sarcinii - PECSA al Sistemului de gestiune sarcini.

Schimbarea priorităţii unei sarcini. Efectuarea unei acţiuni înaintea alteia, se obţine prin acordarea de priorităţi convenabile sarcinilor, la momentul respectiv.

Schimbarea priorităţii unei sarcini se realizează printr-un apel la subProgramul de Schimbare PRIORitate sarcini - PSPRIOR al Sistemului de gestiune sarcini. Schimbarea priorităţii se efectuează de către sarcina respectivă pentru ea însăşi sau pentru altă sarcină. Sarcina este reactivată cu o nouă prioritate.

Dezactivarea unei sarcini. La încetarea activităţii unei sarcini, aceasta trebuie dezactivată - respectiv scoasă din tabelele Sistemului de gestiune sarcini.

Aceasta se obţine printr-un apel la subProgramul de DezACTIVare sarcină - PDACTIV al Sistemului de gestiune sarcini.

Sarcina poate cere dezactivarea sa sau dezactivarea altei sarcini.

Realizarea Sistemului de gestiune sarcini - SGS necesită elaborarea unor tabele şi subprograme corespunzătoare care sînt prezentate în continuare.

. . . TABELELE SISTEMULUI DE GESTIUNE SARCINI

Sistemul de gestiune sarcini este constituit dintr-un ansamblu de tabele necesare îndeplinirii funcțiilor sub-programelor.

Tabelele sînt definite și inițializate dinamic în programul principal, în funcție de numărul maxim de sarcini necesare aplicației.

Informațiile privind sarcinile controlate de către Sistemul de gestiune sarcini, sînt cuprinse în tabele:

- Tabela de sarcini ACTIVE-TACT;
- Tabela de PRIORități sarcini - TPRIOR;
- Tabela de GESTIune sarcini - TGEST;
- Tabela de sarcină LANSată - TLANS;
- Tabela de SARCini în ASteptare - TSARAS.

Tabela de sarcini active - TACT, se prezintă în figura 4.18 .

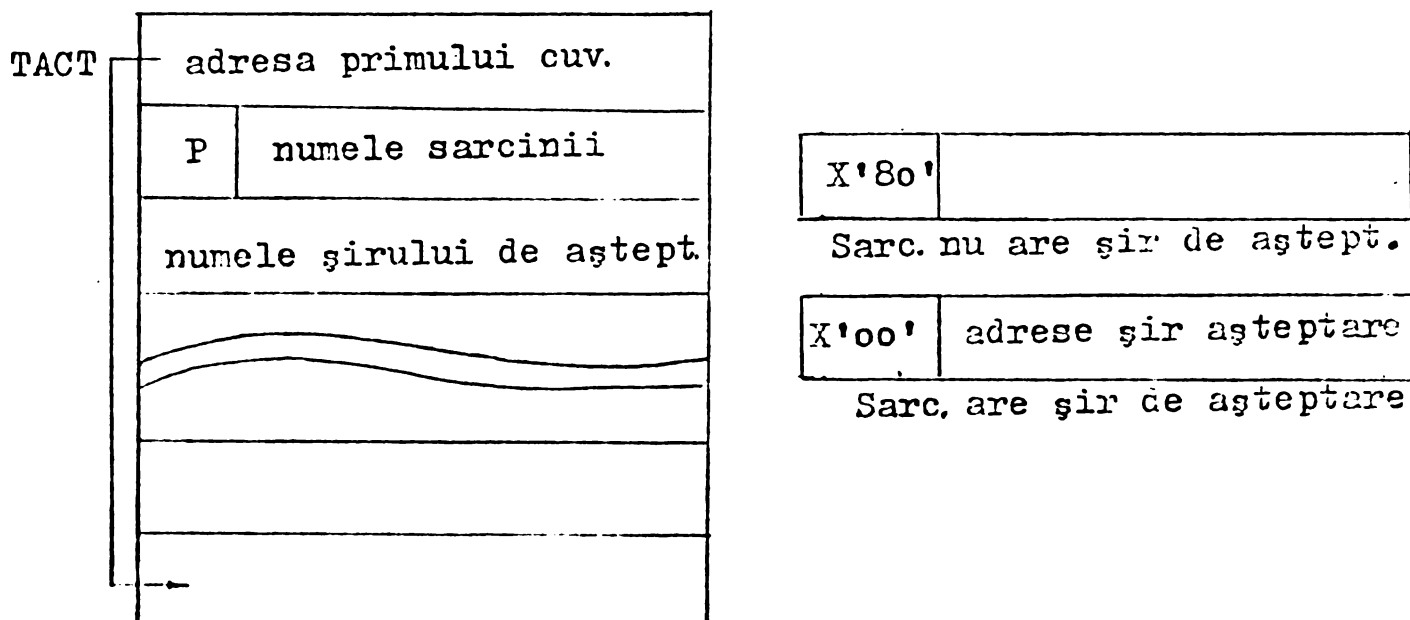


Fig.4.18.Tabela de sarcini active TACT.

Pentru o sarcină activată sînt necesare două cuvinte; sarcinile sînt aranjate în ordinea activării.

Tabela de priorități sarcini -TPRIOR se prezintă în figura 4.19.

TPRIOR	adr. primului cuv. liber	
	P	numele sarcinii
	adr. imag R3 în TGEST	
	P	numele sarcinii

1 intrare

Fig. 4.19. Tabela de priorități sarcini TPRIOR

În TPRIOR adresa imaginii registrului 3 în TGEST permite a se cunoaște numărul de evenimente așteptate de sarcină. În caz că numărul de evenimente este 0, sarcina este în stare pregătită.

Tabela de gestiune sarcini TGEST este prezentată în fig. 4.20.

TGEST	adr. primului cuv. liber	
	imag. registrelor $R_0-R_{11}$	
	numele sarcinii	

1 intrare

Fig. 4.20. Tabela de gestiune sarcini TGEST

La fiecare sarcină se asociază o zonă de lucru a Sistemului de gestiune sarcini. Această zonă este destinată a primi imaginile primelor 12 registre ale sarcinii, în momentul când aceasta execută funcția ei de așteptare.

Tabela de sarcină lansată TLANS se prezintă în fig. 4.21.

TLANS	numele sarcinii	
	adr. zonei de lucru în TGEST	
	nr. max de sarc.	nr de sarc. activate

Fig. 4.21 Tabela de sarcină lansată TLANS

Tabela de sarcini în așteptare TSARAS se prezintă în fig. 4.22.

TSARAS	adr. primului cuv. liber		
	nr. de sarc. în așteptare		
TSARAS + 96	nume sarcină 1		
		1	k
	adresa octet de eveniment 1 pentru sarcina 1		
	maxim 20 adrese octet eveniment pentru o sarcină		

1 intrare

Fig.4.22.Tabela de sarcini în așteptare TSARAS

În tabelă sînt indicate numărul de sarcini în așteptare și elementele caracteristice pentru fiecare sarcină (nume sarcină, număr de evenimente așteptate, număr de evenimente posibile, adresele octeților de eveniment ale TDF, LCB, TWB).

Pentru macroinstrucțiunea MLTW (așteptarea evenimentelor multiple) a monitorului se crează o tabelă TABEV pentru sarcina pusă în așteptare care este apoi încărcată în tabela TSARAS. în fig.4.23.

TABEV	0	1	k
adresa bloc eveniment			

Fig.4.23 Tabela de așteptare evenimente multiple TABEV

Unde 1 este numărul de evenimente așteptate de sarcină și k este numărul maxim de evenimente.

La fiecare eveniment așteptat corespunde adresa blocului eveniment asociat.

SUBPROGRAMELE SISTEMULUI DE GESTIUNE SARCINI

Sistemul de gestiune sarcini este compus din următoarele subprograme:

- Subprogramul PACTIV , de activarea unei sarcini
- Subprogramul PDACTIV , de dezactivarea unei sarcini
- Subprogramul PSPRIOR , de schimbarea priorității sarc.
- Subprogramul PICSA , de introducerea unui cuvînt în șirul de așteptare a unei sarc.
- Subprogramul PECSA , de extragerea unui cuvînt din șirul de așteptare a unei sarc.
- Subprogramul PASTESA , de punere în așteptare a unei sarcini
- Subprogramul PINCP , de lansarea sarcinii

În sistemul de gestiune sarcini, registrul general R8 a fost ales ca registru de apel al subprogramelor acestuia.

Subprogramul - PACTIV - activează o sarcină.

La apelul la subprogramul PACTIV registrele 3 și 4 trebuie să conțină parametri de definire a sarcinii ca în fig.4.24.

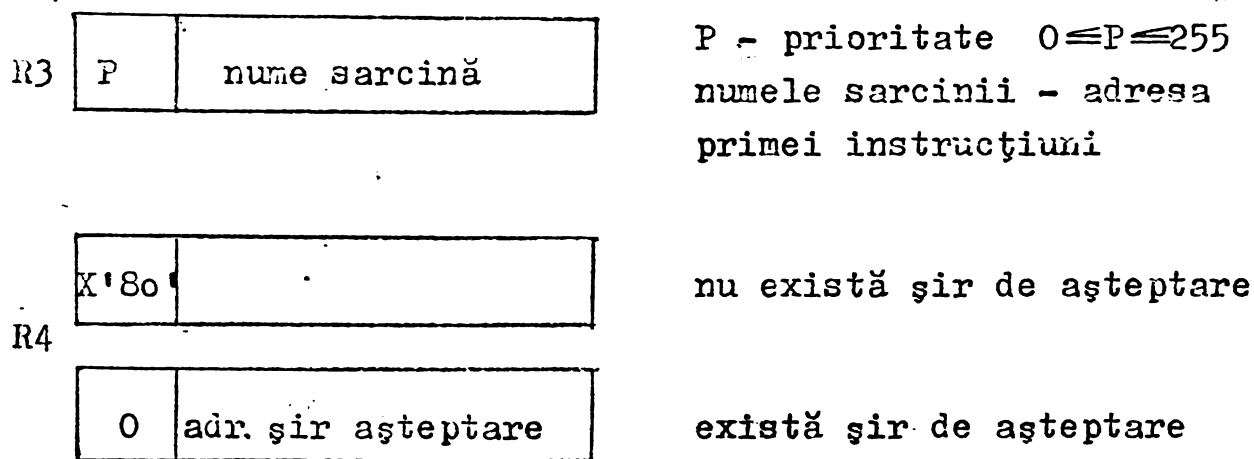


Fig.4.24. Activarea unei sarcini (conținutul registrelor 3 și 4)

Apelul subprogramului PACTIV se efectuează cu instrucțiunea: BAL, 8 PACTIV iar controlul este dat instrucțiunii care urmează.

Singurele registre modificate sînt R<sub>3</sub>, R<sub>4</sub> și R<sub>12</sub>; R<sub>12</sub> conține codul de retur. Astfel, conținutul lui R<sub>12</sub> este:

- la 0, totul s-a desfășurat normal, sarcina a fost activată;
- primul octeț la X '80', sarcina nu a fost activată deoarece maximum de sarcini a fost atins;
- primul octeț la X '01', încercare de dublă activare a unei sarcini, a doua nefiind luată în seamă.

Controlul este dat după execuție, instrucțiunii care urmează BAL, 8 PACTIV.

Subprogramul - PDACTIV - dezactivează o sarcină.

La apelul la subprogramul PDACTIV, registrul  $R_3$  trebuie să conțină parametri de definire a sarcinii. în fig.4.25.

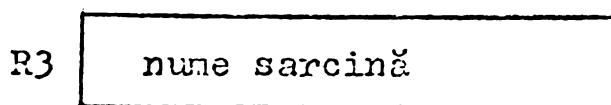


Fig.4.25. Dezactivarea unei sarcini (conținutul registrului 3)

Apelul la subprogramul PDACTIV se efectuează cu instrucțiunea BAL, 8 PDACTIV.

Singurul registru modificat este  $R_3$  care conține codul de retur. Astfel conținutul lui  $R_3$  este :

- la 0, totul s-a desfășurat normal; sarcina a fost dezactivată;
- primul octeț la X '80', sarcină necunoscută; nu s-a schimbat nimic.

Dacă sarcina se dezactivează ea însăși, se relansează sarcina pregătită cea mai prioritară.

Dacă nu mai sînt sarcini activate se dă controlul instrucțiunii care urmează BAL, 8 PINCP, în programul principal.

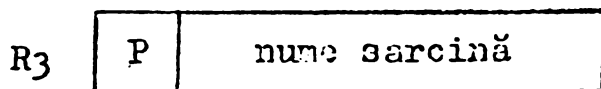
Cînd apelul la PDACTIV privește o altă sarcină decît cea lansată, se continuă secvența în corpul sarcinii.

BAL, 8 PDACTIV nu trebuie să fie programat înainte de lansarea Sistemului de gestiune sarcini.



Subprogramul - PSPRIOR - atribuie o prioritate unei sarcini

La apelul la subprogramul PSPRIOR registrul  $R_3$  trebuie să conțină parametrii de definire a sarcinii ca în fig. 4.26.



P - noua prioritate  
numele sarcinii - adresa  
primei instrucțiuni

Fig. 4.26. Atribuirea priorității unei sarcini (conținutul registrului 3).

Apelul la subprogramul PSPRIOR se efectuează cu instrucțiunea: BAL, 8 PSPRIOR.

Controlul este redat în toate cazurile instrucțiunii care urmează BAL, 8 PSPRIOR.

Registrul modificat este  $R_3$  care conține codul de retur. Astfel conținutul lui  $R_3$  este :

- la 0, totul s-a desfășurat normal; s-a atribuit o nouă prioritate sarcinii;
- primul octeț la X '01', sarcină necunoscută sau nu sînt sarcini activate.

Sarcina care își schimbă prioritatea trebuie să fi fost obligatoriu activată. O sarcină poate schimba prioritatea unei alte sarcini sau pe a sa.

Sarcina este aranjată în tabela de priorități TPRIOR.

Subprogramul - PICSA - introduce un cuvînt în șirul de așteptare a sarcinii.

La apelul la subprogramul PICSA registrele  $R_3$  și  $R_4$  trebuie să conțină numele sarcinii și respectiv, cuvîntul de introdus ca în fig. 4.27.

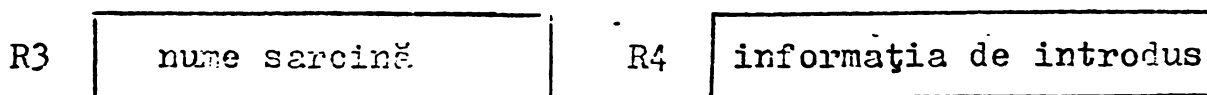


Fig. 4.27. Introducerea unui cuvînt în șirul de așteptare a sarcinii (conținutul registrelor 3 și 4)

Apelul la subprogramul PICSA se efectuează cu instrucțiunea BAL, 8 PICSA, iar după execuție controlul este dat instrucțiunii următoare.

Se modifică numai registrul R<sub>3</sub>, acesta conținând codul de retur. Astfel, conținutul lui R<sub>3</sub> este :

- la 0, totul s-a desfășurat normal; cuvânt introdus în șir de așteptare.
- primul octet la X '01', sarcină necunoscută;
- primul octet la X '02', sarcina nu are șir de așteptare;
- primul octet la X '03', șir de așteptare plin, cuvântul nu poate fi acceptat;
- primul octet la X '80', cuvântul a fost plasat iar șirul a devenit plin.

Șirul de așteptare trebuie să fie organizat ca în fig. 4.23.

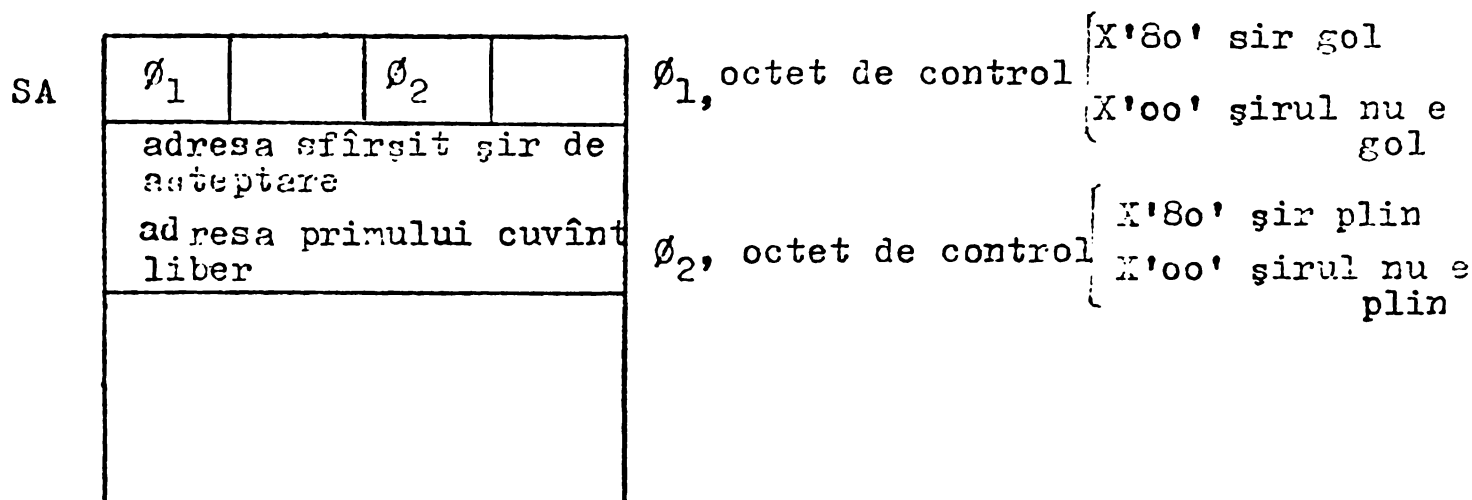


Fig. 4.23 Organizarea șirului de așteptare

In continuare, se dă un exemplu de definire a unui șir de așteptare care conține 3 cuvinte utile, fig. 4.29.

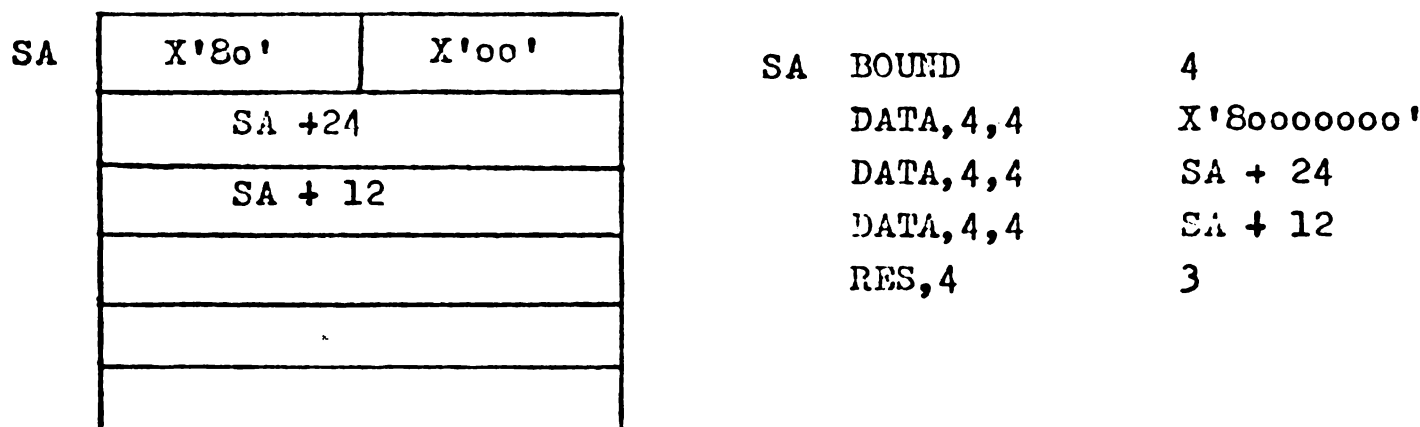


Fig. 4.29. Șir de așteptare conținând trei cuvinte utile

Subprogramul - PECSA - extrage cuvîntul cel mai vechi din şirul de aşteptare al sarcinii.

La apelul la subprogramul PECSA registrul  $R_3$  trebuie să conţină numele sarcinii.

Apelul la subprogramul PECSA se efectuează cu instrucţiunea BAL, 8 PECSA. Controlul este dat instrucţiunii care urmează BAL, 8 PECSA.

Cuvîntul extras, dacă totul se desfăşoară normal, se găseşte în  $R_4$ .

Singurele registre modificate sînt  $R_3$  şi  $R_4$ ;  $R_3$  conţine codul de retur. Astfel, conţinutul lui  $R_3$  este:

- la 0, totul s-a desfăşurat normal;
- la X '01', sarcina nu a fost găsită;
- primul octet, la X '02', nu e şir de aşteptare pentru sarcină;
- primul octet, la X '80', şirul de aşteptare e gol, numic în  $R_4$ .

Subprogramul - PASTESA - pune sarcina în aşteptarea sosirii unor evenimente.

La apelul la subprogramul PASTESA registrul  $R_3$  trebuie să conţină numărul de evenimente aşteptate şi adresa tabelii de evenimente aşteptate TEA în fig.4.30.

R3	1	adresa TEA
----	---	------------

Fig.4.30. Funerea în aşteptare a unei sarcini (conţinutul registrului  $R_3$ ).

TEA, este tabela de evenimente posibile aşteptate, iar l este numărul de evenimente aşteptate din cele k posibile aşteptate pentru sarcina care va fi pusă în aşteptare.

În cazul în care numărul evenimentelor aşteptate l este egal cu zero sarcina nu aşteaptă niciun eveniment şi este considerată ca pregătită pentru lansare.

Apelul la subprogramul PASTESA se efectuează cu instrucțiunea BAL, 8 PASTESA și determină cedarea controlului sarcinii pregătite de lansare, cu cea mai mare prioritate.

Sarcina pusă în așteptare își păstrează conținutul primelor douăsprezece registre, RO-R11, în zona de lucru a tabelii TGEST corespunzătoare sarcinii, pe care le va regăsi odată cu relansarea sarcinii.

Sarcina care efectuează un apel la PASTESA poate pierde controlul chiar dacă i-au sosit evenimentele așteptate, în caz că nu este cea mai prioritară.

Pentru punerea în așteptare a sarcinii, se definește în prealabil, în sarcină o tabelă a evenimentelor posibile așteptate TEA. Aceasta comportă câte un dublu cuvânt pentru cele k evenimente susceptibile de a se produce în fig. 4.31.

X'00'		adr LCB	eveniment SGT
X'01'	CCB <sub>n</sub>	adr abs TDF	eveniment SGF
X'02'		adr abs BE	eveniment TYPE
X'FF'			eveniment de ignorat

Fig.4.31. Tabela cu evenimente așteptate - TEA

Tabela TEA comportă un cuvânt dublu pentru un eveniment așteptat:

- primul octet al fiecărui dublu cuvânt indică tipul de evenimente;
- adresa LCB (Line Control Bloc) este adresa de control a liniei de transmisie;
- CCB<sub>n</sub>, (Command Control Bloc) este numărul blocului de gestiune directivă corespunzător operației efectuate pentru TDF;
- adresa TDF este adresa absolută a tabelii de descriere a fișierului;

- adresa BE este adresa absolută a blocului eveniment, asociat la evenimentul analizat;
- un eveniment de tip X 'FF' va fi ignorat la generarea tabelii TABEV rezultată din TEA.

Din tabela TEA subprogramul PASTESA, generează tabela TABEV și o încărcă în tabela TSARAS, pentru sarcina respectivă.

La fiecare eveniment se asociază un bloc eveniment al cărui prim octet indică dacă a sosit sau nu a sosit evenimentul:

- X'80' - evenimentul nu a sosit
- X '00' - evenimentul a sosit

Adresele octeților de eveniment se vor găsi în tabela TABEV și TSARAS.

Sistemul de gestiune sarcini nu întrerupe o sarcină decât dacă s-a efectuat în sarcină, BAL, 8 PASTESA.

Subprogramul PINCP - lansează în execuție o sarcină.

La apelul la subprogramul PINCP nu se transmit parametri.

Apelul subprogramului PINCP se efectuează cu instrucțiune BAL, 8 PINCP.

Funcțiunile îndeplinite de subprogramul PINCP sînt următoarele:

- Explorarea secvențială a tabelii de sarcini în așteptare TSARAS a SGS-ului respectiv testarea sosirii evenimentelor așteptate pentru sarcinile puse în stare de așteptare (întrerupte) prin testarea octeților de eveniment.

În cazul sosirii numărului de evenimente așteptate pentru sarcina respectivă aceasta este scoasă din tabela TSARAS și este pusă în starea pregătită de lansare în tabela TGEST a sistemului.

Acest ansamblu de funcțiuni este îndeplinit de modulul PTESTSA al subprogramului PINCP.

Căutarea în tabela SGS a sarcinii pregătite cu cea mai mare prioritate și cedarea controlului acesteia. Sarcina va fi lansată fie de la adresa primei instrucțiuni executabile în caz că sarcina nu a mai fost lansată, fie va fi relansată de la adresa imediat următoare celei care a pus-o în așteptare.

Această funcțiune este realizată de modulul PLANS al subprogramului PINCP.

În cazul că nu există sarcini pregătite de lansare partiția se blochează în stare de așteptare multiplă pe prima sarcină în așteptare din tabela TSARAS cedînd controlul altei partiții pînă la producerea evenimentelor așteptate pentru sarcină, cînd monitorul va putea ceda din nou controlul partiției și programul va putea relansa sarcina pregătită cea mai prioritară.

În cazul inexistenței de sarcini activate se dă controlul primei instrucțiuni după prima lansare a subprogramului PINCP în programul principal.

Subprogramul PINCP avînd un rol de coordonator al SGS este apelat independent atît în programul principal cît și în subprogramele PASTESA și PDACTIV.

Așa cum s-a văzut Sistemul de gestiune sarcini este compus dintr-un ansamblu de tabele și subprograme specifice; tabelele SGS fiind definite și inițializate în Programul principal.

Produsul program pentru Sistemul de exploatare, pentru legătura UC-UC începe cu un Program principal amintit mai sus ; acest Program principal cuprinde:

- definiții externe numărul maxim de sarcini NRSARC, tabelele SGS (TACT, TPRIOR, TGEST, TPLANS și TSARAS) numele fișierelor și liniilor de transmisie, adresa primei instrucțiuni care urmează după BAL, 3 PINCP (ADRRET);

- referiri externe - numele sarcinilor, adresele șirurilor de așteptare pentru acestea și numele programelor SGS apelate.

În sarcini se vor defini ca:

- referiri externe - numele șirurilor de așteptare, subprogramele SGS apelate, tabela TABEV, numele fișierelor și a liniilor de transmisie.

#### 4.4. COMENZILE SISTEMULUI DE EXPLOATARE PENTRU LEGATURA UC - UC

În Sistemul de exploatare pentru legătura UC-UC se disting în general două feluri de mesaje care sînt introduse de la mașina de scris-pupitru:

- Mesaje destinate a fi transmise la mașina de scris - pupitru a calculatorului corespondent;
- Mesaje care se adresează Sistemului de exploatare adică comenzi.

În cele ce urmează au fost prezentate numai comenzile, componentă principală a Sistemului de exploatare pentru legătura UC-UC și care au următoarele obiective principale:

- Realizarea unei legături directe fizice și logice între două calculatoare prin stabilirea unui subcanal de la intrare în calculatorul sursă pînă la ieșirea din calculatorul receptor;
- Efectuarea schimburilor de date în calculator și în linie prin elaborarea unor confirmări privind recepția corectă sau necorectă a datelor transmise anterior și scrierea lor în fișiere;
- Suspendarea schimburilor în linie prin întreruperea unui subcanal sau a întregului ansamblu de subcanale.

Se remarcă două tipuri de comenzi:

- Externe, introduse de la mașina de scris-pupitru și care utilizează un limbaj propriu de comandă;
- Interne, generate de comenzile externe și sarcinile Sistemului.

Pentru elaborarea concretă a comenzilor pe calculatorul Felix C-256 s-a plecat de la studiul teoretic realizat anterior în lucrare, unele modificări fiind necesare numai în ceea ce privește memoria care nu se poate alocă dinamic.

Concepția de ansamblu și de detaliu privind elaborarea

și funcționarea comenzilor, modul de realizare practică a acestora precum și unele probleme de mare importanță pentru funcționarea Sistemului ca: elaborarea unui limbaj de comandă care să permită o prelucrare eficientă a comenzilor și determinarea unor resurse și modalități de alocare a acestora-teoretic și practic, sînt contribuții personale cu caracter original în acest domeniu și pot constitui separat lucrări importante care pot fi generalizate și extinse și în alte domenii privind realizarea sistemelor informatice.

### LIMBAJUL DE COMANDA

Pentru Sistemul de exploatare pentru legătura UC-UC, numai comenzile au o semnificație, celelalte mesaje sînt destinate a fi transmise între cele două calculatoare (de la un operator la celălalt operator).

Astfel, pentru identificare orice comandă trebuie să înceapă obligatoriu cu caracterul "\*" iar orice mesaj destinat transmisiei între cele două calculatoare la nivel de operatori trebuie obligatoriu să înceapă cu caracterul ":".

Cele două tipuri de mesaje au în exemplul prezentat următoarea semnificație:

: MONTATI BANDA - Mesaj transmis la mașina de scris  
a calculatorului corespondent;

\*SSURSA - Comanda de stabilire subcanal.

Pe baza unei concepții proprii s-a elaborat un limbaj de comandă, sintaxa limbajului fiind următoarea:

<comandă > :: = nume comandă listă de parametri  
<listă de parametri> :: = <parametru> <listă de parametri>  
<parametru> :: = <blanc> <cuvînt cheie> <blanc> <asignator> <blanc> <argument>  
<asignator> :: = :  
<argument> :: = <număr> | <identificator>



Limbajul de comandă are la bază cuvinte cheie care permit aranjarea parametrilor într-o ordine oarecare.

. . . COMENZILE LEGĂTURII UC-UC

Comenzile Sistemului de exploatare pentru legătura UC-UC realizează stabilirea subcanalelor, schimburilor de date între cele două calculatoare și suspendarea acestor schimburi și sînt situate pe două nivele ierarhice :

- Comenzi externe care sînt lansate de la mașina de scris-pupitr și au la bază un limbaj propriu de comandă;
- Comenzi interne sau ordine care sînt generate de comenzile externe și de unele sarcini ale Sistemului.

Comenzile externe admise de către limbajul de comandă sînt următoarele:

- \* ABORTA
- \* SSURSA
- \* SRECEP
- \* INTRER
- \* INCHID

\* ABORTA este comanda de suspendare a schimburilor de date, provoacă abandonul lucrării prin folosirea procedurii ABORT a Monitorului (sfîrșit anormal de lucru și predarea comenzii sistemului dacă a survenit un incident ireparabil); nu are parametri.

\* SSURSA este comanda de Stabilire a unui subcanal SURSA de date; este emisă de calculatorul care va deservi sursa de date. și are următorii parametri:

<cuvînt cheie> :: = DEST | EXPE

<identificator>:: = <idex fișier> | <periferic>

Destinatarul - DEST este identificat printr-un cod alfanumeric al componentului așa cum este prezentat în continuare:

Cod alfanumeric	Cod numeric intern	Semnificație
CR	01	lector cartele
CP	02	perforator cartele
PR	05	imprimantă
MT	06	derulor de bandă
RD	07	disc amovibil
FD	08	disc fix
MD	0F	dimas

Expeditorul - EXPE este identificat prin idex-ul fișierului.

\* SRECEP este comanda de Stabilire a unui subcanal RECEP; este emisă de calculatorul care va deveni receptor de date și are următorii parametri:

<cuvînt cheie> :: = DEST|EXPE

<identificator> :: = <idex fișier> | <periferic>

\* INTRER este comanda care permite INTRERuperea unui subcanal și are următorul parametru:

<cuvînt cheie> :: = NRSC

<argument> :: = număr subcanal

\* INCHID este comanda de suspendare a tuturor subcanalelor; INCHIDerea Sistemului și nu are parametri.

Erorile eventuale provoacă mesaje care sînt recepționate la mașina de scris-pupitru și prin care utilizatorul este avertizat de natura lor și sînt:

- EROARE; informația introdusă de la mașina de scris - pupitru, nu este comanda sau mesaj transmis.

- COMANDA ANORMALA DE LEGATURA UC-UC; primele 9 caractere introduse la mașina de scris - pupitru nu au sens pentru Sistem.

- COMANDA ERONATA, EROARE N UC-UC; s-a detectat o eroare într-o comandă, N precizînd natura erorii.

0 - sintaxa eronată,

1 - dublă definiție a parametrului,

2 - cuvînt eronat,

3 - număr de parametru eronat,

4 - lipsă de parametri.

- COMANDA REFUZATA MOTIV N; nu s-a putut executa o comanda validă, motivele fiind următoarele:

- 0 - număr maxim subcanale depășit sau închidere cerută,
- 1 - fișier necunoscut
- 2 - fișier indisponibil
- 3 - periferic ocupat sau necunoscut
- 4 - component inexistent (corespondent)
- 5 - component indisponibil (corespondent)
- 6 - lipsă spațiu memorie
- 7 - lipsă spațiu memorie (corespondent)

- COMANDA DE INCHIDERE UC-UC; apare la mașina de scris-pupitru la recepția comenzii \* INCHID.

- INCHIDERE SUBCANAL MOTIV N; a fost închis un subcanal, motivul N are semnificație.

- 1 - închidere subcanal prin comanda \*INTRER,
- 2 - închidere subcanal la sfârșit de transmisie fișier.

Comenzile interne sau ordinele sînt elaborate și generate de comenzile externe la interpretarea (decodificarea) acestora în modulele corespunzătoare din sarcina Decodificare Mesaje mașina de scris - DMMS, sarcina RECEPTIA și interpretarea mesajelor din linie - REC și de unele sarcini utilizator.

Comenzile interne sînt de mai multe tipuri și se referă la stabilirea subcanalelor, la controlul schimburilor de date între calculatoare și la suspendarea schimburilor de date.

Comenzile interne referitoare la stabilirea subcanalelor generate de comanda externă SSURSA sînt următoarele:

- CER1, CERerea de stabilire a unui subcanal de către calculatorul care va deveni sursă de date dacă cererea va fi satisfăcută; are semnificația, stabilire subcanal și este generată de sarcina DMMS.

- ACER1, Acceptare CERere CER1 sau răspunsul pozitiv trimis de calculatorul care va deveni receptor de date; are semnificația, subcanal stabilit și este generată de sarcina REC.

- NACER1, Nu s-a Acceptat CERerea CER1 sau răspunsul negativ trimis de calculatorul corespondent care va deveni receptor de date; are semnificația, subcanal nestabil și este generat de sarcina REC.

Comenzi interne referitoare la stabilirea subcanalelor generate de comanda internă SRECEP sînt:

- CER2, CERerea de stabilire a unui subcanal de către calculatorul care va deveni receptor de date dacă cererea va fi satisfăcută; are semnificația, stabilire subcanal și este generată de sarcina DMMS.

- ACER2, Acceptare CERere sau răspunsul pozitiv trimis de calculatorul corespondent care va deveni sursă de date; are semnificația, subcanal stabilit și este generată de sarcina REC.

- NACER2, Nu s-a Acceptat CERerea CER2 sau răspunsul negativ trimis de calculatorul corespondent care va deveni sursă de date; are semnificația subcanal nestabil și este generată de sarcina REC.

Comenzile interne referitoare la schimburile de date între calculatoare, generate de sarcinile utilizator de ieșire corespunzătoare subcanalelor sînt:

- ACC, Avizare de Confirmare Corectă de recepție mesaj, care informează calculatorul sursă de date că un nou mesaj poate fi primit de către calculatorul corespondent care dispune de spațiu în memorie în urma execuției sarcinii de ieșire (scriere în fișier) a receptorului de date. Are semnificația, mesaj precedent pentru subcanalul corespunzător recepționat corect.

- NACC, NeAvizare de Confirmare Corectă de recepție mesaj, care informează calculatorul sursă de date că a existat o anomalie la recepția mesajului precedent la calculatorul corespondent. Are semnificația, condiții anormale la recepția mesajului precedent.

Comenzile interne referitoare la suspendarea schimburilor de date, generate de comenzile externe INTRER și INCHID și

de sarcini utilizator sînt:

- RPL, RuPerea Legăturii de către calculatorul sursă de date pentru subcanalul receptor în urma recepției unei comenzi interne NACC sau în cazul unui sfîrșit de fișier; are semnificația întreruperea unui subcanal și este generată de sarcina REC sau de către calculatorul sursă sau recepție de date în cazul recepționării comenzii externe INTRER; are semnificația, întreruperea unui subcanal și este generată de sarcina DMMS.

- INCHI, INCHIderea tuturor subcanalelor, este generată de comanda exterioară INCHID.

### . . . PRELUCRAREA COMENZILOR

Sistemul de exploatare pentru legătura UC-UC necesită pentru asigurarea funcționării sale, prelucrarea comenzilor externe și interne.

Comenzile externe introduse de la mașina de scris - pupitru sînt interpretate (analizate) de module din sarcina Decodificare Mesaje Mașină de Scris și ca urmare a validării sintactice, sînt generate comenzile interne corespunzătoare, de module ale sarcinilor; Decodificare Mesaje Mașina de Scris - DMMS, RECeptia și interpretarea mesaje din linie -REC, precum și cele utilizator; comenzile interne sînt interpretate de către module din cadrul sarcinii RECeptie și interpretare mesaje - REC.

Prelucrarea comenzilor implică verificarea existenței și disponibilității fișierelor, componentelor, liniilor și testarea posibilității alocării de memorie pentru crearea tamponelor component.

Existența și disponibilitatea acestor resurse este ilustrată cu ajutorul unor tabele.

Informațiile privind fișierele sînt date în Tabela Fișierelor Cunoscute - TFC cu a ceeași structură pentru calculatorul sursă de date și cel receptor de date și este prezentată în fig. 4.32.

TFC	adresa sfîrșit tabelă		
	idex	ocup	n
	adresa TDF		

ocup = {  
 o, dacă fișierul este disponibil,  
 cod, dacă fișierul este ocupat

TDF este Tabela de Descriere a Fișierului creat de Sistemul de gestiune fișiere - SGF.

Fig.4.32. Tabela Fișierelor Cunoscute - TFC

Informațiile privind componentele admise sînt descrise în Tabela de Componente Admise - TCA cu aceeași structură pentru calculatoarele sursă și receptor de date și este prezentată în fig. 4.33.

TCA	adresa sfîrșit tabela		
	cod x . n	cod	ocup
	nume sarcină intrare/ieșire		
	cuvînt șir de așteptare		

ocup = {  
 o, dacă perifericul e disponibil,  
 nr. SC, dacă perifericul este ocupat,  
 'X'30', dacă perifericul este inexistent.

Fig.4.33 Tabela de Componente Admise - TCA.

Informațiile privind subcanalele sursă de date sînt înregistrate în Tabela SubCanale Active Sursă - TSCAS a cărei compoziție este arătată în fig. 4.34.

TSCAS	adresa sfîrșit tabelă		
	nr.subcan	ocup	idex fișier
	adr.început tampon component		
	lungime alocată		
	adresa curentă în tampon		
	nume sarcini de lectură/scrie		
	adresa TDF		

} 1 intrare  
în tabelă

Fig.4.34 Tabela de SubCanale Active Sursă/Receptor  
 - TSCAS/TSCAR

$$\text{ocup} = \begin{cases} 0; & \text{dacă paragraful e liber} \\ \text{nr.subcanal}; & \text{dacă este alocat} \end{cases}$$

Tabelale TSCAS/TSCAR comportă o intrare pentru subcanalul sursă/receptor activ și sînt inițializate de stabilirea subcanalului. În tabelă, idex-ul fișierului este identificatorul de exploatare pentru fișier a cărui descriere se găsește în TDF-ul asociat, componentul de ieșire se indică prin octetul-ocup, dă informații privind tamponul component și indică numele sarcinii de umplere/golire a tamponului component (lectura sau scriere în fișiere).

Alocarea de memorie pentru tamponul component se poate realiza de un subprogram de Alocare Memorie - ALM în care se face apel la paragrafizare așa cum s-a arătat anterior în lucrare la partea teoretică. Tabelele de alocare Memorie Tampon Component Sursă sau Receptor - TMTCS/TMTCR servesc la gestiunea paragrafelor. Ele conțin un cuvînt pe paragraf, fig.4.35.

TMTCS	adresă sfîrșit tabela	
TMTCR	ocup.	adresă început paragraf

ocup = 0, dacă paragraful e liber,  
nr.subcanal; dacă este alocat.

Fig.4.35 Tabele de alocare Memorie Tampon Component Sursă sau Recepție - TMTCS/TMTCR pentru gestiunea paragrafelor.

Deci problema de alocare memorie pentru tampoanele component deosebit de importantă pentru funcționarea în bune condiții a Sistemului de exploatare pentru legătura UC-UC a fost rezolvată într-un mod eficient atît teoretic cît și practic (programul nu a fost realizat deoarece pentru acest experiment nu prezintă o importanță mare). În concluzie se poate spune că prin comenzi Sistemul de exploatare pentru legătura UC-UC își asigură resursele necesare de echipamente și de memorie.

#### 4.5. EXTINDEREA SISTEMULUI DE EXPLOATARE PENTRU LEGATURA UC-UC LA O REȚEA DE CALCULATOARE.

Lucrarea s-a referit pînă în prezent la o legătură bipunct, între două calculatoare.

Se poate realiza, după cum am arătat anterior, punerea în comun a mai multor calculatoare, printr-o interconectare a calculatoarelor, la o rețea de transmisiuni.

Sistemul de exploatare pentru o legătură UC-UC, poate fi extinsă la o rețea de calculatoare cu unele mici modificări.

Este foarte rațional să considerăm o rețea de calculatoare ca un ansamblu de legături bipunct, indiferent de modul real în care aceste legături sînt stabilite. Astfel, se poate păstra și în continuare noțiunea de subcanal pentru legăturile stabilite în cadrul unei rețele de calculatoare. Fiecare subcanal va fi definit prin parametrii care sînt, în general, legați de terminalele care sînt puse în corespondență și de calculatoarele la care aceste terminale sînt conectate.

Se pleacă de la convenția că fiecare mesaj se adresează la un singur corespondent, adică ansamblul de comenzi și de date, conținute într-un mesaj, este destinat subcanalelor unui singur calculator.

Principalele modificări care trebuie efectuate, constă în adăugarea la fiecare mesaj a unei indicații care să permită cunoașterea calculatorului care l-a emis, ca în fig.4.36.

mesaj, la fel ca pentru legătura bipunct.

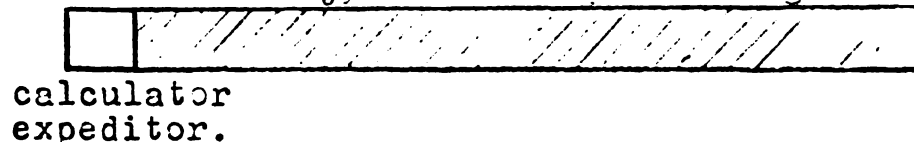
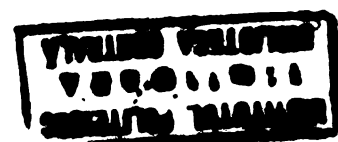


Fig.4.36. Mesajul pentru rețea; adăugarea unei indicații, la mesaj, pentru a identifica calculatorul care l-a emis.





Schimbari vor fi efectuate și la unele funcțiuni (proces), cum ar fi de exemplu, indicarea calculatoarelor corespundente cărora se adresează un mesaj introdus la consolă de către operator.

Gestiunea subcanalelor Sursă de date va putea rămîne aceeași; se va constitui un șir de așteptare a subcanalelor pregătite (pentru care un element a fost format).

La fiecare calculator, la care una sau mai multe legături sînt, sau trebuie să fie stabilite, va corespunde o zonă de comenzi de transmis.

Pentru a atribui o prioritate comunicației între operatorii diferitelor calculatoare, se va plasa subcanalul corespunzător la începutul șirului.

O altă soluție constă în crearea unui șir de subcanale pregătite de către calculatorul Receptor de date corespunzător și unui șir de așteptare circular, ca în fig. 4.37. a calculatoarelor pregătite (adică la care șirul P nu este gol).

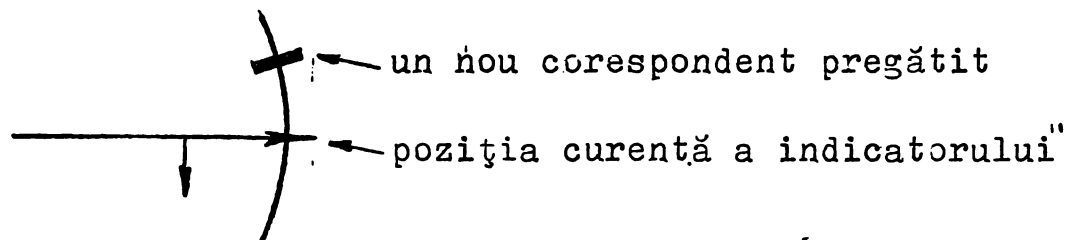


Fig. 4.37. Sirul de așteptare circular al calculatoarelor pregătite.

Sirul circular va permite alegerea legăturii bipunct (între calculatoare) care trebuie să fie luată în considerare.

Acest mod de a proceda, care introduce o ierarhie în gestiunea legăturilor, va da posibilitate de a regrupa subcanalele aceluiasi calculator și va favoriza prin aceasta, multiplexajul în spațiu.

Se recomandă, o gestiune asemănătoare pe două nivele și în cazul, cînd o prioritate va fi atașată la fiecare calculator corespondent.

#### 4.6. IMPLEMENTAREA PRODUSULUI PROGRAM CORESPUNZATOR SISTEMULUI

Produsul program corespunzător Sistemului de exploatare pentru legătura UC-UC, a fost realizat practic, conform specificațiilor prezentate în lucrare, cu funcțiunile principale prevăzute. Au fost elaborate; Sistemul de gestiune sarcini, Procedura de exploatare pentru legătura UC-UC și Comenzile sistemului, într-o structură modulară, în limbajul ASSIPLS - programul sursă fiind prezentat în anexa 1 și constituind un software de bază operațional pentru calculatorul Felix C-256 în domeniul teleprelucrării.

Produsul program este format dintr-un ansamblu de subprograme și tabele care se generează de fiecare utilizator în funcție de cerințele acestuia cu ajutorul unor parametri (număr de subcanale, lungime tampon de recepție și emisie în linie etc.).

Pentru satisfacerea cerințelor utilizatorului produsul program necesită alocarea unor resurse dintre care cea mai principală este memoria operativă compusă din:

- o parte fixă, alocată pentru subprograme ale Procedurii de exploatare - sarcinile permanente, subprograme ale Sistemului de gestiune sarcini și unor tampon de recepție și emisie;
- o parte variabilă alocată ca urmare a stabilirii numărului de subcanale (teleconversii simultane) pentru subprograme ale Procedurii de exploatare, sarcinile utilizator (lectură și scriere), tabelele Sistemului de gestiune sarcini, tabelele Procedurii de exploatare și tampon de recepție - emisie.

Produsul program reprezintă un sistem de teleprelucrare de tip alternat printr-o procedură SEND AND WAIT (transmite și așteaptă).

Implementarea produsului program s-a realizat pe un calculator electronic Felix C-256 folosind cele două partiții

ale acestuia care au fost legate între ele, extern, printr-un canal de comunicație - cablu de rebucclaj, cu o cale a cuplorului CPM afectată fiecărei partiții. Astfel, s-a simulat perfect legătura directă între două calculatoare la distanță; funcționarea produsului program în ambele cazuri este total identică, cu remarcă că nu a fost posibil ca în aceste condiții să se sesizeze influența erorilor ce apar la transmisia datelor la distanță - aspect mai puțin important pentru produsul program.

S-au putut astfel testa și implementa în condiții reale toate funcțiile produsului program precum și legătura acestuia cu Sistemul de gestiune a fișierelor - SGF și cu Sistemul de gestiune a transmisiilor - SGT în special cu protocolul de comunicație TMM UC-UC2.

Produsul program permite în cazul experimentului de teleconversia datelor; citirea unui fișier de intrare, pe cartele, la calculatorul sursă de date și scrierea datelor într-un fișier de ieșire, la imprimantă, la calculatorul receptor de date. Se asigură astfel teleprelucrarea datelor printr-un singur subcanal; teleconversia simultană a mai multor fișiere poate fi realizată prin stabilirea unui număr corespunzător de subcanale, număr restricționat în special din punct de vedere al memoriei disponibile.

Experimentarea produsului program pentru Sistemul de exploatare pentru legătura UC-UC, s-a realizat practic prin introducerea programelor aferente în cele două partiții, partiția 1 și partiția 2 ale unui calculator de tip Felix C-256.

La introducerea programelor într-o partiție, sînt activate sarcinile acestora, programele punîndu-se apoi în așteptarea sosirii unor mesaje din canal sau a unei intervenții de la mașina de scris pupitru.

În urma experimentului realizat sînt prezentate în anexe: consologul anexa 2 și vidajele partițiilor 1 și 2 ale aceluiași calculator - anexe 3 și 4.

Deci, experimentarea produsului program al Sistemului de exploatare pentru o legătura UC-UC a avut ca scop principal și validarea funcțiilor produsului program și în final realizarea unei biblioteci corespunzătoare.

Au fost efectuate experimentări în ceea ce privește:

1. - Transmisia de mesaje între mașinile de scris pupitru ale celor două calculatoare, respectiv transmisia mesajelor între cele două partiții ale unui calculator.

După intervenția pupitru pentru o partiție (în cazul experimentului prezentat, partiția 1), Sistemul de gestiunea sarcinilor lansează în execuție sarcina DMS cea mai prioritară, care cere introducerea unui mesaj de la pupitru. După recepția mesajului acesta este analizat și în caz că începe cu ":" este considerat mesaj de transmis către mașina de scris pupitru a calculatorului corespondent. În caz de validare, mesajul este format de către sarcina FM și emis în linie de sarcina SM care cuprinde și procedura TM, aceasta comutând de pe recepție pe transmitere prin calea fizică a CTM-ului afectat partiției. Semnalul transmis în linia de comunicație formată din cablul de reuclare și recepționat din linie de CTM-ul afectat partiției corespondente (în cazul experimentului partiția a 2-a) este decodificat și scris la mașina de scris pupitru.

În cazul primirii unei confirmări de recepție pozitivă de la calculatorul corespondent este lansată din nou sarcina de recepție a mesajelor din linie-RBC și lansat din nou prin procedura TM, o nouă recepție.

În cazul imposibilității emisiei din cauze hardware sau software, se vor emite mesajele corespunzătoare la pupitru și se va întrerupe sistemul.

În cazul unei neconfirmări pozitive a recepției se va transmite mesajul de mai multe ori în funcție de cele stabilite de utilizator și în caz de imposibilitate recepție corectă se va semnala la pupitru printr-un mesaj.

Trebuie menționat că în cazul mesajului de la mașina de scris pupitru, nu sînt activate sau dezactivate sarcini utilizator - citire și scriere, mesajele fiind transmise prin subcanalul permanent stabilit între pupitrele celor două calculatoare.

În vidajul partiției 1 mesajul introdus în această partiție, VERIFICATI STAREA IMPRIMANTEI, se găsește în zona Tampon mașina de scris la recepție - TMSR la adresa BAE4 iar mesajul recepționat din linie se găsește în vidajul partiției 2 în zona Tampon mașina de scris la emisie - TMSB la adresa 24BE5. Același lucru se poate verifica și în consolog.

2.- Transmiterea comenzilor operator presupune stabilirea sau închiderea unui subcanal sau a sistemului, de confirmare recepție etc. Se realizează un dialog între cele două calculatoare prin intermediul comenzilor interne generate și interpretate de sarcinile de Decodificare a mesajelor de la mașina de scris - DMS și de Recepție a mesajelor din linie - REC. Pentru transmisia de date este necesară în prealabil stabilirea unui subcanal; aceasta se poate realiza prin introducerea de la mașina de scris a comenzilor \* SSURSA sau \* SRECP.

În cazul experimentului a fost realizată funcțional doar comandă externă \* SSURSA, prin care se cere teletransmisia datelor dintr-un fișier de la calculatorul care face cererea de emisie și va deveni în cazul acceptării comenzii sursă de date, către cel corespondent care va deveni în aceleași condiții receptor de date.

Procesul în detaliu este următorul; în urma introducerii comenzii \* SSURSA în cazul îndeplinirii condițiilor de transmis de la calculatorul sursă de date, se va stabili un subcanal și se va emite către calculatorul corespondent receptor de date comanda internă CER1, iar în caz contrar, va fi emis un mesaj de eroare la pupitru.

În cazul validării condițiilor la calculatorul sursă de date și a emisieii comenzii CER1 aceasta va fi analizată la calculatorul receptor de date și vor fi emise de către acest calculator, în funcție de acceptarea sau nu a comenzii

CCRL, comenzile interne ACCRL sau AACRL.

În cazul emisie de la calculatorul ce ar fi trebuit să devină receptor de date, a comenzii AACRL și în urma primirii acesteia de către calculatorul ce solicită să devină sursă de date, va fi emis la pupitrul acestuia din urmă un mesaj de eroare corespunzător subcanalului selectat anterior.

În cazul validării condițiilor impuse de comanda CCRL, la calculatorul receptor de date este activată sarcina de scriere corespunzătoare, apoi se emite comanda internă ACCRL.

La primirea comenzii ACCRL de către calculatorul ce va deveni sursă de date se activează și se lansează sarcina de lectură corespunzătoare și se trece la citirea datelor din fișier.

Astfel în cazul experimentului comanda de stabilire a subcanalului a fost introdusă în partiția 2, în Ramponul mașina de scris recepție - MRSR după cum se poate vedea din consolog și din vidajul partiției 2, la adresa 24A<sub>15</sub>.

În timpul sesiunii de lucru, anterior introducerii unei comenzi valide, au fost introduse în mod intenționat comenzi ce prezentau erori de sintaxă sau semantică.

Astfel pe consolog se pot distinge atât comenzile eronate introduse cât și mesajele de eroare emise de analizorul limbajului de comandă. În vidaj, comenzile eronate introduse nu se pot observa deoarece mesajele și comenzile operator care se introduc în zona MRSR se suprapun peste mesajul anterior.

În cazul nostru a fost păstrat doar ultimul mesaj care era valid.

Comenzile de confirmare pozitivă sau negativă a recepției ACC și AAC pentru un subcanal emise de calculatorul receptor de date și recepționate de calculatorul sursă de date, permit la acesta din urmă punerea în starea activat pregătit de lansare, din starea activat în așteptare sau nu, a sarcinilor de citire, respectiv continuarea transmisiei de date pentru subcanalul respectiv.

În Tamponul de Emisie Linie TABEL al calculatorului receptor de date (vidaj partiția 1) la adresa ADDO se poate observa comanda ACC pentru ultimele date recepționate C3o7ol unde:

- C3 - indicator comandă
- o7 - cod comandă ACC
- ol - numărul subcanalului stabilit

Totodată la primirea comenzii ACC la calculatorul receptor, se testează și sfârșitul fișierului de transmis corespunzător subcanalului selectat. În caz că este sfârșit de fișier, se dezactivează sarcina de citire, se generează și se emite comanda RPL de închidere a subcanalului către calculatorul corespunzător.

În Tamponul de Emisie Linie, TABEL a calculatorului ce a fost sursă de date (în simulare în vidajul partiției 2) se poate observa comanda RPL la adresa 23DDO cu reprezentarea în hexazecimal C3o9olF2 unde:

- C3 - indicator comandă
- o9 - codul comenzii RPL
- ol - numărul subcanalului ce va face obiectul închiderii
- F2 - motiv închidere subcanal sfârșit de fișier.

La recepția acestei comenzi de către calculatorul receptor de date pentru subcanalul respectiv, se dezactivează sarcina corespunzătoare de scriere în fișier și se va emite la pupitrul acestuia mesajul "SUBCANAL INCHIS; MOTIV 2".

În vidajul partiției 1 în care s-a simulat calculatorul receptor de date, în Tamponul de Recepție Linie TARL se poate observa primirea comenzii RPL la adresa B1DO.

La consolog la sfârșitul transmisiei pentru partiția receptoare de date (partiția 1) se poate observa mesajul emis "SUBCANAL INCHIS; MOTIV 2".

3 - Transmisia de date se realizează prin citirea datelor din fișier până la umplerea tamponului component, formarea mesajului de transmis în linie de către sarcina FML și transmiterea în linie de către sarcina LML, după care calculatorul sursă de date se va pune în stare de așteptare lansând sarcina REC.

Calculatorul receptor de date analizează datele transmise și lansează sarcina de scriere corespunzătoare subcanalului.

În cazul scrierii mesajului recepționat în fișierul indicat, se va emite comanda de confirmare pozitivă ACC, iar în caz contrar comanda de confirmare negativă NACC.

Primirea comenzii ACC pentru un subcanal la calculatorul sursă de date permite lansarea sarcinii de lectură corespunzătoare și reluarea ciclului de transmitere a datelor pînă la sfîrșit de fișier.

În cazul experimentului s-a efectuat transmiterea datelor citite de programul din partiția 2 de la cititorul afectat cu adresa 4 și scrierea lor de programul din partiția 1 la imprimantă cu adresa 2.

În vidajul partiției 2 se pot vedea în Tamponul component al sarcinii de citire la adresa 24CE5, ultimele două înregistrări cartelă albă EOF, precum și cea anterioară care conține; S-A TERMINAT TRANSMISIA DE DATE. În tamponul component al sarcinii de citire se citesc cele 3 cartele.

Datele transmise se pot vedea în anexa 5.



## 5. CONCLUZII

Prezenta lucrare reprezintă activitatea proprie de cercetare științifică și de valorificare practică a acesteia, cu scopul principal atât de a realiza teoretic și practic, un instrument informatic unitar, modern, performant și generalizabil, "Sistemul de exploatare pentru legătura UC-UC" cu extindere la o rețea de calculatoare cât și de a elabora, cu această ocazie, pentru diferitele etape de realizare ale sistemului, metode generale teoretice și practice, de calcul și de construire a sistemelor de teleprelucrare și rețelelor de calculatoare.

Lucrarea are un caracter original, fiind realizată pe baza ideilor și conceptelor proprii degajate din cercetările teoretice și aplicative desfășurate de autor pe parcursul anilor; de asemenea lucrarea ia în considerare cele mai valoroase realizări teoretice și practice din informatică, matematică și tehnică de unde desprinde unele teorii și tehnici care prin modificări, adaptări și dezvoltări la condițiile noastre specifice au putut astfel constitui soluții proprii, concrete pentru problemele cerute de elaborarea sistemului.

Realizarea sistemului nostru a fost necesară și oportună avînd în vedere, atât cerințele actuale de dezvoltare a economiei naționale privind introducerea unor astfel de sisteme cât și datorită faptului că pînă în prezent nu există o teorie unitară, precum și metode generale teoretice și practice de calcul și de construire. Astfel că, în acest domeniu încă insuficient cercetat, lucrarea noastră încearcă să aducă un aport teoretic și practic prin elaborarea și valorificarea cercetărilor și dezvoltărilor proprii, oferind soluții eficiente teoretice și practice de realizare a sistemelor complexe de teleprelucrare, cu caracteristici superioare (generalizabil, modular, extensibil, siguranță în funcționare, timp mic de răspuns, etc.). De asemenea lucrarea prezintă implicit, pentru etapele de realizare a sistemului, un ansamblu de metode teoretice și practice care pot fi utilizate la elaborarea sistemelor de teleprelucrare a datelor și rețelelor de calculatoare.

Lucrarea noastră, atât în ansamblu, cât și pe părți componente, prezintă contribuțiile personale, teoretice și practice, în domeniul teleprelucrării datelor și rețelelor de calculatoare, punând la dispoziție o realizare integrală a unui sistem de referință, cu soluțiile teoretice valorificate practic și oferă metode generale de calcul și de construire.

În legătură cu contribuțiile aduse, pot arăta că am acordat un interes deosebit problemelor teoretice, urmărind crearea pe baza unei teorii proprii unitare a unor metode bine fundamentate științific și cu un caracter cât mai general privind arhitectura și comportamentul sistemului.

În ceea ce privește arhitectura sistemului am procedat la definirea optimă a elementelor de bază ale acestuia în vederea obținerii unor performanțe superioare; grad mare de generalizare, modularitate, siguranță în funcționare, timp mic de răspuns și altele. În acest sens, am elaborat lucrări proprii care au un caracter metodologic. Astfel, am definit modelele și algoritmi ce reprezintă funcțiile de bază, atât pentru Procedura de exploatare - procesele paralele, cât și pentru Procedura de comandă - comenzile sistemului și am rezolvat problemele teoretice ale Procedurii de gestiunea proceselor - sincronizarea proceselor paralele cu ajutorul semafoarelor în două medii diferite, elaborând în acest sens algoritmi necesari cu ajutorul unui limbaj adecvat, precum și alocarea resurselor de memorie pentru tampoanele de teleprelucrare, stabilind algoritmi necesari ce au fost rezolvați matematic. De asemenea am conceput și definit: organizarea generală a sistemului în special a memoriei pentru tampoanele de teleprelucrare, configurația, locul și poziția sistemului, structura sa ierarhizată, compoziția optimă a mesajului de transmis și altele.

În ceea ce privește comportamentul sistemului, am procedat la elaborarea parametrilor de bază, în special intensitatea maximă de trafic, pentru a asigura sistemului performanțe superioare și dimensionare optimă (capacitate de memorare, de prelucrare, a liniilor, etc.). Pentru aceasta am definit modelele stochastice care reprezintă sistemul respectiv, caracteristicile acestuia (capacitate limitată, protocoale de transmitere a mesajelor, etc.) și alegând pentru rezolvarea acestora, cele mai adecvate tehnici matematice, în special de teoria așteptării și de

probabilități; am creat totodată comparativ, în funcție de simplificarea mai mare sau mai mică a realității, metode proprii de calculul parametrilor sistemului, metoda analitică-exactă și metodele prin aproximare - echivalență și difuzie.

Am acordat totodată un interes deosebit problemelor practice în vederea realizării unui software de teleprelucrare concretizat într-un produs program cu performanțe superioare și cu largi posibilități de generalizare pentru calculatorul Felix C-256. Pentru aceasta am elaborat o serie de lucrări proprii care pot constitui metode de construire practică a sistemelor și care se referă la: concepția aplicativă și proiectarea tehnică (specificatiile de definire și realizare), construirea programelor (parametrizat și modular), în limbajul ASSIRIS, implementarea produsului program pe calculatorul Felix precum și experimentarea acestuia prin testarea reală a tuturor funcțiilor sale.

Pentru realizarea în cele mai bune condiții a produsului program am elaborat o serie de lucrări proprii necesare ca: algoritmi - transpuși în programe, pentru sarcinile Procedurii de exploatare, pentru funcțiile Sistemului de gestiunea sarcinilor și pentru comenzile sistemului, protocoale pentru stabilirea legăturii și asigurarea schimburilor între calculatoare, limbaj de comandă și altele. De asemenea, am recurs pentru rezolvarea unor probleme concrete (sincronizarea sarcinilor, punerea de acord a calculatoarelor, protocoale de transmisie etc.) la utilizarea facilităților calculatorului Felix (simultaneitate, prelucrări cu intrări/ieșiri, așteptare multiplă, multiprogramare, legătura directă între memorii, procedura de transmitere mod mesaj și altele) și am elaborat unele soluții proprii pentru eliminarea restricțiilor calculatorului în special în ceea ce privește gestiunea fișierelor și gestiunea transmisiilor.

Lucrarea prezentată este completă și unitară, corespunde problemelor actuale privind realizarea sistemelor de teleprelucrare și a rețelelor de teleprelucrare și indică totodată noi direcții de dezvoltare în acest domeniu.

B I B L I O G R A F I E

1. Abranson N, Kuo F. Computer Communication Network. Prentice Hall 1973 (1) (2).
2. Adiri I, Avi Itzhak B. A Time-Sharing Queue with a Finite Number of Customers. J. ACM 16. 1969 (3).
3. Adiri I. Queueing Models for Multiprogrammed Computers. Proc. Int. Symp. Computer Network Brooklin 1972 (3).
4. Anderson G. Design of a Time-Sharing Allowing Interactive System. Proc. ACM 1968 (2).
5. Aronson R. Data Link and Networks Device and Tehniques. Control Eng. feb. 1970 (1).
6. Arzac J. Les Systèmes de conduit des ordinateurs. Dunod 1973 (1).
7. Avi - Itzahak B. A Sequence of Service Station with arbitrary imput and Regular Service Times. Manag. Sci. 1965 (3).
8. Avi - Itzhak B., Heyman D. Approximate Queueing Models for Multiprogramming Computer Systems. nr.6 1973 (3).
9. Avi - Itzhak B, Madin M. A Sequence of two Servers with no Intermediate Queue. Manag. Sc. 11. 1965 (3).
10. Baltac V. Optimizarea sistemelor de operare ale calculatoarelor numerice Ed. Facla 1974 (1) (2).
11. Baltac V. ș.a. Calculatorul Felix C-256; structură și programare Ed. Tehnică 1974 (1) (4<sup>\*</sup>).
12. Baltac V. Probleme de optimizare a exploatării unui calculator numeric de capacitate medie. Rev. Automatica și Electronica nr.4 1973 (1) (2).
13. Bachman C, Canapa M. The Session Control Layer of on Open System Interconnection. Proc. Computer Net. Sept. 1978 (2).
14. Baskett F, Chandy R, Muntz R, Palacios F. Open, Closed, and Mixed Networks of Queues with Different Class of Customers J. ACM 22 1975 (3).
15. Baskett F, Muntz R. Network of queues. Proc. Int. Conf. Princeton 1973 (3).

**Notă.** Asterixul are semnificația că lucrarea a prezentat importanță și a fost consultată cu multă atenție.

16. Saucilhon F. Principes de Conception automatique dans le système SPECTRE. RAIRO oct. 1973 (1) (2).
17. Barber D. The European Computer Network Project. Proc. ICCO. Toronto 1976 (1<sup>x</sup>) (2).
18. Bartlett K. Transmission Control in a Local Data Network. Information Processing. Holland Pub. 1969 (1).
19. Belady L. A study of replacement algoritms for virtual storage Computer. IBM Sys. Jour. vol.5, 1976 (2).
20. Belady L, Kuehner C. Dynamic Space Sharing in Computer Systems. ACM vol.12. nr.5 1969 (3).
21. Betourné C, Boulenger J, Ferrié J, Kaiser C, Krakoviak S, Mossière J. Process Management and Resource Sharing in the Multiaccess System ESOPE. ACM vol.13 nr. 12. dec. 1970 (2) (3<sup>x</sup>) (4).
22. Betourné C, Boulenger J, Ferrié J, Kaiser C, Krakoviak S, Mossière J. Notion D'Espace Virtual dans le Système ESOPE. Congrès d'informatique Paris sept. 1972 (2)(3).
23. Betourné C, Boulenger J, Ferrié J, Kaiser C, Krakoviak S, Mossière J. System design and Implementation using paralel proces. IFIP Cong. Lyubliana 1971 (2)(3<sup>x</sup>).
24. Beteurné C, Boulenger J, Ferrié J, Kaiser C, Krakoviak S, Mossière J. Presentation Generale du Système ESOPE; allocation de ressources dans le System ESOPE. Congrès AFCET 1972 (2<sup>x</sup>) (3<sup>x</sup>).
25. Brandwayn A. Simulation de la charge d'un système conversationnel, RAIRO vol 10. nr.5 1976 (3<sup>x</sup>).
26. Brandwayn A. Procedure pour une liaison entre des ordinateurs. Bul. Université de Paris 1972 (2) (4<sup>x</sup>).
27. Brandwayn A, Gelenbe E, Lenfant I, Potier D. A model of Program and System Behaviour in virtual Memory. Rap. Laboria IRIA 1973 (3).
28. Buzen J. Structural Consideration for Computer Models. Proc. Conf. Systems. Princeton 1974 (3).
29. Bubenov I, ș.a. Sistem automatizat de rezervare a locurilor la transporturile aeriene, Mechanizația i Automațizatiã Uprav. nr.6 1968 (1).
30. Byrns D. Consideration in Designing a Computer Communication System. Dataation nr.10 1969 (1).

31. Carr S. Host - Host Communication Protocol in the ARPA Network. Proc. SICC 1970 (2).
32. Caseau P. Pujolle G. Application de modèles markoviens aux conditions de stabilité dans des systèmes informatiques. IRIA Research Report 1977 (3<sup>\*</sup>).
33. Caseau P. Note concernant le taux de charge maximum d'une suite de station. RAIRO vol.10. nr.5 1976 (3<sup>\*</sup>).
34. Cantor D, Gerla M. Capacity Allocation in Distributed Computer Network. Proc. Conf. System Hawaii 1974 (2).
35. Cashin P. DATAPAC Network Protocols. Proc. ICCO Toronto 1976 (2).
36. CCITT. Recommendation X3, X25, X28, X29, Packet assembly/dis-assembly facility in Public Network; Interface between Data Terminal Equipment and Data Circuit 1977 (2) (4).
37. Cerf V, Kahn R. A Protocol for Packet Network Communication. IEEE Trans. on Com. Vol. 22 1974 (2) (3).
38. Cerf V. Proposal for an Internetwork End - to - End Transport Protocol. IFIP Note 96 Jan. 1978 (2) (3).
39. Chandy M, Russell A. The Design of Multipoint Linkages in a Teleprocessing Tree Network. IEEE Trans. Com. 5 1972.
40. Chandy M. The Analysis and Solution for General Queueing Network. Proc. Systems Princeton 1972 (3).
41. Chou W, Konheim A. On the Analysis and Modeling of a Class of Computer Communication System. IEEE Trans. on Comunic. 1972 (3).
42. Chang W. Queues with feedback for Time-Sharing Computer System Analysis. Operation Research nr.3 1968 (3).
43. Chretien G, Konig W, Rech J. The SIPA Network. Computer Com. Network 1975 (2).
44. CII. Système de gestion de transmission (SGT), Système de gestion de fichiers (SGF), Système de exploitation, Monitor SIRIS 2/3, Stratège, Procédure TMM UC-UC Introduction au télétraitement, Coupleur CTQM et CTC, ASSIRIS etc. (4<sup>\*</sup>).
45. CII. Réseau Coducée. Tribune du Cercle des Utilisateurs de la CII. oct. 1971 (1).

46. Coffman E, Kleinrock L. Feedback Queueing Models for Time-Shared System. Journal ACM 15 1968 (3).
47. Coffman E, Ryan P. A Study of storage partitioning using a mathematical model of locality. Com ACM 1972 (3).
48. Coffman E, Ryan B, Ryan T. Analysis of a technique for loading and executing a proces in paralell. RAIRO nr.2 1979 (3).
49. Courtois P. On the Near Complete Descomposability of Queues and Stochastic Models of Multiprogramming Systems Raport Research Mellon University 1977 (3).
50. Corbato F, Vyssatsky A. Structure of the MULTICS Supervisor Proc. FSCC 1969 (2) (3).
51. Constantinescu P., Negoită C. Sistemele informatice, modele ale conducerii și sistemelor conduse. Ed. Tehnică 1974 (1).
52. Dang N, Quint V, Seguin J, Sergeant G. Un Sous-système normalisé de commutation de processus pour la téléinformatique et les réseaux d'ordinateurs. Rapport de recherche. Institut Polytechnique de Grenoble 1976 (2<sup>x</sup>) (3).
53. Dang N, Fauvrier R, Quint V. SYNCOP. Rapport ENSIMAG 1976 (2<sup>x</sup>)
54. Davies W. The Control of Congestion in Packet Switching Networks. Proc. ACM/IEEE Conf. Palo Alto 1971 (3).
55. Davies W. The principles of a data communication networks for computers and remote peripherals. Congress IFIP 1968 (1) (2).
56. Davies W. Communication Networks to serve Rapid Response Computers. Cong. IFIP 1972 (1).
57. Davies W, Barber D. Communication Network for Computer. Wiley New York 1973 (1) (2<sup>x</sup>).
58. Davies W. Euronet project. Int. Conf. Computer Toronto 1976(1)
59. Dauthine A, Bremer J. An axiomatic discription of the transport protocol of Cyclades. Conf. Comp. Net. Aachen 1976 (1) (2).
60. Davidson J, Hathaway N, Pastel J. The ARPANET. TELNET Protocol ACM.IEEE Computer Symp. 1977 (1) (2).

61. Danet A, Depres R, Les Rest A, Pichon G, Ritzenthaler S.  
The French Public Packet Switching Service; The  
TRANSPAC Network. Proc. ICCO Toronto 1976 (1) (2).
62. Day J. Resource Sharing Protocols. IEEE Computer nr.9 1979(2)
63. Denning P, Schwartz G. Properties of the working set model.  
ACM vol 15.nr.3 1972 (3).
64. Denning P. Thrashing: Its causes and prevention. Proc. F.JCC  
1968 (2) (3).
65. Denning P. Resource Allocation in a Multiprocess Computer  
System MIT 1968 (2) (3).
66. Denning P. The working set model for program behaviour. Com.  
ACM 8.1968 (2) (3).
67. Denning P. Third Generation Computer System. Princeton  
University New Jersey 1974 (2) (3).
68. Dijkstra W. Cooperating Sequential Process. Genuys Academic  
Press 1967 (2\*) (3\*).
69. Dijkstra W. The structure of THE Multiprogramming System:  
Comm. ACM vol.11 nr.5 1968 (2) (3).
70. Drăgănescu M. A douarevoluție industrială. Microelectronica,  
automatica, informatica - factori determinanți.  
Ed. Tehnică 1980 (1).
71. Drăgănescu M. Realizarea unui sistem informatic național  
Viața Economică 1970 (1).
72. Drăgănescu M. Sistem și civilizație. Ed. Politică 1976 (1).
73. Elie M. General Purpose Network of Computers UCLA 1970 (1).
74. Elie M. Le réseau d'ordinateurs de l'ARPA et les réseaux  
generaux d'ordinateurs. RAIRO nr.2. 1976 (1\*).
75. Eyries F, Pujolle G. Validation and prediction of performan-  
ce in the Cigale network. Proc. ICPCI Italy 1978 (3).
76. Fayolle G, Gelenbe E, Pujolle G. An Analytic Evaluation of  
the Performance of the Send and Wait Protocol. IEEE  
Trans. on Comm. nr.3 1978 (3\*).
77. Fayolle G, Gelenbe E, Labetoulle J, Bastin D. The Stability  
Problem of Broadcast Packet Switching Computer Net-  
works. Comp. Arch. and Net. Amsterdam 1974 (3).
78. Ferrié J, Mossiére J. ESOPÉ; Gestion de processus et partage  
des ressources, IRIA Cahier 4 1972 (2\*) (3).



79. Feinler E, Posel B. ARPANET Protocol Hand book 1978 (2).
80. Frank H, Chou W. Topological Optimisation of Computer Network. Proc. IEEE nr.11 1972 (1).
81. Frank H, Frisch I, Chou W. Topological Consideration in the Design of the ARPA Network. Proc. Conf. AFIPS 1970 (1) (2) (4).
82. Frank H, Kohn R, Kleinrock L. Computer Communications Network Design Proc. Conf. AFIPS 1972 (1) (2).
83. Frata L, Gerla M. The Synthesis of Computer Network: Proprieties of the Optimum Solution. ACM. Intern. Comp. Symp. Venice 1972 (1) (4).
84. Friedeman H. Reduction Methods for Tandem Queueing Systems. Operation Research nr. 13 1965 (3).
85. Gaver D. Diffusion Approximations and Models for Certain Congestion Problems. Journal of App Probl. nr.5 1967 (3<sup>\*</sup>).
86. Gaver D, Lehoczki J. A Diffusion Approximation for a Communication System Allowing Messages Interference. IEEE Trans. Com.nr.8 1979 (3).
87. Gaver D, Shedler G. Approximate Models for Processor utilisation in Multiprogrammed Computer Systems. Journal of Computing nr.2 1973 (3).
88. Gaver D, Shedler G. Processor Utilisation in Multiprogramming System Via Diffusion Aproximation. Operations Research nr. 21 1973 (3).
89. Gelenbe E. On Approximate Computer System Models. Journal ACM nr.7 1975 (3<sup>\*</sup>).
90. Gelenbe E. A unified Approach to the Evaluation of a class of Replacement Algoritms. IEEE nr.6 1973 (2) (3).
91. Gelenbe E. Modèles à diffusion de systèmes à ressources partajables; application à l'étude d'un système à mémoire virtuelle Rep. Laboria IRIA 1975 (3<sup>\*</sup>).
92. Gelenbe E. The two-thirds rule for dynamics storage allocation under equilibrium. Inform. Proc. Let. 1975 (3<sup>\*</sup>).
93. Gelenbe E, Boekhorst S, Kessels J. Minimising wasted space in partitions segmentation. Com. ACM nr.9 1975 (2)(3).
94. Gelenbe E, Brandwajn A, Potier D. Gestion optimale d'un système d'ordinateur multiprogrammé à memoire virtuelle. 5th IFIP Congres june 1973 (2<sup>\*</sup>) (3<sup>\*</sup>).

95. Gelenbe E, Pujolle G. Approximation to a single queue in a network. Acta Informatica 7 1976 (3).
96. Gelenbe E, Muntz R. Probabilistic Model of Computer Systems. Acta Informatica 9 1978 (3).
97. Gelenbe E, Pujolle G. Probabilistic Models of Computer System C.ACM 7 1976 (3).
98. Gelenbe E, Labetoulle J, Pujolle G. Performance evaluation of the HDLC. Comp. Network feb. 1978 (2) (3).
99. Gelenbe E, Grange J, Mussard P. Performance Limits of the TMM Protocol: Modeling and Measurement. IRIA Research Report 1977 (3).
100. Gerla M. Approximation and Bounds for the Topological Design of Distributed Computer Network. Proc. Data Com. Symp. Quebec 1975 (1) (2).
101. Gerla M. The Design of Store-and - Forward Network for Computer Communications. Rap. UCLA Los Angeles 1973(3).
102. Gordon j, Newell G. Closed queueing systems with exponential servers. Operations Research nr. 15 1967 (3).
103. Grange L, Pouzin L. Cigale, la machine de Commutation de packets de réseau Cyclades. Rap. IRIA 1973 (2) (4).
104. Grange L, Moussard P. Performance measurement of line control protocols in the Cigale network Proc. Computer Net. Liège 1978 (3) (4).
105. Grandidier M. Présentation d'un système de transmission des informations numériques permettant d'effectuer l'entr des travaux à distance. Collogue Intern. pour la Téléinformatique. Chiron Paris 1969 (1) (2).
106. Guran M. ș.a. Considerații privind cerințele de elaborare a unui model de rețea de calculatoare cu resurse distribuite geografic. Com. Ses. Academiei 1980 (1).
107. Guran M. ș.a. Protocolul de comunicație între procese pentru o rețea de calculatoare eterogene cu resurse distribuite geografic. Simp. Cluj Napoca 1980 (1).
108. Guran M.ș.a. Rețele de comutare a pachetelor: analiză și criterii de dimensionare Ed. Teh. AMC 28 1979 (1).
109. Guran M. ș.a. Comutarea de pachete în rețelele publice de transmisie a datelor - Simpozion Timișoara 1980 (1).

110. Guran M, Filip F.ş.a. Computer aided design of data transmission network for a large real time computer system. Ed. Academiei R.S.R. nr.10 (1) (4).
111. Hansen S. Concurrent programming concepts. ACM Computing Surveys 1975 (2) (3).
112. Habermaun N. Synchronisation of Communicating proces. C.ACM nr.15 1972 (2) (3).
113. Heart R. The Interface Message Processor for the ARPA Computer Network. Proc. SJCC 1970 (2).
114. Heyman D. An Approximation for the busy Period of the M/G/1 Queue using a Diffusion Model. Appl. Prob. nr.11 1974(3).
115. Hildebrand D. On the Capacity of Tandem server Finite Queue Service Systems. Operations Research 1968 (3<sup>\*</sup>).
116. Hildebrand D. Stability of Pandem Server Finite Queue Systems. Journal Appl. Probab. nr.7 1967 (3).
117. IBM Corp. IBM System/370, IBM System/360 (1) (4).
118. Inose H, Saita T. Theoretical Aspect in the Analysis and Syntesis of Packet Communication Networks. Proc. IEEE 1978 (1) (2) (3).
119. Irland M. Queueing analysis of a buffer allocation schema for a packet switch. Proc. IEEE Telecom nov.1975 (3).
120. Jacobsen T, Hogh P, Hansen T. Virtual Terminal Protocol Transport Service and Session Control. CACM nr. 1 1980 (1) (2).
121. Jardins R, White G. ANSI Reference Model for Distributed Systems. Proc. Comp. Com. Net. 1978 (2) (3).
122. Kahn R, Crowther W. Flow Control in a Resource Sharing Computer Network. Proc. ACM Palo Alto 1971 (3).
123. Karp M. Origin development and current status of the ARPANET. Proc. COMP.COM. 1973 (2) (4).
124. Karplus W. Sistem de calculatoare cu divizarea timpului ed. Tehnică 1970 (1).
125. Kaufman I. Sistem de operare. curs Inst. Politehnic Traian Vuia Timișoara 1975.
126. Kingman J. Markov Population Processes. Journal of Appl. Probab. nr.5 1969 (2) (3).
127. Kleinrock L. Analytic and Simulation Methods in Computer Network Design Proc. 1970 (3).
128. Kleinrock L. Queueing Systems: theory and applications,

- Wiley Interscience 1976 (1) (2\*) (3\*).
129. Kleinrock L. Performance models and measurement of the ARPA Computer networks. Proc. Int. Symp. University 1972 (3).
  130. Kleinrock L. Communications Nets; Stochastic Messages Flow and Delay. MC Graw Hill New York 1972 (1) (3\*).
  131. Kleinrock L. Performance of distributed multi-access computer communication system. Proc. IFIP Congress 1977 (3).
  132. Kleinrock L. Resource allocation in computer system and computer communication networks Proc. IFIP Congress Stockholm 1974 (3).
  133. Kleinrock L. On Flow control. Proc. IEEE Toronto 1978 (3).
  134. Kleinrock L. Models for Computer Networks. Proc. IEEE Colorado 1969 (3).
  135. Kleinrock L. Time - Sharing Systems; Analytical Methods. Prentice Hall 1969 (2) (3).
  136. Kleinrock L. Computer Networks-Wiley Interscience New York 1972 (1) (2) (3).
  137. Kleinrock L. Scheduling and Delay in Time -Shared Systems and Computer Network. Computer Communication Network. Prentice Hall 1973 (3).
  138. Kleinrock L, Opderbeck H. Throughput in the ARPANET Protocols and Measurement. Proc. Data Communication Symp. Quebec 1975 (3) (4).
  139. Kleinrock L, Naylor W, Opderbeck H. A study of line overhead in the ARPANET. Com. ACM nr.1 1976 (3) (4).
  140. Kleinrock L, Muntz R. Multilevel Processor Sharing Queueing Models for Time-Shared Models. Proc. 6 th Intern. Teletraffic Congress 1970 (3).
  141. Kobayashi H. Application of the Diffusion Approximation to Queueing Networks. Part. I. and II J. ACM nr.2 și 3 1974 (3\*).
  142. Kobayashi H. Bounds for the Waiting Time in Queueing Systems. Computer Arch. Amsterdam 1974 (3).
  143. Kosten L. Stochastic Theory of Service Systems. Pergamon Press 1975 (3).
  144. Konheim A, Reiser M. A Queueing Model with Finite waiting Room and Blocking J. ACM nr.2 1976 (3\*).
  145. Kuhn P. Approximate Analysis of General Queueing Networks by Decomposition. IEEE Trans. Com. nr.1 1979 (3\*).

146. Lam S. Store - and-Forward Buffer Requirments in Packet Switching Network. IEEE Trans. Com. nr.4 1976 (3).
147. Lam S, Reiser M. Congestion Control of Store and Forward Network by Input Buffer Limits: An Analysis. IEEE Trans. Com. nr.1 1979 (3).
148. Labetoulle J, Pujolle G. A Study of queueing networks with deterministic service, and applications to computer network. Acta Informatica 1976 (1) (3).
149. Labetoulle J, Pujolle G. Modeling of packet switching communication network with finite buffer size at each node. Proc. IFIP 1977 (3).
150. Lavenberg S. Stability and maximum departure rate of certain open queueing networks having finite capacity constraints. IBM Jouran Report 1975 (3).
151. Labrousse J, Girard M. Le réseau d'échange des données Proc. ACM IRIA 1972 (1).
152. Lelam G, Le Goff H. Advances in performance evaluation of communication protocols. Proc. ICCO Toronto 1976 (3).
153. Lelam G. Distributed System-Toward a Formal Approach. Proc. IFIP Cong. 1977 (1).
154. Martin J. Design of real - time Computer Systemy. Prentice Hall 1967 (1) (2) (4).
155. Martin J. Utilisation et programmation des ordinateurs en temp real Ed. organis. Paris 1969 (1) (2) (4).
156. Martin J. Telecommunication and the Computer. Prentice Hall 1969 (1).
157. Martin J. Teleprocessing Network Organisation. Prentice Hall 1970 (1) (2) (4).
158. Martin J. Systems Analysis for Data Transmission. Prentice Hall 1972 (1) (4<sup>x</sup>).
159. Merlin P. Specification and Validation of Protocols. IEEE Trans. Com. nr.11 1979 (1) (2).
160. Metcalfe R. Strategies for Operating Systems in Computer Network. Proc. ACM 1972 (1) (4).
161. Metcalfe R, Boggs D. ETHERNET; Distributed packet switching for local Computer network. C. ACM vol 19 1976 (2)(3).
162. Mc Quillan J, Walden D. The ARPA network design decisions. Comp. Network nr.5 1977 (1).

163. McQuillan J, Crowter W, Cassell P. Improvement in the Design and Performance of ARPA Network. Proc. AFIPS 1972 (1) (2).
164. Mc Kenzie A, Cassell P, Mc Quillan J, Thorpe M. The Network Control Center for ARPA Network. Proc. ICCO 1972 (1).
165. Mitrany I, Avi-Itzhak B. A many Server Queue with Service Interruption. Operat. Research nr.3 1968 (3).
166. Moore F. Computational Model of a Closed Queueing Network with Exponential Servers. IBM J. Res. Dev.1972 (3).
167. Morison J. Two Disret-Time Queues in Tandem. IEEE Trans. Com. vol. 27 1979 (3).
168. Muntz R. Analytic Model for Computer System Performance Analysis. Proc. Conf. Comp. Arch. 1974 (3).
169. Muntz R. Analytic Modeling of Interactive Systems, Proc. IEEE Interact.Comp. 1975 (3).
170. Neuts F. Two Queues in Series with a Finite Intermediate Waitroom. Journal App. Prob. nr.5 1969 (3<sup>rd</sup>).
171. Niger J, Pouzin L. Système de gestion des tâches. Congrès AFCET 1975 (3).
172. Opderbeck H, Kleinrock L. Influence of Control Procedures on the Performance of Packet Switched Networks. Proc. Telegom. Conf. San Diego 1974 (2) (3).
173. Petrescu M. Analiza și sinteza dispozitivelor numerice; Tehnica impulsurilor în circuite de comutație. Curs Inst. Politehnic București (1).
174. Petrescu M, Petrescu A. Minicalculatoare; Tehnologii; structuri, software. Ses. Com. Academia R.S.R. 1974 (1).
175. Petrescu A. Microprogramarea. Principii și aplicații. Ed. Tehnică 1974 (1).
176. Pescaru V, Dumitrescu I, Silciu C, Satran I, Nica L. Inițiere în teleprelucrarea datelor. Ed. Tehnică 1972 (1).
177. Pisău G.ș.a. Elaborarea și introducerea sistemelor informatice. Ed. Tehnică 1977 (1).
178. Pennotti M, Schwartz M. Congestion Control in Store - and-Forward Tandem Links. IEEE Trans. Com. nr.12 1975 (3).
179. Pop V. Bazele logice ale calculatoarelor. Curs Instit. Politehnic Traian Vuia Pimișoara 1975 (1).

180. Pollaczek F. Probabilités stochastiques Posés par les Phénomène de Formation d'une queue d'Attente à une Guichet et par des Phénomènes Apparentes, Villars Paris 1957 (3).
181. Pouzin L. Presentation and Major Design Aspect of the Cyclades Computer Network. Proc. Data Com. Symp. Petersburg Florida 1973 (1) (4).
182. Pouzin L. Cigale the packet switching machine of the Cyclades computer network. Proc. IFIP Stockholm 1974 (1)(2).
183. Pouzin L. Network architecture and components. Proc. Comp. Net. IRIA Arles 1973 (2) (4).
184. Pouzin L. Interconnection of Packet Switching Network ARPA. Network Centre Bull (1) (2).
185. Pouzin L, Zimmerman H., A Tutorial on Protocol. Proc. IEEE nr.11 1978 (3).
186. Pouzin L, Grange L. Cigale la machine de Commutation de packets. Congres AFCET 1973 (1) (2) (3).
187. Pujolle G. Analysis of flow-controls in switching data network by a unified model. Proc. ICCO Kyoto 1978 (3).
188. Pujolle G. The Influence of Protocols on the Stability Conditions in Packet Switching Network. IEEE Transactions in Communication vol.27.nr,3 1979 (3<sup>\*</sup>).
189. Pujolle G. Files d'attente en serie et applications au taux de charge maximale d'un réseau d'ordinateurs. RAIRO vol 10 nr.5 1976 (3<sup>\*</sup>).
190. Pujolle G, Potier D. Accuracy of the Infinite Waiting Room Aproximation in Open and Closed Queueing Systemy. Proc. Inter. Symp. Model 1979 (3).
191. Pujolle G, Potier D. Réseaux de files d'attente a capacité limitée avec des applications aux systèmes informatiques. RAIRO vol.13 nr.2 1979 (3<sup>\*</sup>).
192. Reiser M. A Queueing Network Analysis of Computer Comunication Network with Window Flow Control. IEEE Trans. Com. nr.8 1979 (3).
193. Reiser M, Kobayashi H. Numerical Solution of Semiclosed .. Exponential Server Queueing Network. Proc. Conf. Computer 1973 (3).
194. Reiser M, Kobayashi H. Accuracy of the diffusion approximation for some queueing Systems. IBM J. Res. Dev. 18 1974 (3<sup>\*</sup>)

195. Reiser M, Kobayashi H. Queueing Networks with Multiple Closed chains. IBM Journal Res. Dev. vol.19 1975 (3).
196. Reiser M, Komhein A. Blocking in a Queueing Network with two Exponential Serves. IBM. Res. Aw. Rep. 1974 (3).
197. Roberts G. Telnet: Principles and practice. Proc. Computer Net. London 1975 (1) (2) (4).
198. Roberts G. The Evolution of Packet Switching. Proc. IEEE nr.11 1978 (1) (2).
199. Roberts G. Mesler B. The ARPA Network. Computer Com. Net. 1973 (1) (4).
200. Rogojan Al. Calculatoare numerice, curs. Inst. Politehnic Timișoara 1974 (1).
201. Rogojan Al. Metodă pentru sinteza schemei logice a unui calculator numeric. Inst. Pol. Timișoara 1974 (1<sup>x</sup>).
202. Rousset de Pina. Modèles déterministes pour l'étude de l'efficacité d'un système en temps partagé. RAIRO nr.3 1971 (3).
203. Rubin I. Communication Network: Messages Path Delays. IEEE Trans. Inform. Theory nr.20 1974 (2) (3).
204. Sâmbotin C., Tanasciuc C. Comutația statică în telemecanică. Ed. Tehnică 1965 (1) (2).
205. Sâmbotin C, Tanasciuc C. Comutație statică în automatică. Ed. Tehnică 1970 (1) (2<sup>x</sup>).
206. Sâmbotin C. Modul de organizare și structurare a sistemelor moderne de telemecanică industrială. Rev. Automatica și Electronica 1963 nr.3 (1) (2<sup>x</sup>).
207. Sâmbotin C.ș.a. Aspecte importante ale sistemelor moderne de telecomandă. Rev. Automatica și Electronica nr.5 1964 (1) (2).
208. Sâmbotin C.ș.a. Codurile și convertirea lor în sisteme moderne de telemăsură numerică. Ed. Tehnică publ. Autom. Telemec. Metrol. nr.3 1965 (1) (2).
209. Sâmbotin C. Sistem modern de telemăsură numerică. Ed. Tehnică publ. Aut. Telemec. Metrol. nr.5 1965 (1) (2).
210. Sâmbotin C. Sistem de telecomandă numerică. Ed. Tehnică publ. Aut. Telemec. metrol. nr.2 1966 (1) (2).
211. Sâmbotin C. Posibilitatea de realizare a sistemelor informatice care utilizează calculatoare electronice Felix



- C-256. Comunicare la sesiunea de comunicări științifice Inst. Politehnic Timișoara iunie 1974 (1) (2).
212. Sâmbotin C.ș.a. Sistemul informatic pentru conducere la o intrepr. constructoare de mașini. Comunicare la sesiunea de comunicări ICI 1975 (1).
213. Sâmbotin C.ș.a. Sistemul informatic pentru conducere producției la o întreprindere cu profil electrotehnic de serie mare. Comunicare Simpozion. CTCE Cluj 1977 și 1978 (1).
214. Sâmbotin C.ș.a. Sistem informatic pentru producție. Comunicare la Simpozion CTCE Timișoara 1979 (1).
215. Sâmbotin C. Proiect tip pentru conducerea producției în intr. din industria construcțiilor de mașini. Comunicare la sesiunea ICI 1978 și 1979 (1).
216. Sâmbotin C.ș.a. Produs program pentru conducerea producției. Comunicarea la sesiunea jubiliară de comunicări științifice ICI 1980 Buc. (1).
217. Schweitzer P, Lam S. Buffer over flow in a store-and-forward network. node IBM J. Res. Dev. 1976 (3).
218. Stone J. The Large Scale Data Processing Network. Chiron. Paris 1969 (1).
219. Tabagi F, Gerla M, Peebls R, Manning B. Modeling and Measurement Techniques in Packet Communication Networks. Proc. IEEE nr.11 1978 (3).
220. Weckler S. Computer Network Architectures. IEEE Comp. vol.12 1979 (2) (4).
221. Vissotsky V, Corbato F, Graham R. Structure of MULTICS Supervisor. FJCC 1968 (1) (2) (4).
222. Wilson C. The infrastructure for European Teleinformatics Computer Network vol.3 nr.4 1979 (1).
223. Wirth N. A language for modular multiprogramming. ETH Zurich 1976 (1) (4).
224. Worley A. Practical aspects of data communications. Datamatig nr.10 1969 (1).
225. Zimmerman H. The Cyclades end-to-end Protocol. Proc. ACM Symp. Quebec 1975 (3).
226. Zimmerman H. The Cyclades experience; Results and impact. Proc. IFIP Congress 1977 (1) (4).