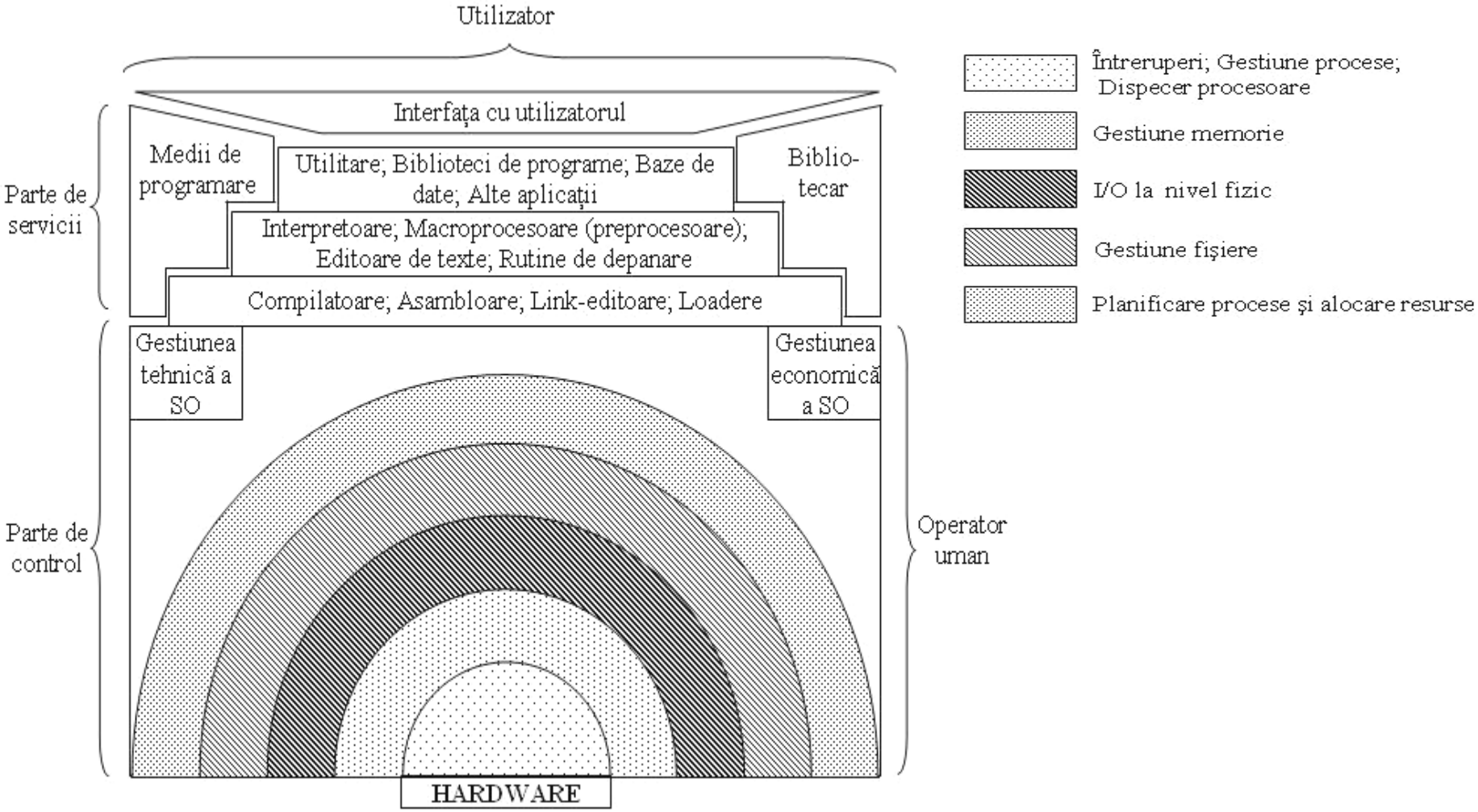
Sisteme de Operare

- Structura unui sistem de operare
- Organizarea memoriei
- Procese

Structura unui sistem de operare



Structura unui sistem de operare (2)

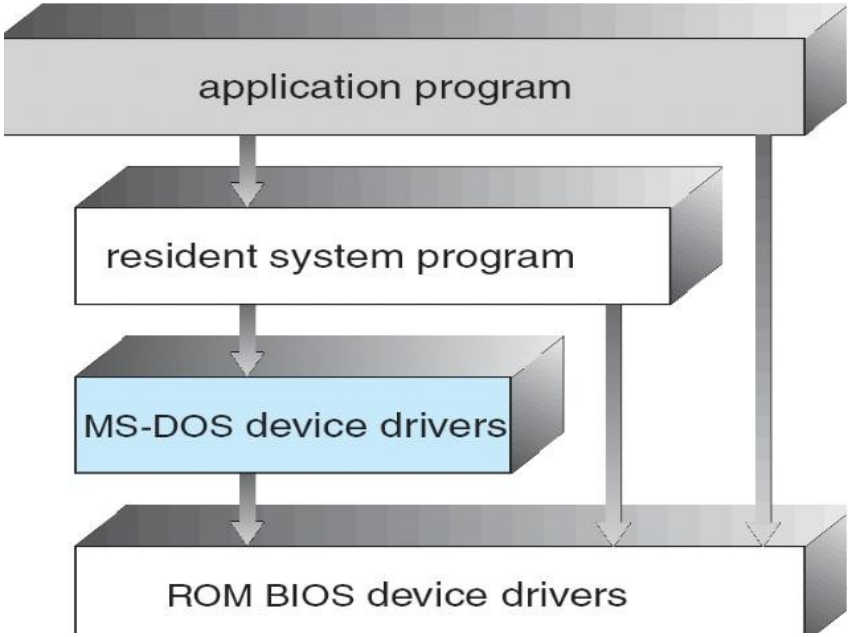
- partea de control:
  - se execută în mod nucleu şi realizează legătura cu sistemul de calcul;
  - întreruperi
  - gestiune procese
  - dispecer procesoare
  - gestiune memorie
  - I/O la nivel fizic
  - gestiune fişiere
  - planificare procese şi alocare resurse
  - gestiune tehnică a SO
  - gestiune economică a SO

Structura unui sistem de operare (3)

- partea de servicii:
  - se execută în mod utilizator, folosind facilităţile părţii de control şi asigură legătura cu utilizatorul;
  - compilatoarele
  - asamblorul
  - link-editorul sau editorul de legături
  - loader: încarcarea programelor
  - interpretor comenzi
  - macroprocesor (preprocesor)
  - editorul de texte
  - rutine de depanare
  - bibliotecarul
  - mediile de programare
  - interfaţa cu utilizatorul

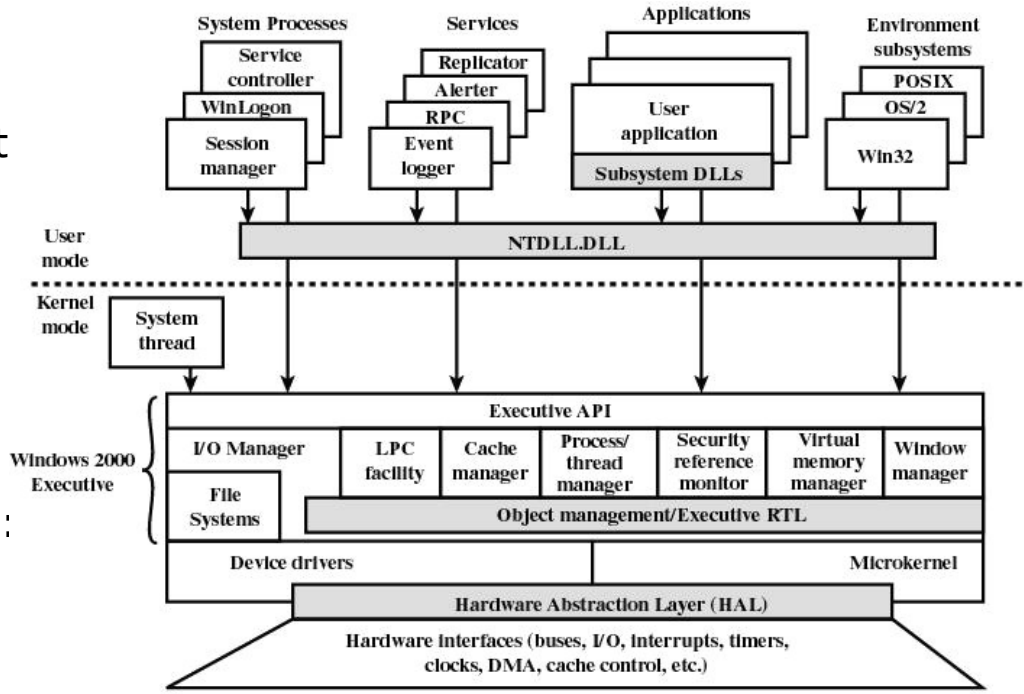
Exemple: MS-DOS

- Nu este modular
- Nivelurile nu sunt foarte bine separate

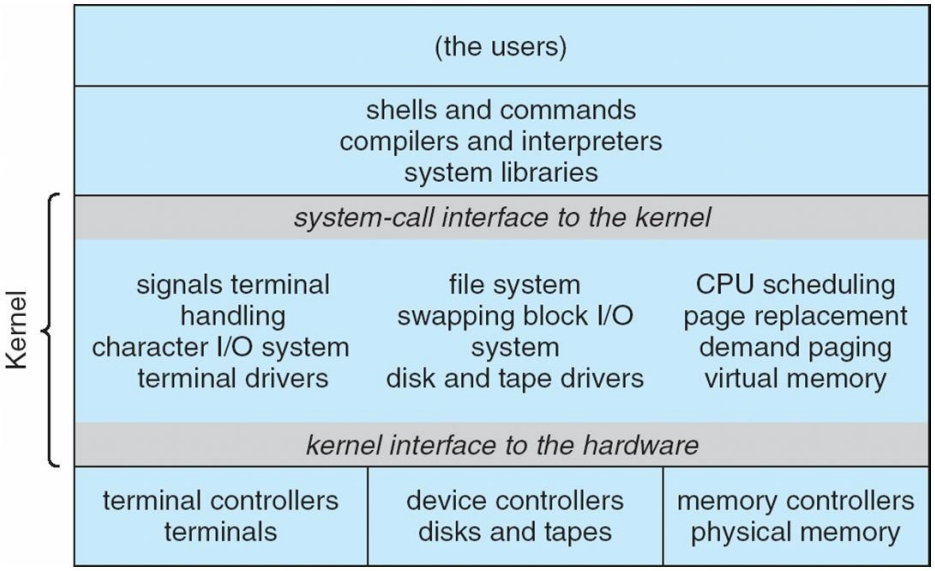


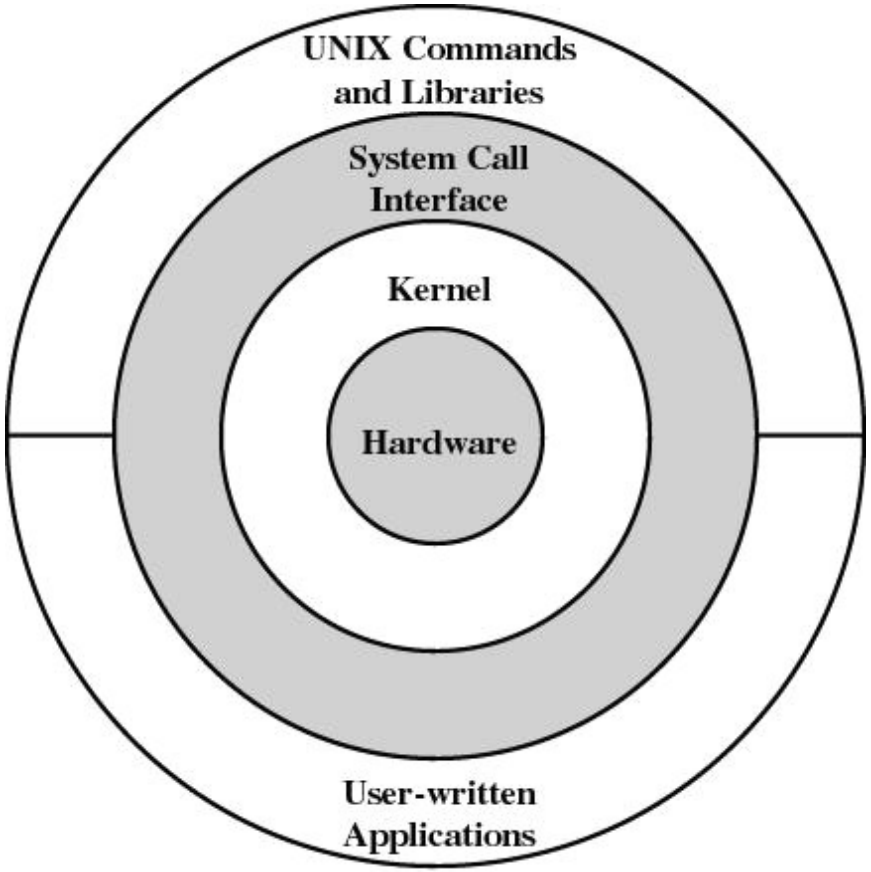
Exemplu: Windows 2000

- microkernel modificat:
  - nu are un microkernel pur;
  - multe din funcţiile sistemului sunt în afara microkernel-ului şi rulează în modul kernel;
  - modulele pot fi şterse, înnoite, înlocuite fără a fi necesară rescrierea întregului sistem
- Structura pe niveluri (layere):
  - Hardware abstraction layer (HAL): izolează sistemul de operare de specificul platformei hardware;
  - Microkernel
  - Drivere

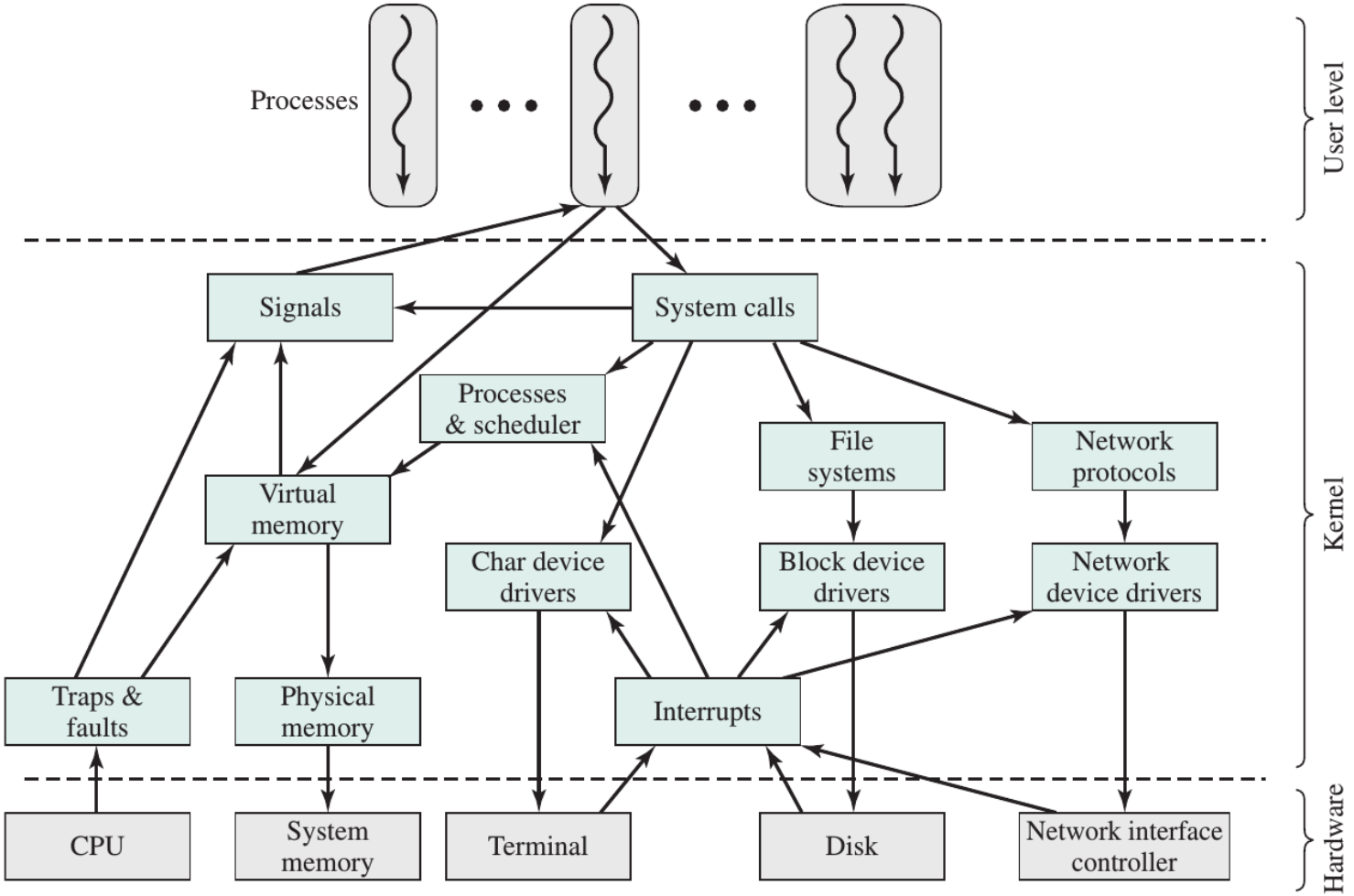


Exemplu: UNIX

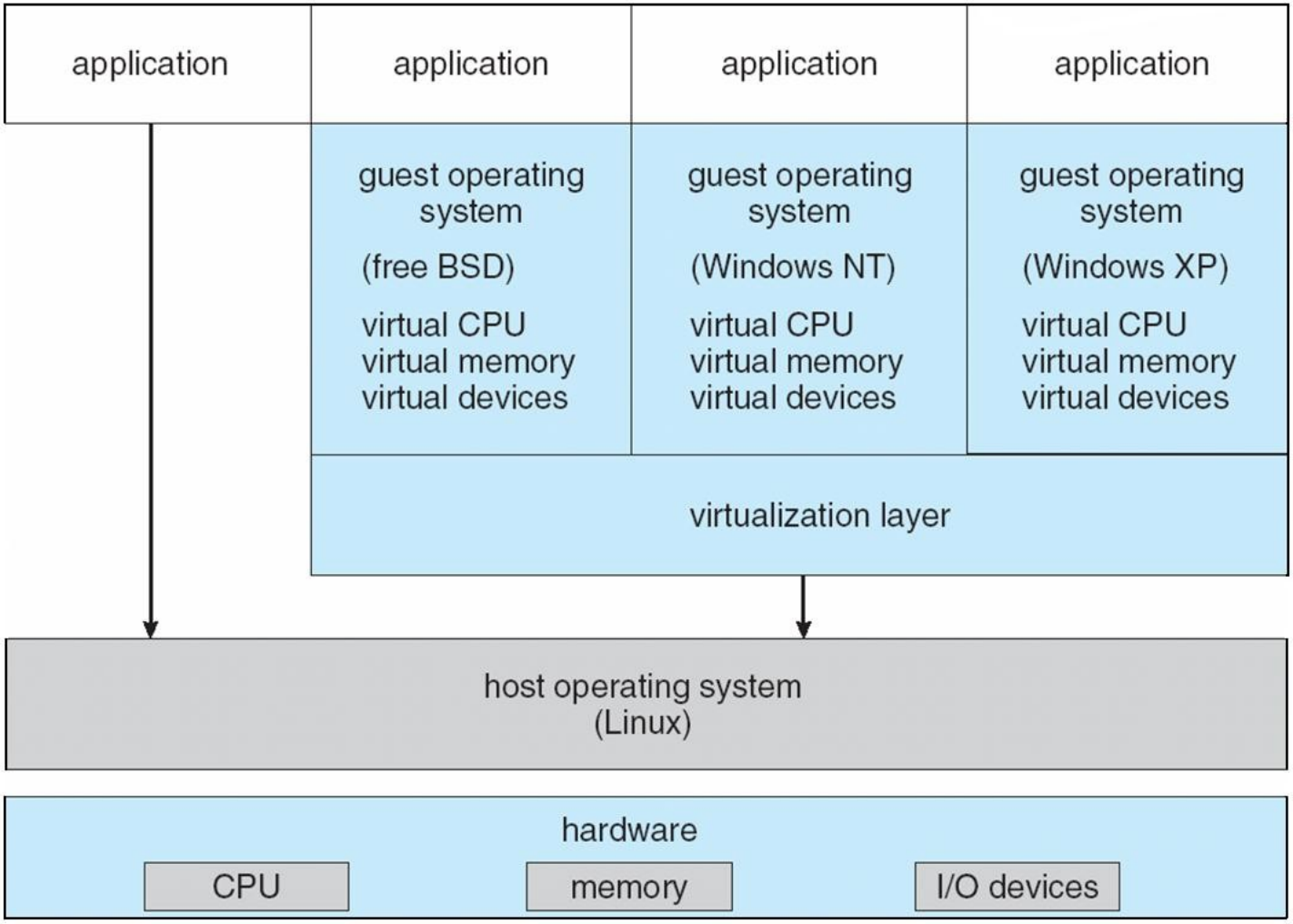




Exemplu: Linux



Exemplu: VMware



Nucleul (kernel) unui sistem de operare

- partea rezidentă a unui SO
- conţine proceduri care tratează:
  - planificarea proceselor
  - tratarea erorilor
  - verificarea securităţii
  - tratarea iniţială a apelurilor sistem
- ocupă o zonă fixă a memoriei
- include regiunea cu adresele cele mai mici, regiune în care se găsesc vectorii de întrerupere.

Nucleul (kernel) unui sistem de operare

- partea rezidentă a unui SO
- conţine proceduri care tratează:
  - planificarea proceselor
  - tratarea erorilor
  - verificarea securităţii
  - tratarea iniţială a apelurilor sistem
- ocupă o zonă fixă a memoriei
- include regiunea cu adresele cele mai mici, regiune în care se găsesc vectorii de întrerupere.

Nucleul (kernel) unui sistem de operare(2)

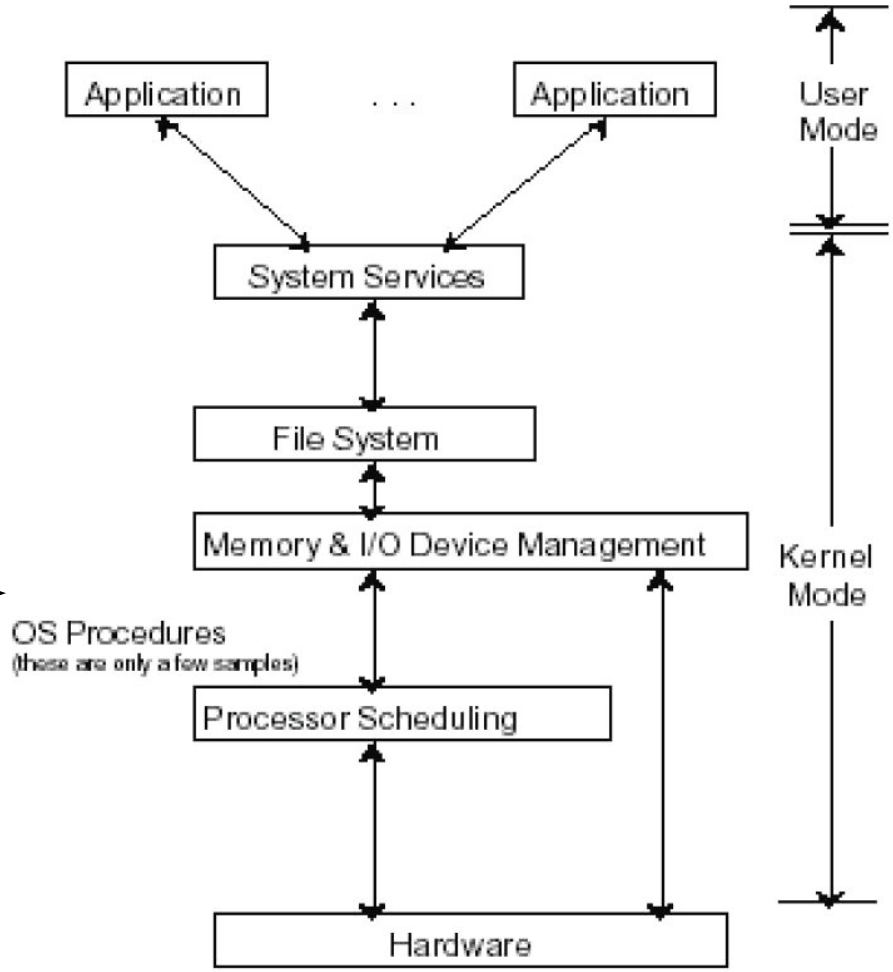
- Componentele mai puţin utilizate (componentele tranzitorii)
  - componentele pot avea o zonă de memorie rezervată
  - pot fi încărcate în orice zonă de memorie disponibilă
  - memory mapped I/O

memory mapped I/O

- o porţiune a spaţiului de adrese este rezervată pentru I/O şi sunt mapate peste un set de registre.
- aceste adrese fac referinţă la registrii din interfeţele hardware cu echipamentele periferice
- prin operaţiile de citire/scriere de la aceste adrese se transferă date spre şi de la echipamentele periferice.

Kernel – variante de implementare

- **nucleu monolit**:
  - implică implementarea funcţiilor legate de gestionarea proceselor, gestionarea fişierelor, I/O si a memoriei într-un singur modul care va conţine toate funcţiile aferente lor.
- **nucleu modular:**
  - asigură funcţionalitatea de baza la o clasă de module distincte (potrivit cu împărţirea funcţiilor), interacţiune cu ele facându-se prin apel de proceduri sau prin comunicaţie interproces.
- **nucleu extensibil:**
  - este o combinaţie între nucleul monolit şi cel modular pentru a realiza un schelet de nucleu (maşina abstractă) ce se poate îmbunătăţi conform cerinţelor funcţionale (are un numar minim de functii la care se pot adăuga altele).
- **nucleu multinivel:**
  - legat de conceptul de masina abstractă, constă în împărţirea funcţiilor în interfeţe apelabile intre ele.
- **micronucleu (microkernel):**
  - sistemul de operare este alcătuit din mai multe procese, fiecare asigurând anumite servicii.



Organizarea memoriei

- hărţile de memorie (**memory maps**)
  - conţinutul **spaţiului virtual de adrese** când este în execuţie un program utilizator;
  - conţinutul spaţiului virtual de adrese când este în execuţie o componentă a sistemului de operare;
  - conţinutul memoriei fizice;

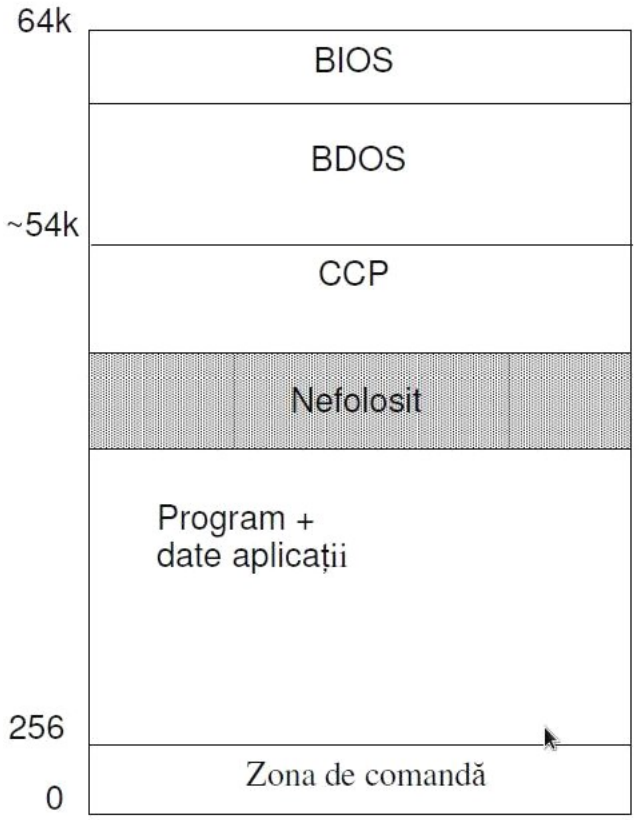
Spațiul virtual de adrese

- disproporţia dintre necesarul de memorie al unui program și memoria fizică (limitată) este rezolvată de conceptul de **spațiu virtual de adrese***
  - Spaţiul de adrese disponibil pentru procese este fix și declarat în SO pentru toate procesele
  - SO moderne permit utilizarea a 2^N bytes de memorie, unde N este 32 sau 64

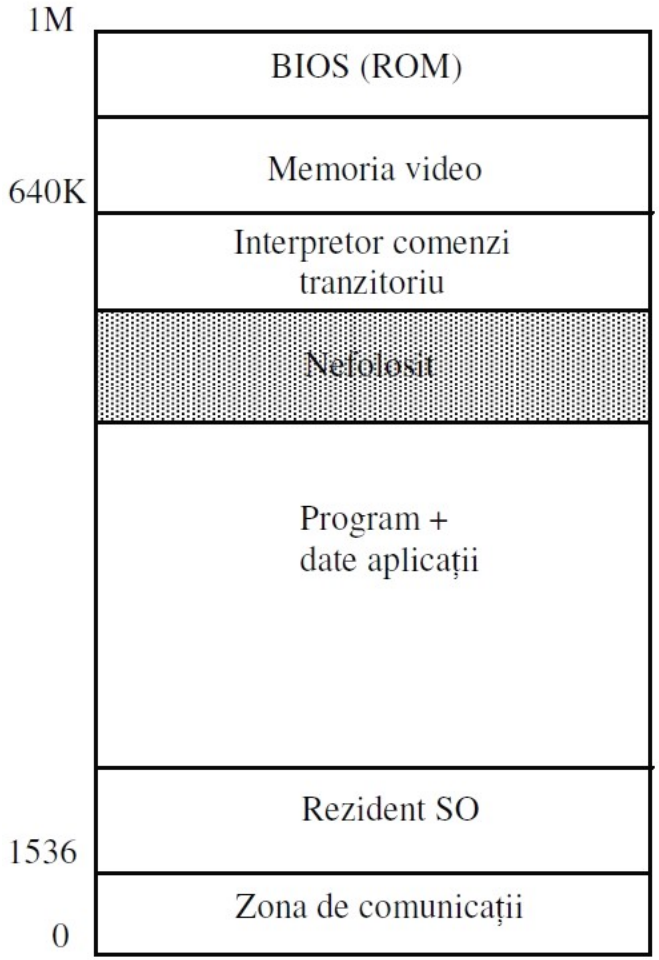
* în Cursul 1 – sisteme cu memorie virtuală

Organizarea memoriei CP/M

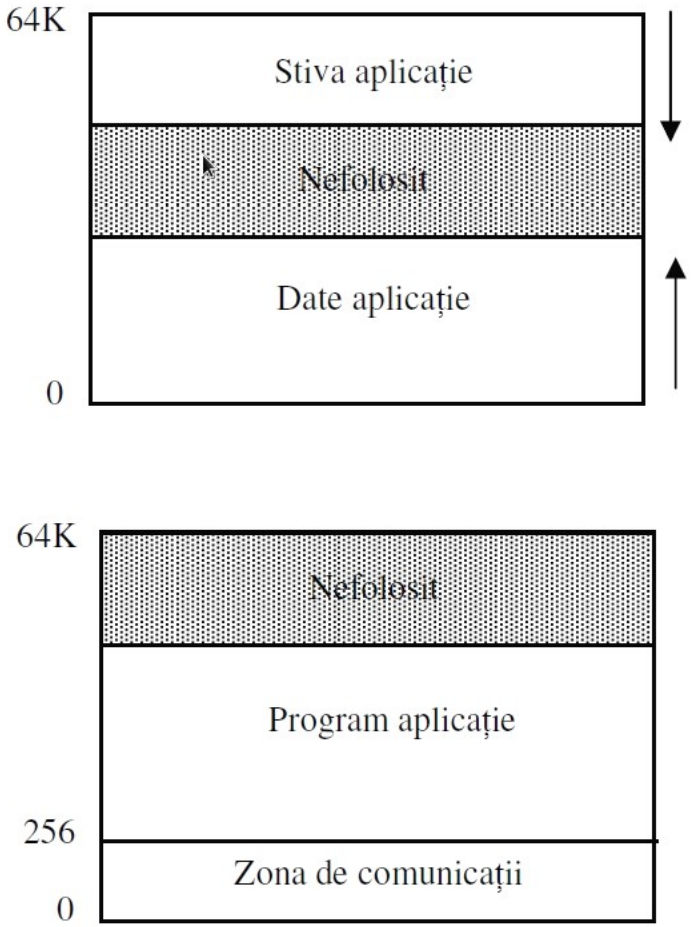
- Memoria direct adresabilă în sistem este de 64kocteţi
  - limită impusă de arhitectura procesorului 8080
- primii 256 octeţi
  - pentru vectorii de întrerupere şi pentru informaţii de sistem
- 10 kocteţi din zona superioară a spaţiului de adrese
  - Componentele rezidente BIOS (Basic Input / Output System) şi
  - BDOS (Basic Disk Operating System)
- 2 kocteţi
  - Interpretorul de comenzi CCP (Command Control Processor)



Organizarea memoriei MS-DOS



Harta memoriei la MS-DOS



Adrese virtuale într-un proces

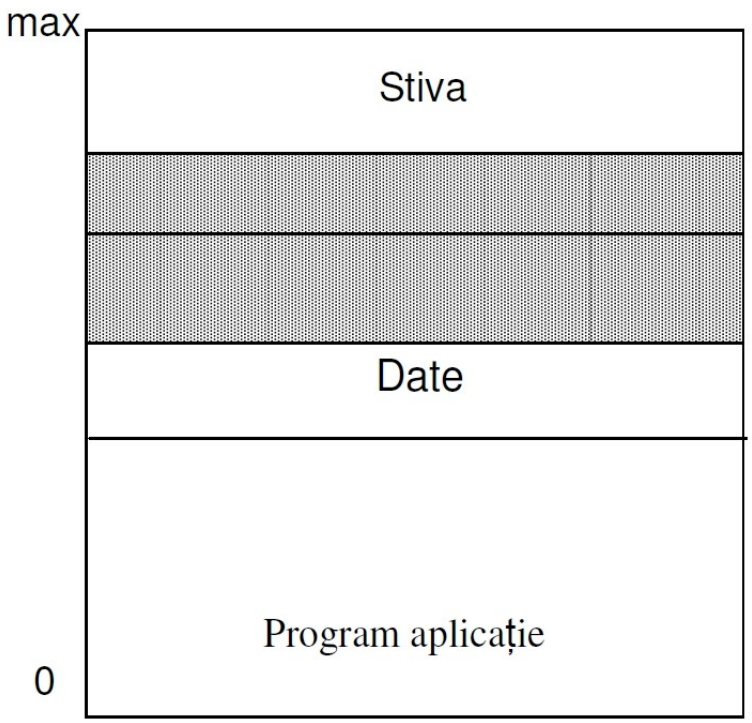
Organizarea memoriei MS-DOS

- Procesorul 8086 permite adresarea directă a 1Moctet de memorie
- rezervă zona peste 640K pentru memoria video şi pentru programe memorate în ROM (o serie de programe de test activate la pornirea calculatorului)
- vectorii de întrerupere şi o zonă tampon de 512 octeţi ocupă primii 1536 octeţi
- Partea rezidentă ocupă zona începând de la 1536 - include şi o porţiune din interpretorul de comenzi
- Procesele:
  - spaţiul de adrese este limitat la 4 segmente de 64Ko fiecare
  - un segment pentru cod
  - un segment pentru stivă
  - două segmente pentru date

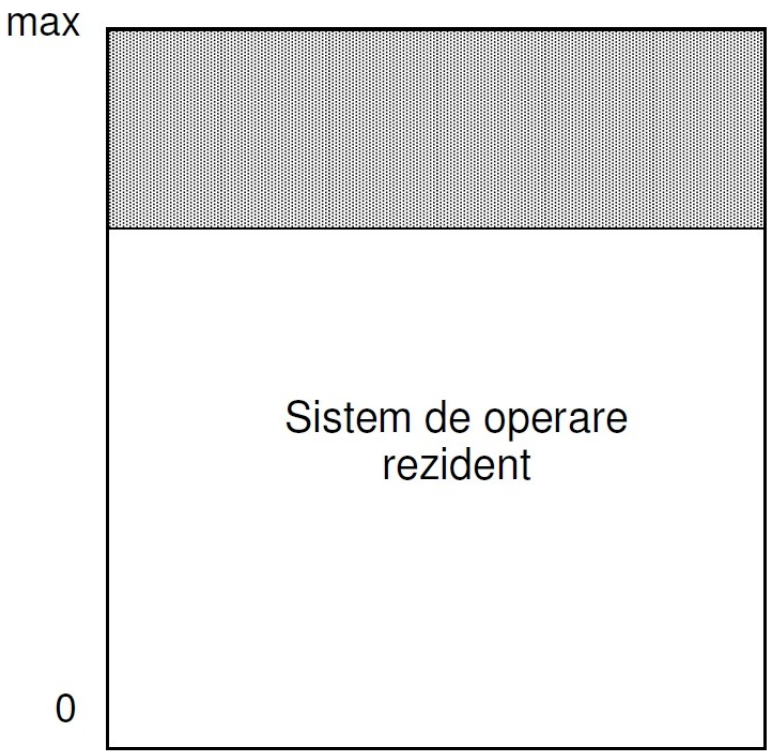
Organizarea memoriei UNIX

- Harta de memorie:
  - trebuie ţinut cont de caracteristicile arhitecturale ale calculatorului gazdă
  - din spaţiul de adrese al unui proces nu este vizibilă nici o porţiune a sistemului de operare.

Organizarea memoriei UNIX



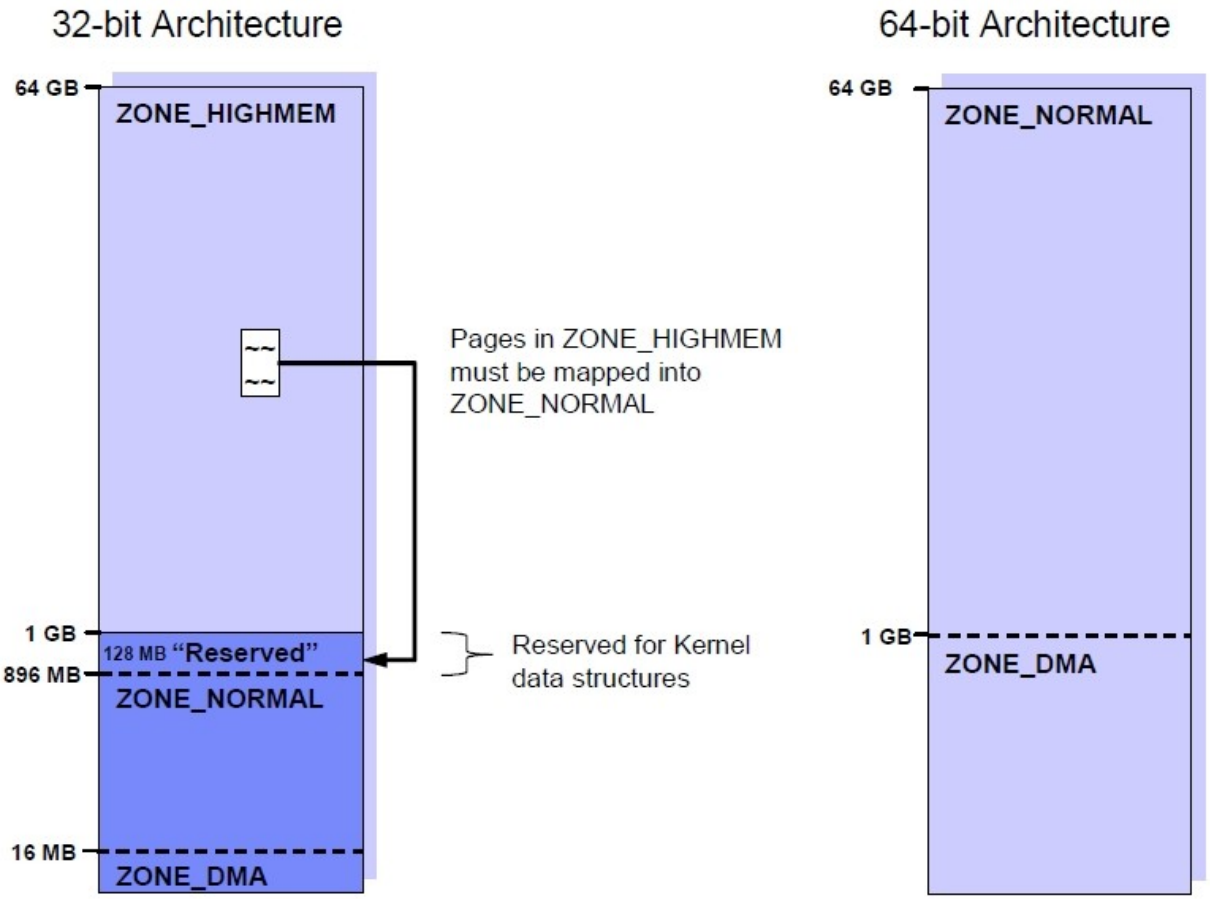
UNIX - Adrese virtuale într-un proces



UNIX - Adrese virtuale în sistemul de operare

Organizarea memoriei Linux

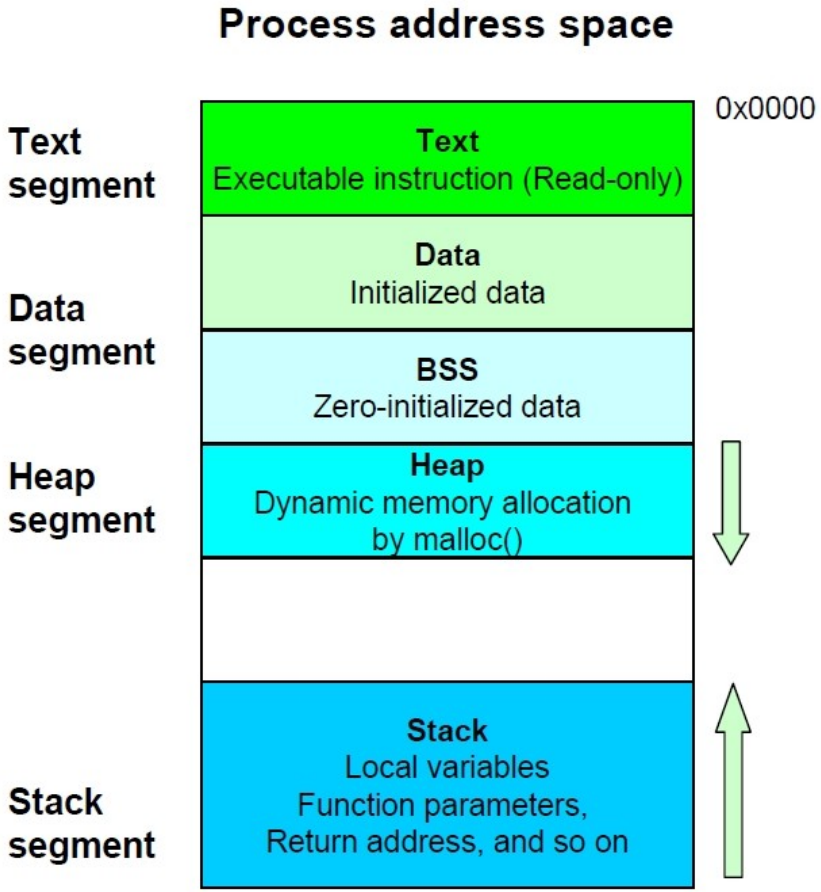
Sursa: Linux Performance and Tuning Guidelines - https://lenovopress.com/redp4285.pdf



- Procesoarele 386:
  - folosind memoria paginată, fiecare proces poate accesa 4GB de memorie (2^{32})

Organizarea memoriei Linux

- Memoria vazută de un proces

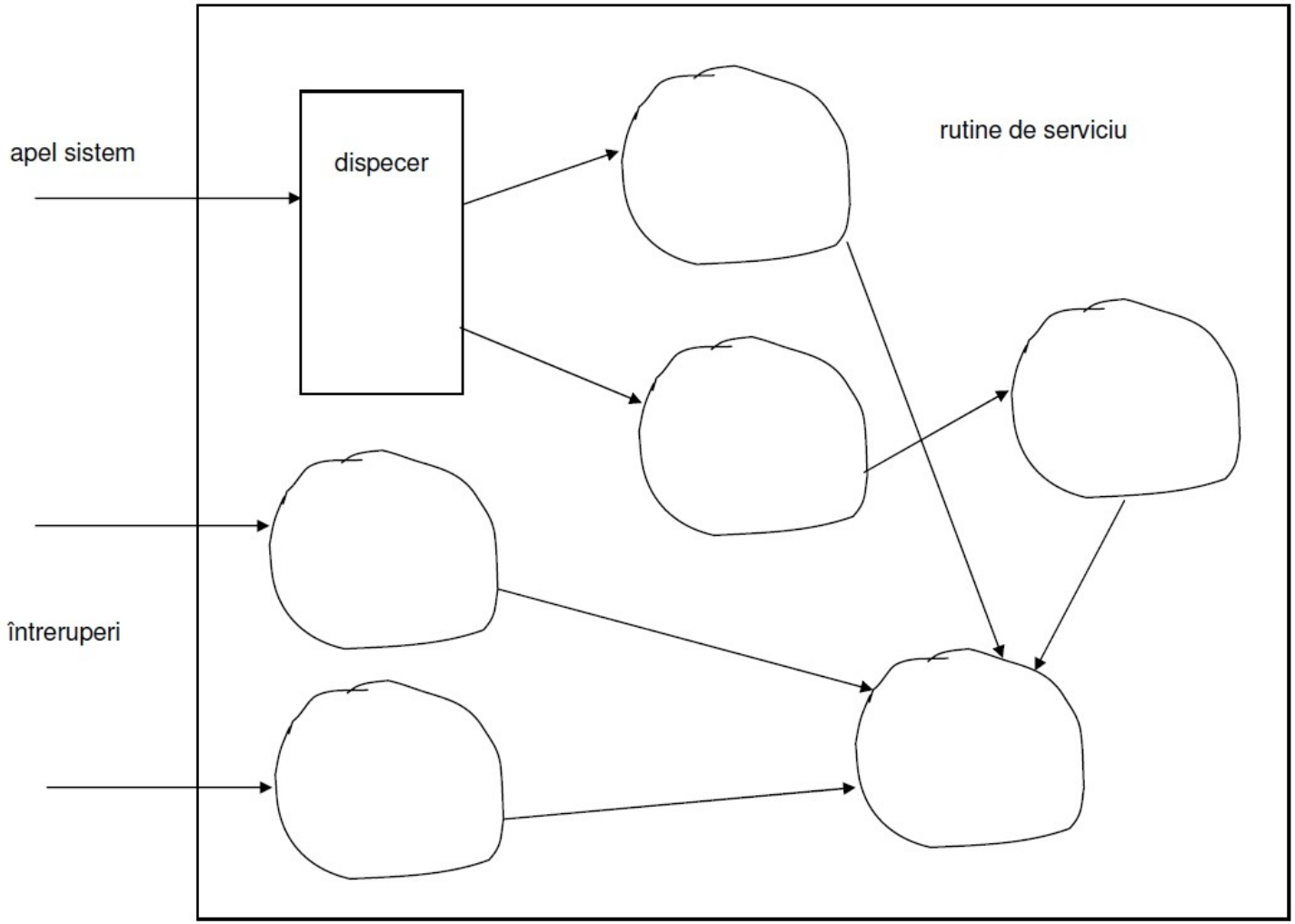


Sursa: Linux Performance and Tuning Guidelines - http://www.redbooks.ibm.com/abstracts/redp4285.html

Interacțiunea componentelor la execuție

- Sistemul de operare poate fi privit ca o colecţie de rutine invocate prin apeluri sistem sau ca urmare a unor întreruperi
- Apelurile sistem:
  - sunt diferite de apelurile normale de rutine,
  - duc la creşterea nivelului de privilegii în timpul execuţiei
  - pot folosi toate instrucţiunile
  - au acces la toti registrii procesorului
  - au acces la toate locaţiile de memorie.
  - au un mecanism de declanşare care este analog producerii unor întreruperi, dar sursa este în program şi nu în afara acestuia
  - au acelaşi punct de intrare în sistemul de operare
  - fiecare apel este însoţit de informaţii suplimentare (codul apelului) transmise prin registrii procesorului
  - codul apelului este folosit de dispecer pentru a selecta rutina adecvată tratării apelului.

Declanşarea execuției rutinelor sistemului de operare



Structura pe niveluri a componentelor

- Sistemele de operare sunt structurate pe niveluri deoarece fiecare nivel poate fi dezvoltat şi testat separat
- Orice procedură a unui sistem de operare ar trebui să poată apela orice altă procedură sau ar putea avea acces la orice structură de date ceea ce ar duce la o mare dificultate în determinarea efectelor schimbărilor în proceduri sau în structurile de date.
- Nu există reguli generale în repartizarea funcţiilor pe fiecare nivel
- Sarcinile critice ca planificarea proceselor, gestiunea memoriei, protecţia, trebuie să apară pe un nivel mai jos decât cele ca gestiunea fişierelor şi interfaţa cu utilizatorul.
- Numărul de niveluri nu poate fi prescris prin reguli fixe

Adaptabilitatea la configurația hardware

- este de dorit ca sistemele de operare să fie cât mai portabile
- trebuie să se poată adapta la orice configuraţie
  - poate duce la unele modificări în codul sursă sau doar la o specificare a unor anumiţi parametri la iniţializarea sistemului
- să detecteze automat caracteristicile hardware ale sistemului; să se "autoconfigureze" la încărcare

Procese

- Definiţii:
  - Procesul este un program în execuţie.
  - Un program este o secvenţă de instrucţiuni.
    - Pentru un program dat (fişier executabil) pot exista unul sau mai multe procese asociate numite instanţe.
    - Procesul reprezintă invocarea dinamică a unui program împreună cu resursele necesare pentru lansarea în execuţie.
    - Resursele necesare pentru rularea unui program includ: stiva utilizator, stiva sistem, memorie, identificatori de fişier, etc.

Procese (2)

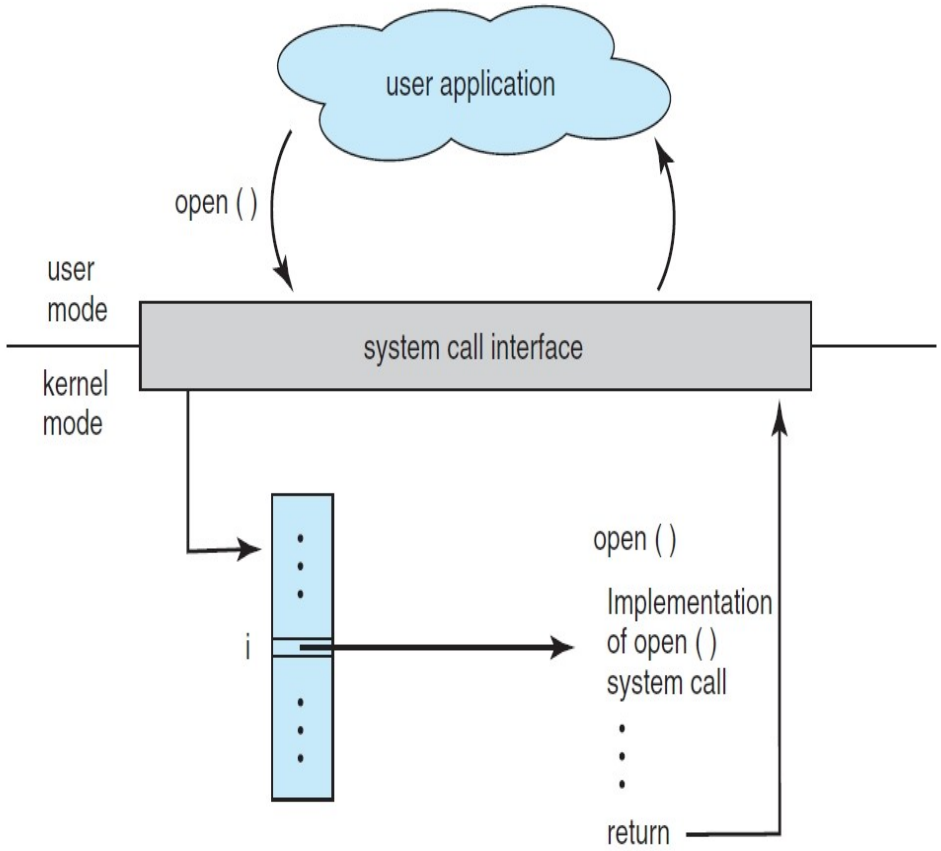
- **Fiecare proces se execută într-un spaţiu de adrese propriu.**
- Spaţiul de adrese virtual este cuprins între 0 şi adresa virtuală maximă accesibilă şi este format din următoarele zone (segmente):
  - Segmentul text
    - codul programului, este protejat la scriere;
  - Segmentul de date
    - date predefinite (cunoscute la compilare) sau date alocate dinamic;
  - Segmentul de stivă
    - conţine argumente, variabile locale şi alte date pentru execuţia funcţiilor în modul utilizator

Procese (3)

- Sistemele uniprocesor:
  - execuţia unui proces este secvenţială
- Sistem multitasking:
  - pe un procesor se pot executa mai multe procese, fiecare proces având alocată o cuantă de timp pentru execuţie după care urmează altul
  - execuţia se numeşte secvenţial concurentă sau aparent paralelă

Execuția unui proces

- Modul utilizator:
  - procesele au acces numai la propria zonă de cod, date şi stivă utilizator;
- Modul nucleu:
  - procesul conţine instrucţiuni privilegiate şi poate avea acces la structurile de date ale nucleului.



Crearea proceselor

- la iniţializarea sistemului (reboot);
- la execuţia unui apel de funcţie pentru creare de procese (fork);
- la cererea unui utilizator de creare a unui nou proces;
- la iniţializarea unui job.

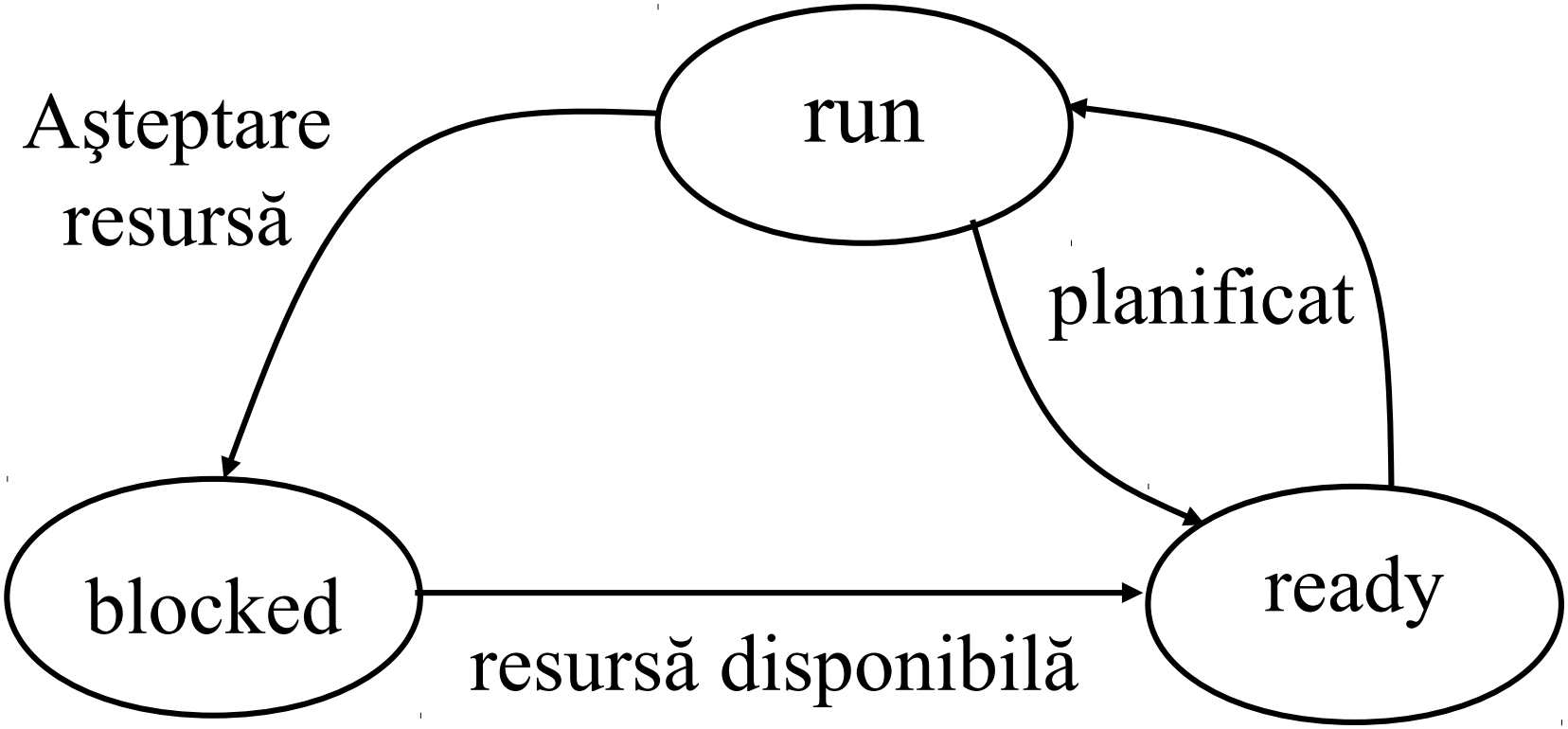
Terminarea execuției proceselor

- normal – terminarea programului (voluntar)
- eroare – terminare cu eroare (voluntar)
- fatal error – divide by 0, core dump (involuntar)
- terminat de un alt proces (involuntar)

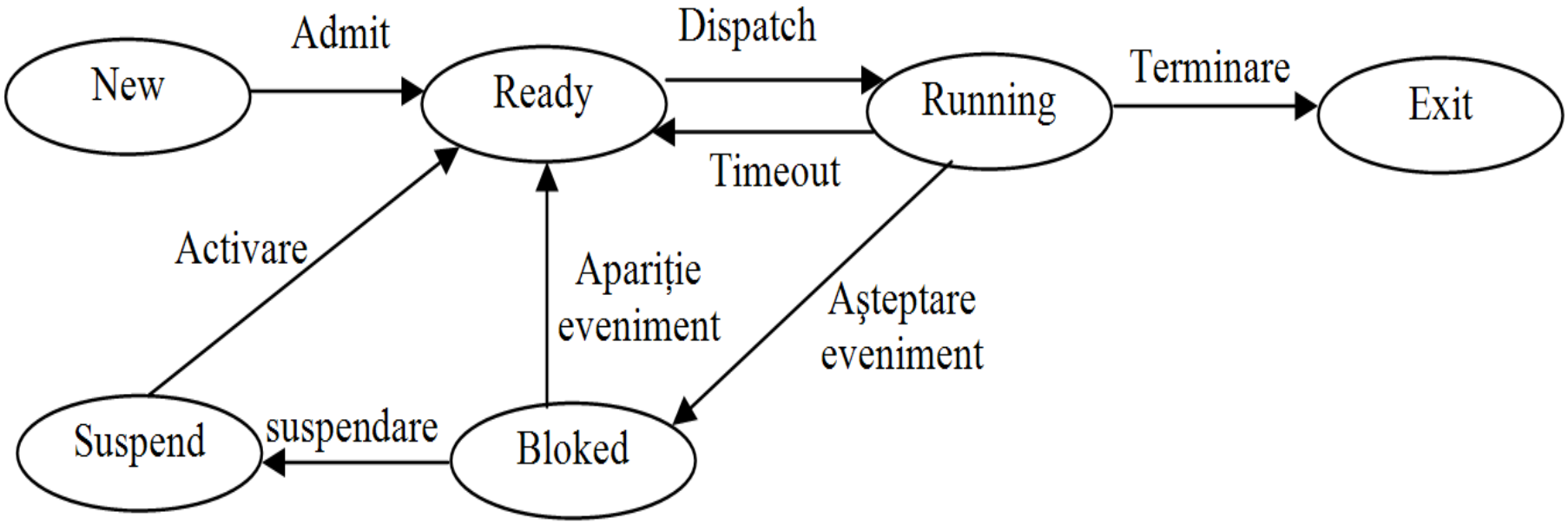
Starile unui proces

- **Starea unui proces evidenţiază activitatea procesului**
  - **New** – Procesul este creat
  - **Run** – Un proces care se execută
  - **Blocat** – Dacă procesul este implicat într-o operaţie de I/O şi dispozitivul periferic nu este liber sau pur şi simplu este lent sau nu este pregătit
  - **Ready** – Dacă sistemul este multitasking, atunci un proces foloseşte procesorul pe durata unor cuante de timp fiind oprit de către sistemul de operare care alege alt proces pentru execuţie
  - **Terminat** – Execuţia procesului este terminată

Diagrama de stare simplificată pentru un proces



Stările unui proces cu o singură stare de suspendare



Stările unui proces - UNIX