

Sisteme de operare. Mecanisme interne și principii de proiectare

1. Introducere

Cursul tratează conceptele și principiile fundamentale ale teoriei și practicii sistemelor de operare (SO). Sunt prezentate definițiile principale și unele clasificări, interfețele sistemelor de operare, modul de organizare a procesului de calcul, gestionarea memoriei și a dispozitivelor periferice, administrarea informației și gestiunea fișierelor. Lucrarea acoperă aspectele asociate bazelor contemporane ale metodelor și mijloacelor de elaborare a resurselor program de sistem (inclusiv, operații asincrone, tratarea întreruperilor, compromisele dintre dispozitivele tehnice și resursele program, interfețele sistemelor de operare), având drept obiectiv final pregătirea cititorului pentru analiza și proiectarea sistemelor de operare. Cursul include capitole teoretice și materiale ilustrative practice, chemate să familiarizeze cititorul nu atât cu modalitatea utilizării unor sisteme de operare concrete, ci în special cu metodele și algoritmii, care stau la baza mecanismelor interne ale sistemelor de operare și modul în care acestea sunt concepute, proiectate și implementate.

Un sistem de calcul constă din două tipuri de resurse: resursele fizice și resursele logice. Resursele fizice posedă caracteristici tehnice avansate și pot fi utilizate în cele mai diverse scopuri. Însă aceste resurse fără componentele logice de sistem (software de sistem) întâmpină dificultăți mari în relația cu mediul în care trebuie să funcționeze. *Acesta este unul dintre motivele principale ale creării sistemelor de operare, destinația cărora este administrarea (gestiunea, controlul) resurselor tehnice principale și asigurarea unei interfețe comode (plăcute, prietenoși) între utilizator și calculator* (fig.1.1, [1]).

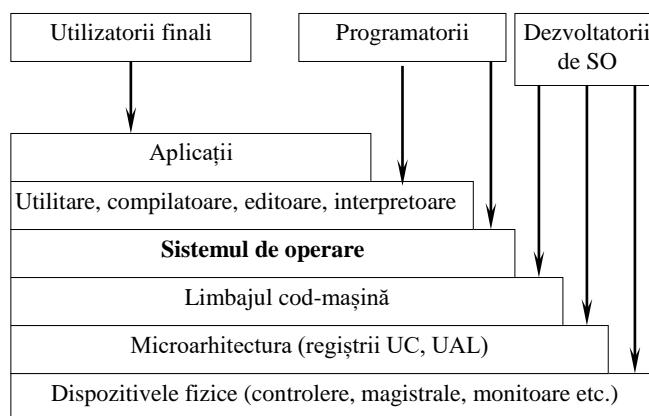


Fig.1.1. Locul sistemului de operare în cadrul unui sistem de calcul

Există mai multe motivații ale necesității studierii de către viitorii specialiști IT a sistemelor de operare, cele mai importante fiind următoarele:

- pentru utilizarea resurselor hardware în scopuri speciale poate fi necesar să fie creat un sistem de operare propriu sau să se introducă modificări într-unul existent;
- de alegerea corectă a sistemului de operare și a versiunii concrete poate depinde viabilitatea și eficacitatea sistemului de calcul;
- este ideal ca utilizatorul să interacționeze cu sistemul de operare cunoscând toate subtilitățile ultimului, deoarece sistemul de operare este un intermediar între calculator și utilizator;
- multe metode și concepte, utilizate în domeniul SO, pot fi implementate cu succes și în alte domenii.

Prin noțiunea **sistem de operare** înțelegem în primul rând modulele program ale unui sistem de calcul, care administrează resursele tehnice (procesoare, memoria operativă și secundară, dispozitive de intrare/ieșire, fișiere). Modulele în cauză soluționează situațiile de conflict, optimizează productivitatea sistemului, sporesc eficiența utilizării lui. Ele sunt un fel de intermediar (interfață) între programele utilizatorului și componentele tehnice ale calculatorului. Alte denumiri istorice: program de administrare, monitor, supervisor.

Modulele destinate unor domenii anume, cum ar fi traductoarele, depanările, bibliotecile, mediile integrate de dezvoltare etc., nu sunt incluse în definiția unui SO, fiind considerate și ele „utilizatori” ai sistemului de operare.

1.1. Noțiuni de bază și clasificări

Un calculator constă dintr-un ansamblu de componente funcționale fizice și logice, care cooperează pentru a satisface cerințele utilizatorilor privind introducerea, stocarea, prelucrarea, transmisia și căutarea informațiilor. Aceste componente funcționale sunt structurate pe niveluri, care interacționează prin interfețe bine definite.

Suportul fizic (*resurse tehnice, hardware*) constituie nivelul inferior al sistemului de calcul construit pe baza unor componente electronice, magnetice, optice, mecanice etc., mai mult sau mai puțin sofisticate în funcție de stadiul de dezvoltare a tehnologiilor respective.

1.1.1. Noțiuni și termeni din domeniul resurselor tehnice

Pentru a trece la noțiunile principale, legate de hardware, vom face cunoștință mai întâi cu funcțiile de bază ale unui calculator. Pot fi evidențiate cinci funcții esențiale [2]: *inițializarea (bootup)*, *introducerea datelor*, *procesarea datelor*, *stocarea datelor* și *extragerea rezultatelor*:

- **Inițializarea** implică testarea părților importante ale calculatorului, rularea fișierelor de pornire și încărcarea altor fișiere necesare, cum ar fi driverele de dispozitive;
- **Introducerea** reprezintă transferul datelor dintr-o sursă externă în calculator. Surse externe pot fi dischetele, tastatura, mouse-ul etc.;
- **Procesarea** se referă la manipularea datelor introduse în scopul producerii unui rezultat (ieșirea);
- **Stocarea** constituie procesul salvării informațiilor (date sau programe) într-un dispozitiv de păstrare, de exemplu discul fix, pentru recuperarea ulterioară.

Prin **structura** unui calculator vom înțelege componentele (dispozitivele) care formează calculatorul și legăturile dintre ele. Componentele principale sunt: *procesorul*, *memoria*, inclusiv *unitățile de stocare pe termen lung*, *dispozitivele de intrare-ieșire* (tastatura, display-ul, mouse-ul etc.). Relațiile (legăturile) dintre aceste componente pot fi foarte variate, o structură devenită apoi clasică, este **structura John von Neumann** (fig.1.2).

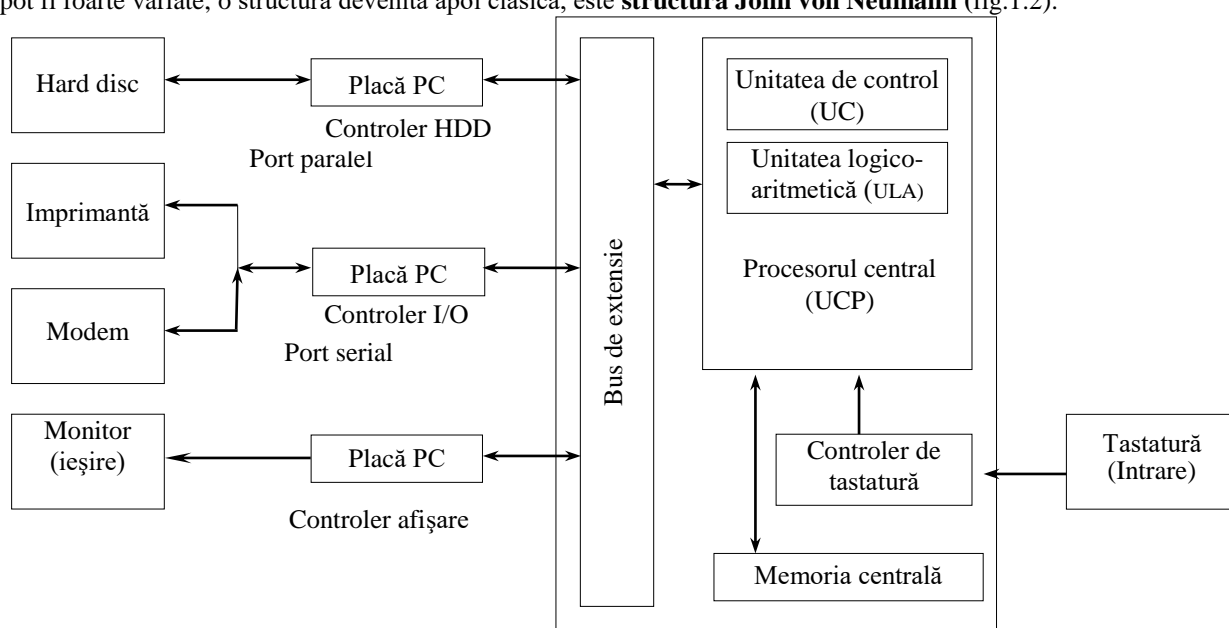


Fig. 1.2. Structura unui calculator

Componentele principale se conectează la placa de bază (motherboard) direct sau prin conectoare speciale, numite plăci de extensie (daughterboards). **Unitatea centrală de procesare – procesorul (CPU)** se găsește într-un singur circuit integrat (cip) incorporând **unitatea de comandă (Control Unit, CU)** și **unitatea logico-aritmetică (Arithmetic Logical Unit, ALU)**.

Unitatea de comandă controlează funcționarea unității logico-aritmetice. **Memoria** este o zonă de lucru de mare viteză, unde sunt stocate datele și programele pentru a fi accesate de CPU în mod rapid. Memoria poate fi organizată în mod ierarhic, caz în care există cel puțin două nivele de ierarhie – **memoria centrală** (operativă, internă) și **memoria secundară** (externă, de lungă durată). Memoria operativă este electronică sub formă de cipuri, de obicei cu acces aleator (RAM – Random Access Memory) și trebuie să fie alimentată cu tensiune pentru a păstra datele (memorie vie). Pentru salvarea datelor atunci când se întrerupe alimentarea sau pentru păstrare de lungă durată, datele sunt stocate în memoria secundară, care le reține oricât de mult timp. Dispozitivele cele mai obișnuite de introducere a datelor sunt tastatura și mouse-ul, iar dispozitivul de ieșire cel mai utilizat este monitorul. Procesorul și memoria operativă formează **nucleul** calculatorului, toate celelalte dispozitive fiind cunoscute sub denumirea de **periferie** (dispozitive periferice).

Instrucțiunile care vor fi îndeplinite de calculator sunt stocate în memorie sub formă de programe. Unitatea de comandă ține evidența și interpretează instrucțiunile dintr-un program, fiind responsabilă cu transmiterea de sarcini

specifice diferitelor elemente ale calculatorului. Unitatea de control controlează în principal funcțiile de intrare-ieșire (I/O), de memorizare și stocare, colaborează cu ALU, care răspunde de efectuarea operațiilor de calcul.

Mai menționăm noțiunile: mărimea magistralelor interne și externe de date, mărimea adresei de memorie, frecvența ceasului.

Mărimea magistralei interne de date: procesoarele păstrează datele în locații speciale, numite **registre**. Datele sunt transferate între registre, CU, ALU și alte componente ale procesorului prin intermediul unei magistrale realizate în circuitele procesorului. Numărul de linii în această magistrală oferă o măsură a cantității de date pe care procesorul o poate transfera într-o singură operație. Valoarea ei poate varia între 8 și 32 de biți (în scopuri speciale 64, 128 sau chiar mai mult).

Mărimea magistralei externe de date măsoară câte date pot fi transferate între procesor și dispozitivele periferice într-un tact de ceas. Magistrala este un sistem de conectări și cablări ce distribuie datele prin calculator. Cu cât magistrala de date e mai mare, cu atât performanțele calculatorului se îmbunătățesc. Numărul de linii și aici variază între 8 și 32 de biți.

Mărimea adresei de memorie determină volumul de memorie care poate fi gestionat de calculator fără eforturi speciale. Prin eforturi speciale înțelegem resurse fizice sau logice, care permit gestionarea unei memorii cu capacitate mai mare decât volumul obținut prin calcularea adresei reieșind din numărul de linii fizice ale memoriei.

Un **ceas** electronic (generator de sincronizare) asigură coordonarea corespunzătoare a componentelor calculatorului. Componentele calculatoarelor cu performanțe superioare pot opera la **frecvențe de ceas** mai mari. Frecvența ceasului indică viteza de operare a CPU și se măsoară în miliarde de impulsuri pe secundă, adică în *GigaHertzi* (GHz).

1.1.2. Noțiuni și termeni din domeniul sistemelor de operare

Un **sistem de operare** este un ansamblu de programe de control și de serviciu care ghidează un calculator în executarea sarcinilor sale și asistă programele de aplicație și utilizatorul prin intermediul anumitor funcții. Natura funcțiilor și modul în care acestea sunt realizate determină **atributele** care caracterizează un sistem de operare: *timpul de răspuns, simultaneitatea utilizării, eficiența, partajarea și protecția, universabilitatea, flexibilitatea, extensibilitatea, fiabilitatea și disponibilitatea, transparența și vizibilitatea* [3].

Timpul de răspuns exprimă durata intervalului delimitat de lansarea unei cereri de serviciu și achitarea acesteia de către sistem. Are două componente: timpul de așteptare pentru ca cererea să fie luată în considerație și timpul de execuție a acestei cereri.

Simultaneitatea utilizării măsoară gradul în care un sistem poate să execute în același timp mai multe lucrări.

Eficiența măsoară proprietatea unui sistem de a folosi în mod optim resursele de care dispune.

Partajarea și protecția caracterizează nivelul la care utilizatorii au posibilitatea să utilizeze în comun informația prezentă în sistem și nivelul la care pot să comunice între ei, în deplină siguranță (în sensul evitării accesului neautorizat și/sau alterării intenționate sau accidentale a informației).

Universabilitatea, flexibilitatea, extensibilitatea măsoară gradul în care un sistem poate fi folosit (universabilitate) și adaptabil (flexibilitate) unui context specific, exprimat prin nivelul de limitare impus programelor utilizatorului, precum și gradul în care se pot include în sistem noi componente hardware și software fără eforturi de proiectare și programare suplimentare (extensibilitate).

Fiabilitatea și disponibilitatea exprimă proprietatea unui sistem de a cădea foarte rar în pană și de a evita goluri în funcționare din cauza defectării uneia sau mai multor componente ale sale.

Transparența și vizibilitatea exprimă pe de o parte proprietatea unui sistem de a face invizibil utilizatorului ceea ce se află sub interfața de utilizare care i se oferă și, pe de altă parte, capacitatea de a permite utilizatorilor săi să obțină anumite informații despre modul cum el lucrează, informații de care în mod teoretic ei nu au nevoie pentru a beneficia de o utilizare completă, însă care ar putea să-i ajute la obținerea unei utilizări mai eficiente [3].

Resursele program reprezintă seturi de programe și date utilizate pentru soluționarea anumitor probleme. **Programul** este transcrierea într-un limbaj de programare a unui algoritm, altfel – programul este o secvență de instrucțiuni sau simplu **cod**. **Utilizatorul** este oricare doritor să îndeplinească anumite lucrări la calculator. Prin **lucrare (sarcină, task)** vom înțelege un set de acțiuni, necesare pentru îndeplinirea unui lucru anume. Sarcina poate conține mai mulți **pași**. **Pașii de sarcină** sunt unități de lucru, care vor fi îndeplinite consecutiv, de exemplu trei pași – compilare, încărcare și executare. Primind o lucrare de la utilizator, sistemul de operare poate crea câteva **procese**, prin proces înțelegând la moment, **calcule** care pot fi efectuate paralel cu alte calcule.

Procesul mai poate fi definit drept traiectoria procesorului, atunci când ultimul îndeplinește un set oarecare de programe. Ansamblul programelor și datelor accesate în timpul procesului, formează **spațiul de adrese**. Una din destinațiile sistemului de operare este de a asigura **proiectarea** spațiului de adrese a unui proces în memoria fizică. Pentru rezolvarea acestei probleme sunt utilizate atât resurse tehnice (sisteme cu organizarea memoriei pe segmente sau pe pagini), cât și resurse program speciale.

Multiprogramarea este un termen utilizat în cazul unui sistem în care pot exista simultan câteva procese în etapa de execuție. Un proces se consideră în **execuție**, dacă calculele au început, dar la momentul considerat nu au fost terminate sau întrerupte (terminare din cauza unei erori sau din alte motive). Nu este obligatoriu ca un proces care se află în **execuție** să fie și executat de procesor la un moment dat.

Resursele hardware de **protecție** sunt utilizate cel mai des pentru controlul accesării memoriei. **Întreruperea** este un mecanism care impune procesorului să observe anumite evenimente. Pot exista mecanisme care permit să nu se acorde atenție unei anume întreruperi – **întrerupere mascată**. Resursele hardware de **întrerupere** permit sistemului de operare să coordoneze operațiile simultane; pot fi utilizate și pentru a schimba ordinea de execuție a programelor.

1.1.3. Tipuri și exemple de sisteme de operare

Valorile concrete ale atributelor sistemelor de operare și combinații ale acestora determină diverse tipuri de sisteme și restricții de implementare. Conform acestor atribute pot fi evidențiate sisteme de operare [3]:

- secvențiale
- cu multiprogramare
- cu prelucrare multiplă
- în timp real, etc.

Majoritatea sistemelor de operare recunosc **programul** ca **cea mai mică unitate de prelucrare**, căreia i se atribuie o identitate și pe care un utilizator o poate prezenta spre execuție. Unele sisteme permit ca un program să fie considerat ansamblu de sarcini ale căror execuții (inclusiv în paralel) contribuie la atingerea obiectivului urmărit de acest program.

Un sistem **secvențial** (tratare pe loturi, batch processing en., traitement par lots fr.) execută la un moment dat un singur program, care trebuie terminat înainte de a lansa un alt program în execuție.

Sistemele cu **multiprogramare** acceptă la un moment de timp dat mai multe programe în memoria centrală, acestea aflându-se în diverse stadii de execuție.

Un sistem de calcul cu **prelucrare multiplă** dispune de mai multe procesoare, care pot să execute simultan unul sau mai multe programe. Utilizarea efectivă a prelucrării multiple necesită atributul de multiprogramare. Execuția simultană a unui singur program de către mai multe unități presupune existența posibilității de a descompune acest program în mai multe **procese** sau **fire de execuție**.

Sistemele în **timp real** sunt dedicate, de obicei, funcționării în cadrul unor sisteme de comandă și este necesar ca valorile anumitor atribute să se încadreze în anumite limite, dictate de dinamica proceselor comandate.

Clasificarea pe tipuri de sisteme de operare enumerate mai sus nu este nici disjunctă și nici exhaustivă. Majoritatea sistemelor existente pot fi încadrate în mai multe clase, atunci când se face o analiză prin prisma obiectivelor pe care le urmăresc. La capitolul obiective vom aminti în primul rând **maximizarea eficienței** sistemului de calcul și a **satisfacției** utilizatorilor. Tot la obiective poate fi trecută și cererea de **minimizare a posibilității de apariție a erorilor** și de **maximizare a transparenței** sistemului de operare, **garantarea securității** datelor, **optimizarea controlului comunicațiilor** în cazul SO de rețea.

Un obiectiv foarte important este **minimizarea efortului concepție-realizare** a sistemului, ultim în enumerare, dar foarte important pentru specialiști.

Toate aceste obiective sunt consecințe ale dezideratului principal: un sistem de operare **este destinat să administreze** resursele sistemului de calcul și anume memoria, procesorul (procesoarele), dispozitivele și informația.

Un sistem de operare este obligat:

- să păstreze informația despre starea fiecărei resurse
- să ia decizia cărui proces să i se aloce resursa, în ce cantitate și când
- să aloce resursa și
- la momentul oportun să o retragă.

Exemplele care urmează [4] vor ilustra diversitatea funcțiilor îndeplinite de către un sistem de operare, fără pretenții de exhaustivitate. Pentru fiecare exemplu vom indica funcțiile puse în șarja sistemului de operare și caracteristicile principale ale acestuia.

1.1.3.1. Cazul calculatoarelor personale

Configurația cea mai simplă a unui calculator personal (PC) include o unitate centrală, o memorie principală, un display, o tastatură și un mouse. Această configurație, de obicei, este completată de o memorie secundară și o imprimantă (fig.1.3).

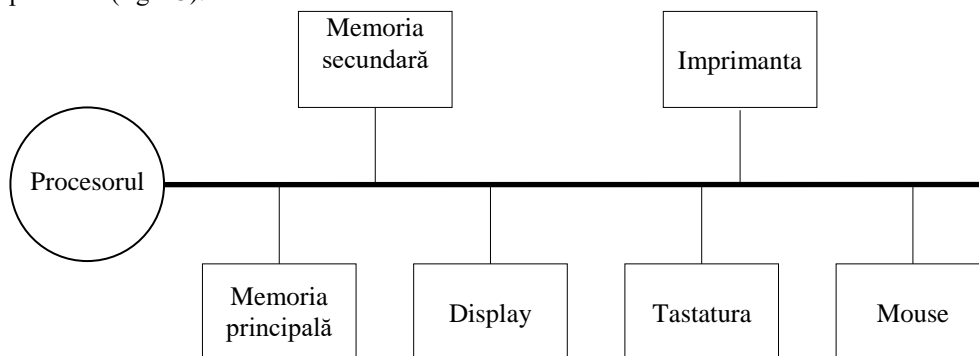


Fig. 1.3. Structura unui calculator personal

Utilizatorul unui atare sistem va cere minimum următoarele două tipuri de servicii:

- executarea unor programe existente în PC sau introduse sub formă de fișiere; introducerea datelor necesare pentru executarea programului (de la tastatură, din fișier sau din alte surse); listarea rezultatelor la display, imprimantă sau copierea lor într-un fișier;
- identificarea și crearea unor fișiere sau mulțimi structurate de informații; stocarea acestor fișiere în memoria secundară; transferarea informațiilor între fișiere și dispozitivele de intrare/ieșire.

Sistemul de operare poate acorda aceste servicii prin intermediul unui limbaj special, numit **limbaj de comandă**, introducându-se de la tastatură instrucțiuni de forma <acțiune> <parametri>, sau utilizând mouse-ul și o interfață grafică a utilizatorului (GUI - graphical user interface), acțiunile mouse-ului fiind imediat interpretate de sistem.

Iată două exemple de secvențe tipice de activități în cazul unui PC:

- elaborarea unui program;
- introducerea programului cu ajutorul tastaturii și a unui editor de texte;
- executarea programului introducându-se datele necesare de la tastatură și extrăgând rezultatele la display sau imprimantă;
- modificarea programului, dacă rezultatele nu sunt satisfăcătoare și repetarea execuției;
- perfectarea versiunii finale a programului, inclusiv documentarea la necesitate a acestuia;
- exploatarea unui program;
- cererea de executare a unui program deja existent. Vor fi pregătite în prealabil date de intrare sau acestea vor fi introduse în mod interactiv la cerere cu ajutorul tastaturii;
- afișarea rezultatelor pe ecran, listarea la imprimantă sau copierea lor într-un fișier pentru o utilizare ulterioară.

Într-un atare sistem funcția *partajare a resurselor* poate fi lipsă, or PC-ul este folosit de un singur utilizator care are controlul total asupra acestuia. Alocarea resurselor este legată de gestionarea memoriei și administrarea fișierelor. Funcțiile principale vizibile ale sistemului de operare constau în administrarea fișierelor, realizarea operațiilor de intrare/ieșire și interpretarea comenzilor provenite de la interfața utilizator-sistem de operare.

Pentru acest tip de sisteme cele mai importante caracteristici sunt:

- fiabilitatea;
- eficacitatea;
- simplitatea utilizării;
- facilitatea extensibilității prin adăugarea unor utilite noi sau adaptarea la periferice noi.

Ultimele două aspecte pun în evidență importanța interfețelor oferite de sistem (limbajul de comandă sau GUI).

1.1.3.2. Comanda unor procese industriale

La o uzină chimică sunt utilizate două materii prime *A* și *B* pentru sinteza produsului *C* conform fig. .4. Procesul de producere este comandat de un calculator care îndeplinește următoarele funcții:

- *Reglare*. Pentru o derulare bună a procesului de fabricație parametrii de funcționare (temperatura, presiunea, concentrația, etc.) trebuie să se afle într-o plajă de valori predefinite. Pentru aceasta va fi acționat debitul de intrare a materiilor prime *A* sau *B*. Parametrii de funcționare sunt mășurați cu ajutorul unor captoare.

Calculatorul preia aceste măsurări și, în dependență de algoritmul de comandă, acționează robinetele de intrare.

- **Înregistrare.** Rezultatele măsurărilor sunt periodic înregistrate; valorile lor sunt afișate pe un tablou de bord și recopiate într-un fișier ("jurnal de bord") în scopul unor prelucrări ulterioare (date statistice).
- **Securitate.** În cazul în care unul dintre parametrii mășurați depășește o valoare critică predefinită reactorul trebuie oprit imediat.

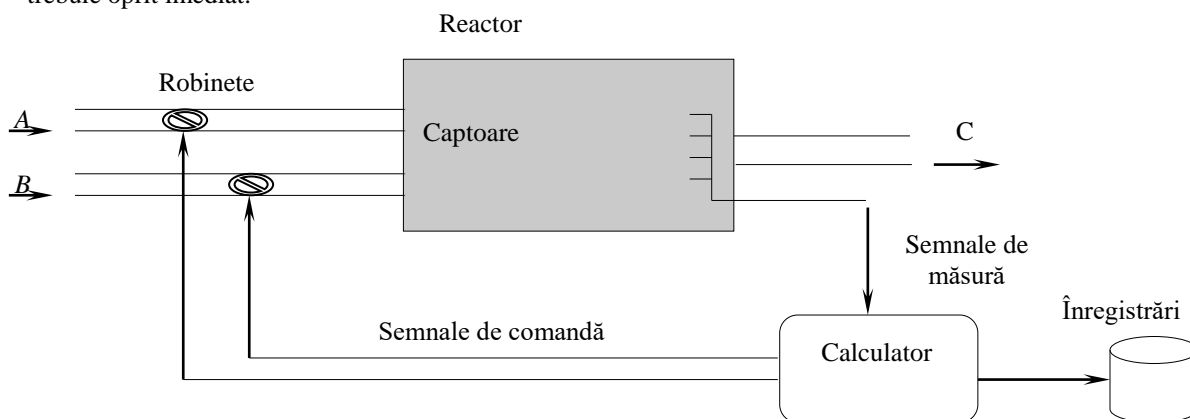


Fig. 1.4. Schema unui proces chimic

Acest mod de funcționare introduce unele restricții:

1. Măsurările se produc periodic; fie T valoarea intervalului de timp dintre două măsurări consecutive (perioada de eșantionare), iar t - timpul total de prelucrare a datelor de către calculator (măsurarea propriu-zisă a semnalelor observate, înregistrarea, calcularea semnalelor de comandă și acționarea robinetelor). Sistemul va funcționa doar în cazul respectării relației $t \leq T$.
2. Securitatea sistemului are prioritate maximă. Depășirea unor valori critice trebuie să fie detectată în orice moment și tratarea acestor accidente va întrerupe toate operațiile în curs de execuție.

Funcțiile principale ale sistemului de operare sunt:

- acționarea organelor externe (citirea semnalelor captoarelor, comanda robinetelor);
- evidența timpului real (declanșarea periodică a ciclului de calculare a semnalelor de comandă);
- reacția la evenimentele exterioare (oprire de urgență);
- gestiunea informațiilor (păstrarea și întreținerea jurnalului de bord).

Existența unor restricții stricte privind durata de prelucrare a informațiilor, noțiunea de tratare prioritară, conectarea la dispozitive exterioare de măsurare și acționare sunt caracteristice aplicațiilor informatice *în timp real*. Pot fi menționate și alte domenii cu comandă în timp real: centralele telefonice, comanda aparatelor de zbor, robotica, monitoringul medical, etc.

În cazul acestor sisteme **caracteristica principală** este **fiabilitatea**, or rezultatele unei funcționări neadecvate pot fi catastrofale. Sistemul trebuie să garanteze un serviciu minim în cazul unor căderi în pană a dispozitivelor tehnice, unor evenimente accidentale sau erori umane.

1.1.3.3. Sisteme tranzacționale

Caracteristicile principale ale sistemelor tranzacționale sunt următoarele:

- sistemul gestionează un set de informații sau baze de date, care pot atinge volume importante;
- asupra acestor informații pot fi executate anumite operații predefinite, sau tranzacții, adesea interactive;
- sistemul este dotat cu un mare număr de puncte de acces și un mare număr de tranzacții pot derula simultan.

Ca exemplu pot fi menționate sistemele de rezervare a biletelor, de gestiune a conturilor bancare, de arhivare și consultare a documentelor.

Restricțiile sunt în primul rând legate de integritatea și coerența internă a informațiilor, care formează bazele de date. Aceste restricții depind, evident de aplicație. De exemplu, numărul de locuri rezervate într-un avion nu poate depăși numărul locurilor disponibile, un loc distinct poate fi atribuit unei singure persoane, etc.

Calitățile obligatorii ale unui sistem tranzacțional sunt disponibilitatea și fiabilitatea; pentru unele sisteme poate fi importantă și toleranța la defecțiuni. O caracteristică importantă ale sistemelor tranzacționale este multitudinea activităților paralele, iar în multe cazuri și repartizarea geografică a componentelor.

1.1.3.4. Sisteme cu partajarea timpului

Destinația principală a unor astfel de sisteme este furnizarea serviciilor necesare unei mulțimi de utilizatori, fiecare dintre ei beneficiind de servicii:

- echivalente serviciilor unui calculator individual;
- legate de existența unei comunități de utilizatori: partajarea informațiilor, comunicații între utilizatori.

Problemele care apar datorită conceptului de partajare a timpului sunt o combinație a problemelor existente în cazul unui calculator individual cu cele din sistemele tranzacționale și pot fi clasificate după cum urmează:

- definirea mașinii virtuale, oferite fiecărui utilizator;
- partajarea și alocarea resurselor fizice comune: procesoare, memorii, organe de comunicație;
- gestionarea informațiilor partajate și a comunicațiilor.

Caracteristicile obligatorii unui atare sistem combină în egală măsură calitățile unui sistem de operare al unui calculator individual și al unui sistem tranzacțional: disponibilitatea, fiabilitatea, securitatea, exploatarea optimă a resurselor fizice, calitatea interfeței și serviciilor utilizatorului, simplitatea adaptării și extensibilității.

1.2. Puncte de vedere asupra sistemelor de operare

1.2.1. Sistemul de operare și procesele

Noțiunea de **proces**, introdusă mai sus, este asociată conceptului de lucrare (pentru a lua în considerare aspectele dinamice) și poate fi definită altfel ca o **suită temporală de execuții de instrucțiuni**, fiind o entitate de bază în descrierea sau analiza funcționării unui sistem de operare. Evoluția în timp a unui proces presupune un consum de resurse, dictat de natura și complexitatea instrucțiunilor de executat. Orice utilizare a unei resurse este asociată, la un moment dat, unui proces și procesul respectiv își asumă răspunderea utilizării acestei resurse. În particular, rezultă că ori de câte ori se execută procedurile de sistem, resursele pe care le utilizează sistemul intră în administrarea procesului (fie el și al utilizatorului), care a cerut serviciul. Mulțimea resurselor (procesorul, memoria centrală, informația, dispozitivele) alocate unui proces variază în timp (dinamica procesului).

Anterior un sistem de operare a fost definit ca un set de programe destinat să administreze resursele. Care sunt relațiile dintre programele sistemului de operare? Atunci când un proces este creat, care este ordinea de utilizare a unui anume program al SO? Pentru a răspunde la aceste întrebări (și la altele) vom face cunoștință cu ciclul de viață a unui proces. În fig. 1.5 sunt prezentate trei procese (trei sarcini ale utilizatorilor, lansate în execuție) existente într-un sistem cu multiprogramare.

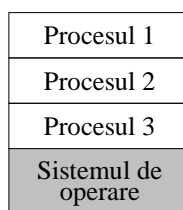


Fig. 1.5. Trei procese într-un sistem cu multiprogramare

Ciclul de viață a unui proces poate fi reprezentat printr-un set de stări ale procesului și tranzițiile de la o stare la alta. Vom evidenția trei stări **elementare** ale unui proces: proces **ales** (sau *exe*) – *procesului i s-a alocat un procesor, este în curs de execuție*, proces **blocat** (*wait*) – *procesul așteaptă să se producă un anumit eveniment a cărui apariție este indispensabilă*, proces **eligibil** (*ready*) – *procesul are la dispoziție toate resursele necesare lipsă fiind doar procesorul, adică este pregătit să fie executat din momentul alocării unității centrale* (fig. 1.6).

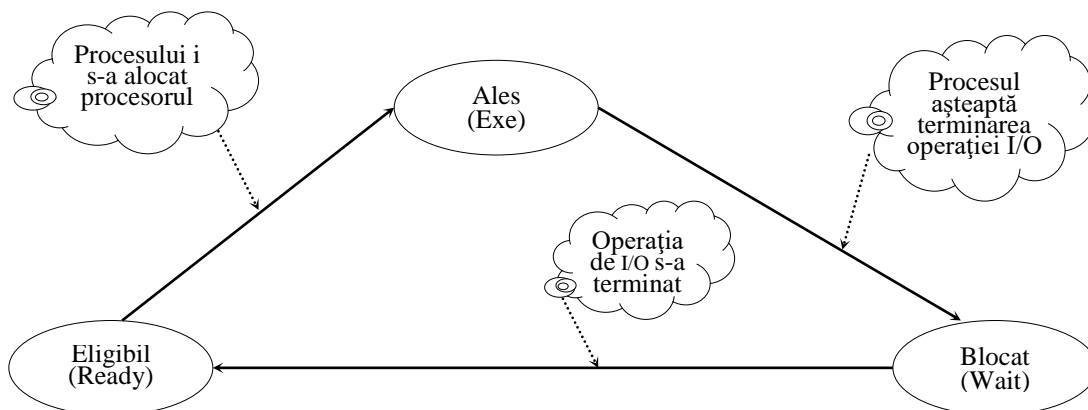


Fig. 1.6. Stările elementare din ciclul de viață a unui proces

Modelul prezentat în figura 1.6 este unul puternic simplificat, în realitate existând abateri substanțiale în direcția complexității (fig. 1.7). Sistemul de operare este aceea componentă, care va ghida procesul prin toate aceste stări.

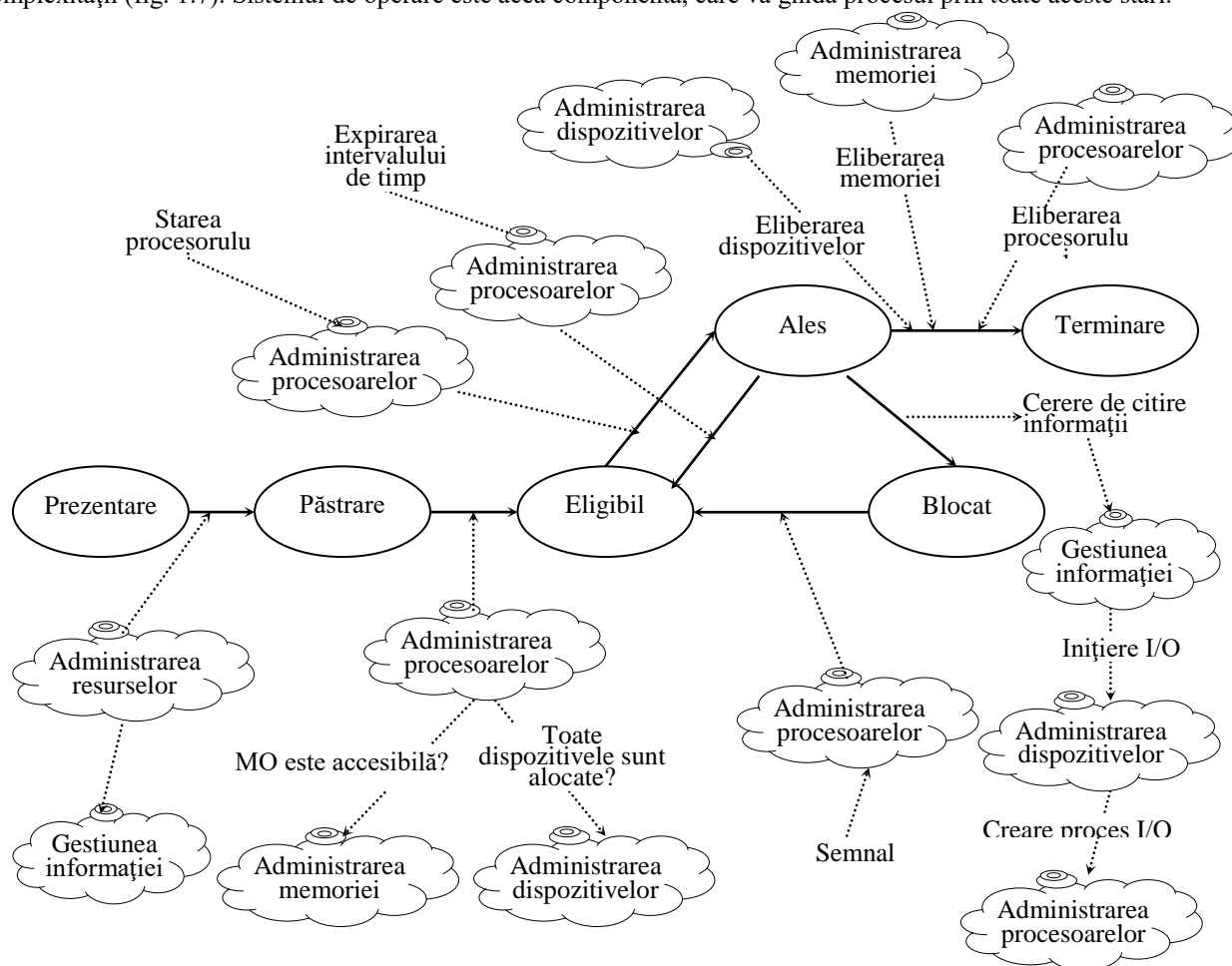


Fig. 1.7. Modelul stărilor unui proces și componentele SO responsabile de tranziții

1.2.2. Mașină ierarhică și mașină extinsă

Astăzi este greu de închipuit că doar cu câteva decenii în urmă un utilizator era nevoit să programeze în zerouri și unități, utilizând o “mașina goală”. Chiar și specialiștii nu prea iubesc să scrie programe într-un limbaj de asamblare sau (și mai evident) utilizând doar instrucțiuni din setul, garantat de resursele fizice. Un program elaborat de un specialist poate fi de forma [1]:

- | | |
|--------------------------|--|
| 1. Transferă C, B | Stabilește C=B |
| 2. Găsește zona 80, X | Să se găsească 80 de octeți de memorie liberi și să se plaseze adresa zonei în X |
| 3. Introdu date în X | Să se citească datele indicate în zona X |
| 4. Compară X(2), '/' | Coincide conținutul primilor 2 octeți ai zonei X cu '/'? |
| 5. Dacă da , STOP | Dacă coincide, salt la STOP |

Instrucțiunile 1, 4 și 5 sunt instrucțiuni standard în multe calculatoare contemporane. Însă execuția corectă și eficientă a instrucțiunilor 2 și 3 poate solicita îndeplinirea a zeci, sute sau chiar mii de comenzi din setul standard, deoarece aceste instrucțiuni cer interacțiunea cu unele resurse cheie, cum ar fi memoria operativă și dispozitivele de intrare-ieșire. Sistemul de operare asigură instrucțiuni pentru realizarea unor astfel de funcții de administrare a resurselor: instrucțiunile 2 și 3 sunt instrucțiuni ale **mașinii extinse** ele neavând echivalenți în setul de instrucțiuni hardware. Sistemul de operare completează setul standard, realizat hardware, cu instrucțiuni de acest gen.

Setul de instrucțiuni realizat hardware împreună cu instrucțiunile suplimentare ale sistemului de operare formează **sistemul de comenzi al mașinii extinse**. Grafic conceptul de mașină extinsă poate fi reprezentat conform fig. 1.8. Sistemul de operare este executat pe mașina “goală”, iar programele utilizatorului – pe mașina extinsă.

Să facem cunoștință acum cu modul în care sunt organizate într-un tot întreg componentele sistemelor de operare.

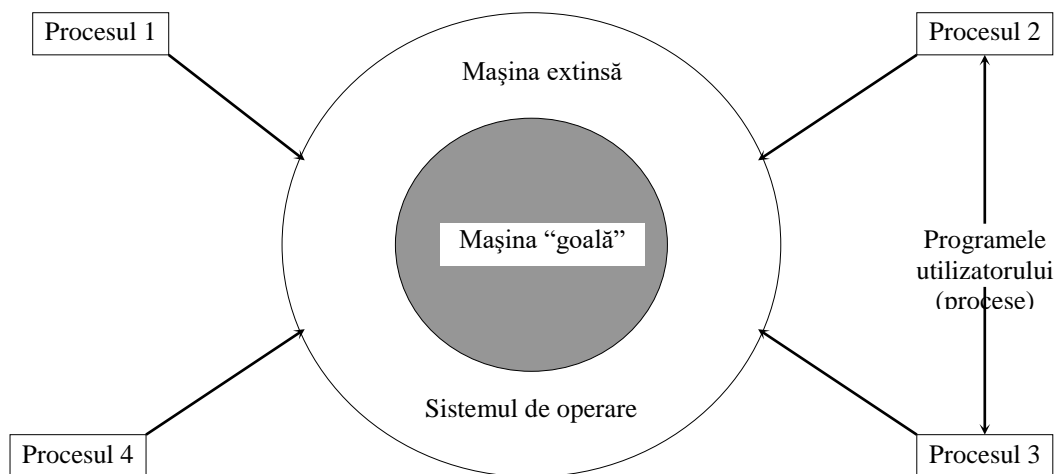


Fig. 1.8. Mașina extinsă

Primele sisteme de operare erau formate dintr-un singur program mare. Dar, odată cu sporirea complexității sistemelor, această abordare liniară conducea la dificultăți serioase și s-a propus să se utilizeze și în acest domeniu conceptul de *mașină extinsă*. Acest concept, în cazul sistemelor de operare, poate fi utilizat în două nivele (fig. 1.9) și conduce la noțiunea de *mașină ierarhică* [4]: primul nivel - funcțiile cheie, utilizate de majoritatea modulelor de sistem, pot fi realizate în cadrul unei *mașini extinse interne* și nivelul doi - unele module pot fi executate în cadrul unei *mașini extinse externe*, analogic proceselor utilizatorului.

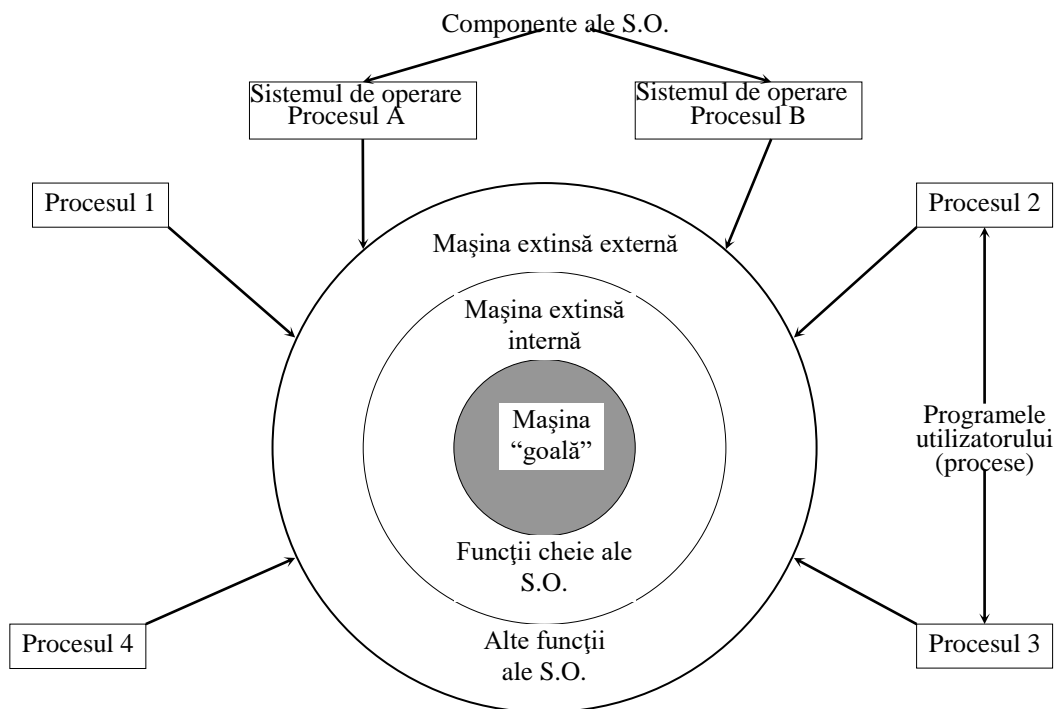


Fig. 1.9. Ilustrarea conceptului de mașină ierarhică

În această abordare ierarhică apare imediat problema alegerii corecte a nivelului de ierarhie pentru fiecare modul al sistemului de operare (în mașina extinsă internă, externă sau va fi prezent în calitate de proces extern). Pot fi evidențiate subnivele ale celor două mașini extinse, iar interacțiunea proceselor sistemului de operare ne conduce la necesitatea introducerii mai multor *straturi ale proceselor*. Modulele sistemului, plasate în cadrul mașinii extinse, spre deosebire de modulele care aparțin straturilor proceselor, formează **nucleul** sistemului de operare. Concepția mașinii ierarhice este pe larg utilizată în proiectarea programelor mari utilizând metodele, cunoscute sub denumirea *programare modulară* sau *programare structurală*. Nu există reguli stricte în privința numărului de nivele, amplasării modulelor pe nivele, componența nucleului. De obicei, nucleul conține doar cele mai necesare și evidente funcții, celelalte, atunci când este posibil, vor fi prezente ca procese de sistem separate. O detalizare a conceptului de

mașină ierarhică este adusă în fig. 1.10. Procesele (incluse în dreptunghiuri) se adresează către funcțiile nucleului și utilizează împreună sursele sistemului. Unele procese generează sau comandă alte procese (granița dintre ele este prezentată de liniile în zig-zag, care separă diferite straturi ale proceselor). Într-o realizare strict ierarhică modulele, situate într-un nivel oarecare, pot accesa (se pot adresa) numai resursele nivelelor inferioare.

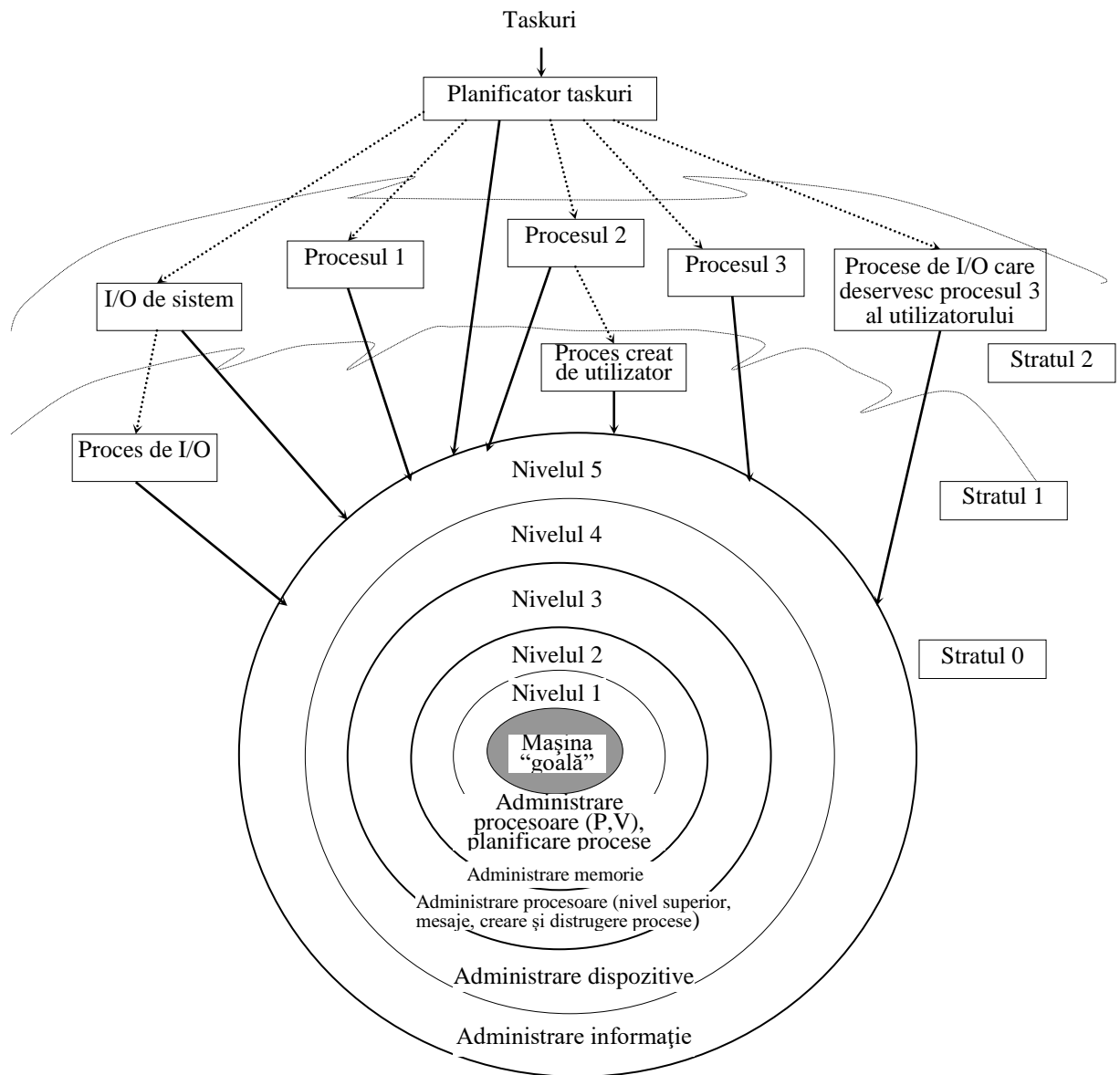


Fig. 1.10. Structura ierarhică a sistemului de operare

La nivelul 1 (nivelul cel mai inferior) sunt situate funcțiile de care au nevoie toate componentele administrării resurselor. Una dintre acestea este funcția care urmărește repartizarea resurselor, funcție, care la rândul său, solicită anumite mijloace de sincronizare. Aceste operații elementare sunt numite *P-operator* (ocuparea resursei sau cererea de acces) și *V-operator* (eliberarea resursei). Sincronizarea se face printr-o tehnică de programare, numită *semafor*. Fiecărei resurse îi este atașat un semafor. Atunci când în sistem apare o cerere pentru o resursă oarecare, pentru testarea semaforului respectiv este utilizat *P-operatorul*; dacă semaforul este “pe verde” (resursa este liberă), *P-operatorul* îl trece “pe roșu” (îl închide) și returnează controlul. În caz contrar, procesul care a generat cererea este trecut în stare de așteptare, pentru ca mai târziu, atunci când *V-operatorul* va elibera resursa și va trece semaforul “pe verde”, să acceseze resursa.

Amplasarea funcțiilor elementare pe nivele poate fi făcută în felul următor:

Nivelul 1. Administrarea procesoarelor (nivelul inferior)

- P - operatorii de sincronizare
- V - operatorii de sincronizare

- planificarea proceselor (aparatură de multiprogramare)

Nivelul 2. Administrarea memoriei

- alocarea memoriei
- eliberarea memoriei

Nivelul 3. Administrarea procesoarelor (nivelul superior)

- crearea și distrugerea unui proces
- transmiterea și recepționarea mesajelor între procese
- lansarea unui proces
- oprirea unui proces

Nivelul 4. Administrarea dispozitivelor

- urmărirea stărilor tuturor dispozitivelor periferice
- planificarea intrărilor/ieșirilor
- inițierea operațiilor de intrare/ieșire

Nivelul 5. Administrarea informației

- crearea și distrugerea unui fișier
- deschiderea și închiderea unui fișier
- citirea și înscrisarea unui fișier.

Nucleul sistemului de operare este format de subprogramele, care asistă execuția proceselor. Pentru a decide care funcții pot fi realizate în formă de procese separate este necesar să se stabilească funcțiile care pot fi executate independent și în mod paralel cu procesele utilizatorului (nu se va câștiga nimic evidențiind în procese separate funcții care trebuie să fie îndeplinite secvențial). Sistemele avansate permit crearea oricărui număr de procese, ceea ce este foarte comod pentru organizarea calculelor paralele sau a regimurilor de timp real. În cadrul tehnologiilor noi noțiunea de proces a fost substanțial modificată, introducându-se alte concepte (fire, thread en.), care exploatează într-un mod mai eficient ideile multitasking-ului și multiprogramării.

1.2.3. Alte puncte de vedere

În compartimentele precedente au fost tratate sistemele de operare din diferite puncte de vedere, cum ar fi SO și procesele, SO și mașina extinsă sau SO și mașina ierarhică. Există și alte puncte de vedere asupra sistemelor de operare pe care un specialist ar trebui să le cunoască.

Pentru un utilizator obișnuit, convins că un calculator este doar un instrument care îl ajută în rezolvarea unor probleme din domeniul său de activitate, noțiuni cum ar fi administrarea memoriei cu paginare sau driverele dispozitivelor nu semnifică prea multe. Destinația principală a unui sistem de operare pentru această categorie de utilizatori este punerea la dispoziție a unui set de programe care ar ajuta în formularea și rezolvarea problemelor concrete. Abordarea sistemelor de operare din acest punct de vedere (abordare funcțională) poate conduce la confundarea lor cu unele programe, utile și foarte importante cum sunt traductoare, bibliotecile, mediile integrate, etc. Pentru a evita posibilitatea apariției unei astfel de probleme aceste programe, de obicei, nu sunt considerate componente ale sistemului de operare.

Interfața sistemului de operare cu utilizatorul prezintă un interes aparte. Progresul în acest domeniu este spectaculos, dacă vom lua în considerație că în primele sisteme de operare utilizatorul era obligat să indice în mod explicit și manual fiecare pas, oricât de ne semnificativ ar fi părut. Formularea pașilor cu ajutorul limbajului de control al lucrărilor (Job Control Language, JCL) nu a schimbat substanțial situația. Acest limbaj, nefiind agreat de utilizatorii simpli, care l-au denumit *limbaj pășăresc*, așa și nu a fost acceptat de către aceștia.

Conform JCL utilizatorul trebuie să încorporeze programul propriu într-un set de instrucțiuni, care indicau începutul, sfârșitul programului și al datelor de intrare, pașii și conținutul concret al pașilor. JCL în principiu era un metalimbaj de programare (programare la nivel macro). Pentru mijloacele tehnice de la acea perioadă JCL a sporit substanțial eficiența sistemelor de calcul, deși au existat multe inconveniente, principalul fiind lipsa posibilității lucrului interactiv.

Microprocesoarele și memoriile anilor 1970 au pus problema lansării pe piață a calculatoarelor personale (PC) cu toate consecințele asociate. Una din consecințe este și interfața utilizator-calculator, sistemul de operare devenind până la urmă responsabil de aceasta. Interfața grafică a utilizatorului (Graphical User Interface - GUI) a apărut mai întâi ca un complement al sistemului de operare (pentru MS DOS - Windows 1, Windows 2 sau chiar Windows 3, de exemplu), pentru ca mai apoi să fie integrată în cadrul sistemului (Windows 95, Windows NT, etc.). Un sistem de operare nu este, în principiu, obligat să posedă o interfață sofisticată, totul este determinat de baza tehnică utilizată și

de necesitățile concrete. Oricum, un specialist trebuie să distingă aceste două noțiuni – sistemul de operare și interfața utilizatorului.

1.3. Evoluția sistemelor de operare

O analiză cronologică a dezvoltării sistemelor de operare este greu de realizat, deoarece multe din principiile foarte importante au fost realizate pentru prima dată cu mult înainte de a deveni unanim acceptate. De exemplu, conceptele de memorie paginată și memorie virtuală au fost realizate pentru prima dată în 1959 în cadrul sistemului “Atlas” [5], fiind utilizate la mijlocul anilor 1960 în unele sisteme cu destinație specială, pentru ca în 1972 să fie preluate de firma IBM în cadrul familiei de calculatoare mari.

Primele sisteme erau caracterizate prin **prelucrarea secvențială** a sarcinilor. Timpul de execuție a programelor era relativ mare, instrumentele de depanare – primitive, fiecare programator își încărca în mod individual programul (pachetul de cartele perforate), apăsa butoane, controla conținutul locațiunilor de memorie, etc. (1950 – 1956).

Sporirea vitezei de calcul, dar și a prețului calculatoarelor cerea o utilizare mai eficientă a timpului de calculator. Nu putea fi tolerată situația ca un calculator să “nu facă nimic”, atunci când utilizatorul își încarcă în mod manual programul. Au fost propuse programe de monitorizare (monitoare), care treceau de la o lucrare la alta în mod automat, utilizatorul fiind responsabil de organizarea corectă a programelor în cadrul unui pachet – primele încercări de **prelucrare pe loturi** (1956 – 1959).

Odată cu creșterea complexității calculatoarelor, îndeosebi în ceea ce consta administrarea dispozitivelor periferice, au fost propuse **sisteme** supervizoare (**executive**), care se aflau în memoria calculatorului în mod constant și acordau utilizatorilor servicii în gestiunea operațiilor de intrare/ieșire (1959 – 1963). În aceste sisteme de operare erau realizate și o serie de facilități noi, cum ar fi controlul unor posibile încercări din partea programului de a încălca restricțiile existente în sistem, culegerea informațiilor de evidență, etc.

Au urmat apoi sistemele cu **multiprogramare** menite la început să rezolve problema armonizării vitezei de calcul a unității centrale și a perifericelor. Drept consecință, au apărut o mulțime de limbaje de control a lucrărilor, a fost realizată o standardizare substanțială a operațiilor de intrare-ieșire.

După 1965 au apărut primele sisteme cu **partajare a timpului** (time sharing), au fost propuse sisteme sofisticate de administrare a informației (sisteme de gestiune a datelor sau sisteme de fișiere, File Systems). Principiul **time sharing** oferea posibilitatea lucrului interactiv a mai multor utilizatori pe un singur calculator, fiecărui utilizator în mod ciclic acordându-i-se un interval anume de timp (cuantă de timp) și, datorită vitezei mari de calcul a unității centrale, creându-i-se impresia posesiei tuturor resurselor calculatorului.

Memoria virtuală și **mașinile virtuale** sunt niște principii care nici până astăzi nu au fost exploatate până la capăt. Progresele ultimilor ani în domeniul resurselor tehnice au permis implementarea acestor principii nu numai în cadrul sistemelor de calcul mari, ci și pentru calculatoarele personale. Specificațiile sistemelor de operare au fost în mare măsură standardizate, diversitatea SO devine tot mai mică, mulți specialiști exprimându-și îngrijorarea de o posibilă monopolizare a domeniului într-un viitor apropiat. Evident, aceasta nu poate să sugereze nici într-un caz ideea că studierea principiilor de bază (mai vechi și mai noi) ale sistemelor de operare ar fi de prisos, ca și familiarizarea sau chiar cercetarea minuțioasă a unor sisteme existente, nicidecum nu poate însemna, în special pentru un specialist, pierderea interesului față de analiza și concepția sistemelor de operare.

O prezentare succintă a evoluției sistemelor de operare facilitează înțelegerea caracteristicilor actuale ale acestora și a termenilor deja introduși.

1.3.1. De la "poartă deschisă" la tratarea pe loturi

Primele calculatoare nu dispuneau de sisteme de operare. Fiecărui utilizator i se rezerva pentru un timp determinat calculatorul cu toate resursele acestuia. Interacțiunea era directă, programul și datele fiind introduse în mod manual sub formă de zerouri și unități. Utilitățile care au apărut aveau destinația de a asista elaborarea programelor (asamblatoare, compilatoare, etc.) sau de a facilita operațiile de intrare-ieșire.

Acest mod de exploatare, numit "poartă deschisă", era de o eficacitate minimă, dispozitive foarte costisitoare fiind utilizate ineficient. Din această cauză la sfârșitul anilor 1950 au apărut primele "**monitoare de înlănțuire**" - programe care permiteau executarea secvențială a unui set de lucrări, pregătite anticipat, trecerea de la o lucrare la alta fiind realizată în mod automat.

Funcția principală a unui atare sistem era gestiunea resurselor: memoria, procesorul, dispozitivele periferice. Automatismul acestei gestionări implică o funcție de protecție a setului de lucrări contra unor riscuri perturbatorii în caz de eroare:

- limitarea timpului de ocupare a procesorului pentru a evita blocarea sistemului atunci când un program conține o buclă infinită;

- administrarea corectă a intrărilor-ieșirilor pentru a evita blocajele în utilizarea perifericelor;
- protecția zonei de memorie rezervate monitorului pentru a împiedica modificarea accidentală a acestuia.

Deși utilizarea monitoarelor de înlănțuire a ameliorat notabil randamentul utilizării procesorului, acest randament rămânea foarte scăzut din cauza că procesorul nu era folosit în timpul operațiilor de intrare-ieșire. O soluționare ar fi constat în utilizarea a două calculatoare - unul (principal) pentru executarea programelor și altul (auxiliar) pentru operațiile de intrare-ieșire. O planificare adecvată a executării lucrărilor permitea utilizarea celor două calculatoare în paralel, dând posibilitatea sporirii la maximum a randamentului calculatorului principal.

1.3.2. Multiprogramarea

Progresul tehnologic și conceptual al anilor 1960 - 1970 a permis excluderea unor limitări caracteristice sistemelor de tratare pe loturi. Astfel, au fost introduse procesoare specializate pentru operațiile de intrare-ieșire (unitățile de schimb sau canalele), care permiteau eliberarea procesorului central de gestionarea dispozitivelor de intrare-ieșire. Introducerea principiului multiprogramării cu partajarea memoriei între mai mulți utilizatori au permis o utilizare și mai bună a procesorului central.

1.3.2.1. Organizarea intrărilor - ieșirilor în zone de memorie tampon

Un canal este un procesor specializat în executarea autonomă a operațiilor de intrare-ieșire, paralel cu procesul de prelucrare a informațiilor. Viteza de lucru a organelor periferice este relativ mică din cauza unor dispozitive mecanice care intră în componența acestora. Pentru excluderea influenței perifericelor asupra vitezei de lucru a sistemului de calcul s-a propus să se păstreze în memorie în anumite zone tampon datele de intrare (care vor fi utilizate la necesitate) și rezultatele (care vor fi imprimate în mod autonom) mai multor lucrări. Această situație este prezentată în fig.1.11.

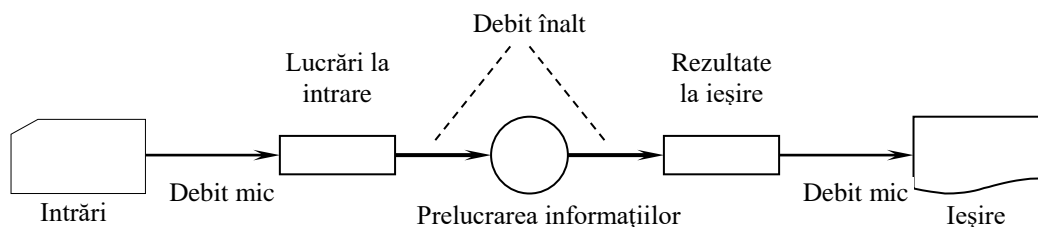


Fig.1.11. Buferizarea intrărilor-ieșirilor

Pentru a exclude utilizarea ineficientă a memoriei operative prin formarea zonelor-tampon, acestea erau organizate în memoria secundară de capacitate înaltă, transferurile între aceste două nivele ale memorie fiind de asemenea comandate de un canal.

Deși utilizarea memoriilor tampon prezintă o serie de avantaje privind sporirea randamentului dispozitivelor calculatorului, totuși două momente negative pot fi menționate:

- atunci când lucrarea în curs de execuție are nevoie de date unitatea centrală rămâne inactivă pe toată perioada citirii acestora;
- o lucrare de scurtă durată, sosită în timpul execuției unei lucrări "lungi", trebuie să aștepte terminarea acesteia din urmă.

1.3.2.2. Multiprogramarea

Ultimele două momente au condus la ideea utilizării unui mod de funcționare a SO în care:

- o lucrare aflată în starea eligibilă să poată folosi unitatea centrală, eliberată de o lucrare care trece în așteptarea datelor,
- unitatea centrală să poată modifica timpul său de alocare înainte de terminarea unei lucrări în scopul satisfacerii cerințelor legate de timpul de răspuns.

În acest caz este necesar ca valoarea timpului de comutare a unității centrale să fie mai mică în raport cu durata unui transfer între nivelele memoriei. Aceasta implică prezența simultană în memoria operativă a mai multor programe sau părți de programe. Acest mod de funcționare este numit **multiprogramare** [6] și este prezentat în figura 1.12.

Schema din figura 1.12 evidențiază două momente importante: rolul principal al memoriei operative (în SO anterioare acest rol era deținut de către unitatea centrală) și fluxul informațional între memoria operativă și cea secundară. Drept consecință, volumul memoriei operative și viteza de transfer între cele două nivele ale memoriei devin caracteristici determinante ale performanței sistemului.

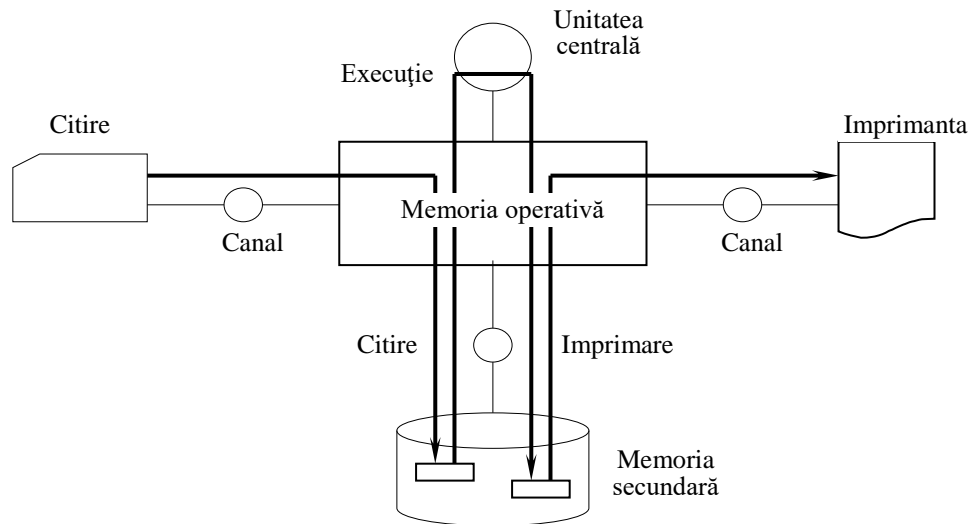


Fig. 1.12. Fluxul informațional într-un sistem cu multiprogramare

Principalele avantaje și restricții ale multiprogramării pot fi prezentate în rezumat după cum urmează:

- un sistem cu multiprogramare este mai complicat de conceput și implementat pentru că el trebuie să asigure partajarea memoriei și protecția reciprocă a programelor;
- multiprogramarea solicită dispozitive speciale pentru relansarea programelor și protecția memoriei;
- un sistem cu multiprogramare asigură o utilizare mai uniformă a resurselor: unitatea centrală, memoria, organele de intrare-ieșire;
- multiprogramarea permite reducerea timpului de răspuns în cazul lucrărilor de durată mică într-un sistem cu prelucrare secvențială.

1.3.2.3. Partajatarea timpului

Un sistem de operare cu partajarea timpului trebuie să garanteze fiecărui utilizator un timp acceptabil de răspuns. Acest rezultat este obținut prin alocarea succesivă a procesorului pentru tranșe de timp (*cuante*) relativ mici programelor utilizatorilor. Viabilitatea unei asemenea tehnici este legată de caracteristicile lucrului interactiv. Activitatea unui utilizator conține două componente: *timpul de reflecție (gândire)*, în care utilizatorul elaborează, propune subiecte de lucru, introducând în calculator informații și *timpul de așteptare*, când așteaptă prestarea serviciului solicitat. Prima componentă este de o durată medie mult mai mare decât a doua și sistemul poate să servească simultan mai mulți utilizatori, folosind timpul mort, datorat perioadei de reflecție.

Ca exemplu, fie un sistem cu partajarea timpului, care deservește 1000 utilizatori cu un comportament mediu identic. Admitem că durata timpului de gândire este în medie de 9 ori mai mare decât a timpului de așteptare, acesta din urmă reprezentând 10% din timpul total. Avem în medie 100 utilizatori activi (care se află în așteptarea unui răspuns la un serviciu cerut). Presupunem că valoarea cuantei este de 5 ms. Dacă executarea unei cereri durează o cantă, timpul de răspuns va fi de ordinul unei jumătăți de secundă.

Am admis în acest exemplu că toate programele utilizatorilor activi se află în memoria principală: timpul de comutare între programele a doi utilizatori se reduce la timpul de realocare a unității centrale, care poate fi neglijat în comparație cu valoarea cuantei. Multiprogramarea devine obligatorie, dacă ținem cont de raportul dintre acest timp și timpul necesar pentru încărcarea unui program în memoria secundară. Situația este de fapt și mai complicată: volumul memoriei principale nu permite aflarea aici a tuturor programelor utilizatorilor activi. Este posibil ca o informație să fie lipsă în memoria operativă atunci când unitatea centrală are nevoie de ea. Rolul sistemului de operare este de a minimiza pe cât este posibil probabilitatea apariției unui atare eveniment.

Dezvoltarea sistemelor de operare cu partajarea timpului a pus în evidență importanța interacțiunii om-calculator. Acest aspect a fost mult timp neglijat, dar cei care au procedat altfel au avut doar de câștigat.

Apariția microprocesoarelor cu posibilități mari de calcul și prețuri tot mai mici a permis pe de o parte utilizarea la scară largă a calculatoarelor personale, iar pe de altă parte, satisfacerea cerințelor utilizatorilor sistemelor informatice de partajare a resurselor fizice și logice distribuite geografic. Au apărut noi tipuri de sisteme informatice: calculatoarele personale, rețelele locale și rețelele globale.

Conceptele și tehnicile elaborate pentru sistemele centralizate au rămas valabile și în cazul serverelor sau sistemelor de operare pentru calculatoarele personale. Repartizarea distribuită a informației, precum și transmiterea

ei la distanțe mai mari sau mai mici a ridicat probleme noi, cum ar fi coordonarea activităților la distanță sau menținerea coerenței informațiilor distribuite. Toate acestea au trebuit să fie luate în considerație la elaborarea sistemelor de operare de rețea.

1.3.3. Windows, Unix și alte sisteme

Paralel cu evoluția tehnică și funcțională a sistemelor de operare a avut loc și o importantă evoluție conceptuală, care a permis o mai bună înțelegere a funcționării SO și a condus la elaborarea unor metode proprii de concepere.

Anul 1964 poate fi considerat de debut pentru cercetările științifice în domeniul sistemelor de operare, cercetări care au fost generate de rezultatele obținute în sfera tehnică - primele sisteme cu partajare a timpului (Thor, CTSS), anunțarea seriei IBM 360 și introducerea canalelor de intrare-ieșire, prima memorie paginată (Atlas), etc.

În perioada 1965-1968 au fost fundamentate concepte teoretice importante, necesare pentru conștientizarea gestionării activităților fizice sau logice paralele: proces secvențial, excluderea mutuală, sincronizarea, semaforul. Acestea au fost aplicate cu succes la elaborarea sistemelor de operare: sistemul THE (1967) utilizează semafoarele, sistemul Multics (1964-1970) conține un nucleu de gestionare a proceselor. Metode și utilitare destinate sistemelor distribuite sunt propuse la sfârșitul anilor 1970. Utilitățile de sincronizare încep să fie introduse în limbajele de programare. Sunt propuse primele metode de specificare și control al validității sincronizărilor.

Problema definirii informației a fost pusă inițial în seama limbajelor de programare: fiecare limbaj definea universul său de obiecte accesibile și mijloacele efective de accesare. Sistemul de operare trebuia doar să implementeze obiectele definite de limbajul de programare în memoria fizică, direct adresabilă.

Restricțiile de capacitate și preț a memoriei principale a condus foarte repede la modificarea modului de utilizare a memoriei secundare și introducerea unor mecanisme de schimb de informații între aceste două nivele ale memoriei. Prima memorie paginată apare în 1962 (Atlas), tot atunci compilatoarele limbajului Algol 60 folosesc paginația virtuală pentru gestionarea programelor în curs de execuție. În 1965 este fundamentată noțiunea de segmentare, deși încă în 1959 sistemul Burroughs B5000 utiliza un mecanism de adresare logică prin unități de volum variabil - segmente. În 1966 este propus un model de arhitectură pentru calculatoarele cu partajare a timpului, implementat în IBM 360/370. În acest calculator segmentele sunt doar simulate printr-o dublă paginație, din care cauză noțiunea de segmentare va rămâne pentru un timp puțin înțeleasă de comunitatea informatică. Problemele partajării și protecției informației, alocării resurselor conduc la noțiunea de memorie virtuală, care permite izolarea mecanismelor de gestionare a ansamblului “memorie principală - memorie secundară”, lăsând utilizatorilor simpli impresia unui spațiu de adrese uniform (contiguu). Tot atunci au fost introduse noțiunile de modularitate și structurare ierarhică, obiect, etc.

Astăzi sistemele de operare în sensul tradițional de concepere sunt mai mult obiectul unor elaborări industriale, decât de cercetare. Aceasta se datorează atât existenței unei rezerve extraordinare de metode și algoritmi, cât și standardizării stricte a funcțiilor și interfețelor sistemelor de operare.

1.3.3.1. SO UNIX și sistemele deschise

Sistemul de operare UNIX, primul sistem “mobil” care asigură un mediu fiabil de dezvoltare și utilizare a softului de aplicație, este fundamentul practic de elaborare a sistemelor fizico-logice deschise. Implementarea largă a SO UNIX a permis trecerea de la declararea conceptului sistemelor deschise la dezvoltarea în practică a acestui concept. Este greu de supraestimat și influența activităților de standardizare a interfețelor SO UNIX asupra dezvoltării domeniului sistemelor deschise. Cu toate acestea, pot fi evidențiate câteva versiuni ale SO UNIX, care diferă atât prin realizare, cât și prin interfețe și semantică, chiar dacă, odată cu dezvoltarea procesului de standardizare, aceste diferențe devin tot mai nesemnificative.

Modulele sursă ale SO UNIX au fost scrise de către colaboratorii companiei AT&T și timp îndelungat drepturile de autor, ca și drepturile de licențiere, au aparținut acestei companii. Mai târziu, din mai multe motive (complexitate tehnică în elaborarea și mentenanța acestui produs program complicat, probleme juridice, etc.), compania AT&T a creat entitatea USL (UNIX System Laboratories), responsabilă exclusiv de dezvoltarea și susținerea SO UNIX.

Compania USL a propus versiunea UNIX System V 4.0, care a devenit standardul de-facto și baza mai multor versiuni UNIX, create de producătorii de stații de lucru și servere. În ultimul succes al USL în calitate de filială a firmei AT&T - versiunea SVR 4.2 - pentru prima oară în istoria SO UNIX a fost realizat mecanismul proceselor “lejere” (fire, en. thread), care erau executate în baza unei memorii virtuale comune și permitea exploatarea conceptului “arhitectură multiprocesorală simetrică” în care mai multe procesoare au drepturi egale de accesare a memoriei operative comune.

În anul 1993 USL a fost absorbită de compania Novell, devenind astăzi departament al acesteia, marca înregistrată UNIX fiind cedată consorțului X/Open. La începutul anului 1995 compania Novell a anunțat o variantă nouă a SO UixWare 2.0 bazată pe System V 4.2. Era un SO cu un sistem de fișiere foarte fiabil, fiind admis

accesul la fișierele păstrate pe serverele NetWare, administratorul avea la dispoziție o interfață grafică bine pusă la punct, etc.

Pe parcursul a mai multor ani sistemul de operare de bază al companiei Sun a fost UNIX BSD. Însă, începând cu SunOS 4.0, s-a trecut la System V 4.0, firma Sun introducând o serie de modificări și extensii în această versiune. Ca exemplu, Sun a implementat paralelizarea programelor pentru sistemele multiprocesorale simetrice. Solaris este o interfață a lui SunOS cu mijloace suplimentare GUI și resurse de nivel înalt pentru organizarea lucrului în rețea (de exemplu, apelarea procedurilor la distanță – RPC). Și la baza SO HP/UX, DG/UX și AIX se află SVR 4.x din care cauză setul de bază al funcțiilor de sistem și bibliotecă este același. Variantele SO UNIX, propuse de compania SCO și destinate exclusiv platformelor Intel, sunt bazate pe modulele inițiale ale System V 3.2, fiind compatibile cu toate standardele de bază

Open Software Foundation (OSF) a fost prima companie comercială, care a încercat elaborarea SO UNIX în baza micronucleului Mach. A fost creat sistemul de operare OSF-1, care nu era „curat” (în sens de licențiere), deoarece folosea o parte a modulelor inițiale din SVR 4.0.

Variantele SO UNIX, propuse de Universitatea din California, sunt o alternativă reală pentru UNIX AT&T. De exemplu, UNIX BSD 4.2 era pus la dispoziție în module sursă și folosită chiar în fosta URSS pe calculatoarele de tip CM-3, CM-4. Grupul BSD a influențat enorm dezvoltarea sistemelor de operare UNIX, printre realizări amintim mult controversatul SO UNIX BSD 4.4, construit în baza principiilor „micronucleare”, sistemul FreeBSD și altele.

Nu putem să nu amintim aici și de realizarea originală a SO UNIX pentru platformele Intel, propusă de Torvald Linus – LINUX. Este un sistem de operare foarte popular în mediul studențesc, care are și una din cele mai populare teleconferințe în Internet.

1.3.3.2. Standarde UNIX

Odată cu ieșirea SO UNIX pe piață și creșterea substanțială nu numai a numărului de utilizatori, ci și a numărului de specialiști din toate colțurile lumii, care propun modificări și chiar variante proprii, a apărut necesitatea elaborării unor standarde, care ar asigura compatibilitatea.

Unul dintre primele standarde de-facto a fost cel publicat de USL pentru versiunea SO UNIX System V Release 4 - System V Interface Definition (SVID). Majoritatea variantelor comerciale respectau standardul SVID. Evident, SVID fiind un document de firmă, publicat fără discuții publice nu putea fi adoptat ca și standard oficial.

Paralel exista direcția BSD (Berkeley Standard Distribution), susținută de comunitatea universitară. Deși majoritatea realizărilor comerciale se bazau pe Sistem V, UNIX BSD era foarte popular în universități din care cauză a fost elaborat un standard, care a unit practic AT&T cu BSD. Acest lucru a fost început de asociația programatorilor profesioniști din cadrul UniForum (Sistemele Deschise) și continuat de grupurile de lucru POSIX (Portable Operating System Interface). Cel mai popular standard, adoptat de ISO la recomandarea IEEE, POSIX 1003.1 definește cerințele minime pentru componentele unui sistem de operare.

Organizația internațională X/Open, care activează în domeniul elaborării și propagării ideilor sistemelor deschise, culege și sistematizează standardele de-jure și de-facto de importanță industrială în așa numitul X/Open Common Application Environment (CAE). Specificațiile interfețelor componentelor, care formează CAE, sunt publicate în X/Open Portability Guide (XPG).

Pentru lumea UNIX este foarte important și standardul limbajului de programare C, adoptat mai întâi de ANSI și apoi de ISO. În acest standard sunt specificate, în afara limbajului C, bibliotecile obligatorii într-o realizare standard. Deoarece chiar de la apariție limbajul C, sistemele de programare erau strâns legate de UNIX, componentele bibliotecilor standard corespundeau exact mediului standard al SO UNIX.

Mai menționăm standardul de-facto SPARC Compliance Definition, propus de organizația SPARC International, propunerea organizației 88/Open pentru procesoarele RISC Motorola, standardul sistemului de ferestre, susținut de X Consortium (Institutul de Tehnologie din Massachusetts) și OSF/Motif, elaborat de Open Software Foundation.

1.3.3.3. Sisteme de operare cu micronucleu

Micronucleul este partea minimă principală a unui sistem de operare, folosită pentru asigurarea modularității și transportabilității. Noțiunea de micronucleu a fost introdusă de compania Next prin sistemul de operare cu micronucleul Mach. Nucleul acestui sistem de operare, de dimensiuni mici, în jurul căruia se situau subsistemele executate în regim user, trebuia să asigure o flexibilitate și modularitate foarte înaltă. Dar în realitate acestea au fost umbrite de prezența serverului monolit, care realiza sistemul de operare UNIX BSD 4.3, ales de compania Next în calitate de nivel superior pentru micronucleul Mach. Totuși, utilizarea micronucleului Mach a permis introducerea administrării mesajelor și a unei serii de funcții de serviciu orientate pe obiecte, în baza cărora a fost creată o interfață grafică elegantă a utilizatorului cu mijloace simple de configurare, administrare și dezvoltare program.

Următorul SO cu micronucleu a fost MS Windows NT, în care momentul principal declarat era, în afara modularității, transportabilitatea. Acest sistem de operare poate fi utilizat în sistemele mono- și multiprocesor, bazate pe procesoarele Intel, Mips, și Alpha. Mai mult, deoarece NT trebuia să execute și programele scrise pentru DOS, Windows, OS/2 și SO, compatibile cu standardele Posix, compania Microsoft a folosit modularitatea abordării „micronucleare” pentru crearea unei structuri generalizate, care nu repetă sistemele de operare existente, fiecare SO fiind emulat printr-un modul separat sau printr-un subsistem.

Au aderat la tehnologia „micronucleară” și companiile Novell/USL, Open Software Foundation (OSF), IBM, Apple și altele. Unul din concurenții principali ai lui NT în domeniul SO cu micronucleu sunt Mach 3.0, creat în Universitatea Carnegie-Mellon, și Chorus 3.0 al companiei Chorus Systems.

Un exemplu practic

Scopul exercițiilor de mai jos [7] este de a pune în evidență influența evoluției istorice a sistemelor de operare asupra unor caracteristici ale acestora.

Vom considera un calculator periferia căruia este un dispozitiv de introducere a cartelor perforate (1 000 cartele pe minut) și o imprimantă (1 000 linii pe minut). O *lucrare medie* este definită astfel:

- citește 300 de cartele,
- utilizează procesorul 1 minut,
- tipărește 500 de linii.

Se presupune, că toate lucrările au caracteristici identice cu ale *lucrării medii*.

Definim două caracteristici de performanță a sistemului:

- debitul mediu D - numărul de lucrări executate într-o oră,
- randamentul unității centrale η - partea din timpul total de utilizare a unității centrale în care aceasta execută lucru util, altul decât gestionarea perifericelor.

Exercițiul 1. Presupunem că periferia este gestionată de către unitatea centrală. Să se calculeze D și η în următoarele ipoteze de funcționare:

- sistemul este exploatat în regim "poartă deschisă"; durata unei sesiuni de exploatare este de 15 minute,
- sistemul este exploatat utilizând un monitor cu înlănțuire secvențială a lucrărilor.

Exercițiul 2. Presupunem că periferia este gestionată de un calculator separat, care conține o bandă magnetică în care sunt introduse datele la intrare și o bandă magnetică pentru ieșire (tipărirea se va efectua de pe această bandă). Datele la intrare sunt introduse în calculatorul principal de pe banda magnetică, acesta producând datele la ieșire pe bandă. Timpul necesar citirii de pe și scrierii pe benzi nu se va lua în considerație. Timpul de transfer a benzilor de pe un calculator pe altul este de 5 minute în fiecare sens; se presupune că o bandă va grupa un lot de 50 de lucrări (v. fig. 1.13).

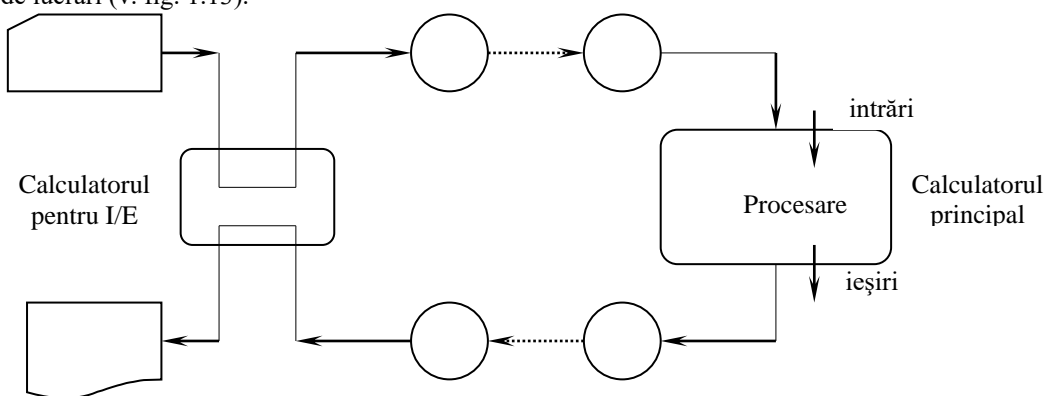


Fig. 1.13. Exemplificarea cazului 2.

Presupunem că intensitatea sosirii lucrărilor este suficientă pentru a ocupa tot timpul calculatorului central. Calculați D și η .

Stabiliți modul de planificare a seturilor de lucrări și calculați timpul mediu de așteptare a unui utilizator (timpul din momentul predării lucrării și până la recepționarea rezultatelor). Vom admite, că lucrările sosesc într-un ritm regulat, că timpul necesar formării unui program (pregătirea unui complet de cartele perforate corespunzătoare unui program) este 10 minute și timpul distribuirii rezultatelor unui lot (decuparea și trierea listelor utilizatorilor) la fel este 10 minute.

Exercițiul 3. Periferia este administrată de un canal de intrare-ieșire. Sistemul este monoprogram, iar monitorul de înlănțuire permite unității centrale să prelucreze o lucrare paralel cu citirea următoarei și tipărirea precedentei lucrări. Calculați în aceste condiții D și η . Aceeași întrebare dacă lucrarea medie presupune citirea a 1200 de cartele și imprimarea a 1500 linii într-un minut de utilizare a unității centrale.

Exercițiul 4. Operațiile de intrare-ieșire sunt gestionate cu ajutorul unei memorii-tampon (bufer) pe discul fix (spooling pentru citire și imprimare). Lucrarea medie este cea definită în punctul 3.

Presupunem că o cartelă conține 80 octeți, iar o linie de imprimare 120. Care este capacitatea minimă a memoriilor-tampon pentru citire și imprimare necesare pentru ca unitatea centrală să fie folosită la randamentul maxim? Care va fi în acest caz debitul lucrărilor?

Intensitatea sosirii lucrărilor și capacitatea memoriei-tampon pentru citire sunt cele din a), iar memorie-tampon pentru tipărire este de 2 Mo. Care este randamentul unității centrale?