

Arhitectura Calculatoarelor
AIA, An 1

NIVELUL SISTEMULUI DE OPERARE

Curs #9

CUPRINS

- Memoria virtuală
- Paginare
- Segmentare
- Fișiere
- Procese
- Exemple de sisteme de operare



INTRODUCERE

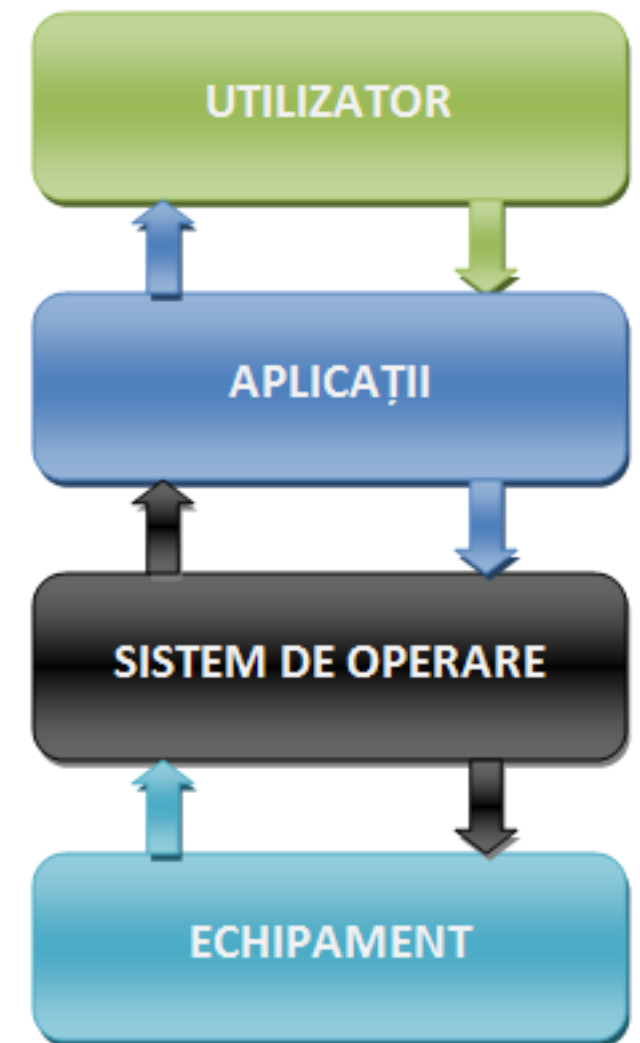
- ❖ Un calculator modern este construit ca o succesiune de niveluri, fiecare adăugând un spor de funcționalitate celui imediat inferior lui.
- ❖ Am studiat până acum nivelul logicii digitale, nivelul microarhitecturii și nivelul arhitecturii setului de instrucțiuni.

SISTEM DE OPERARE

- ❖ Un **sistem de operare** este un program care, din punct de vedere al utilizatorului, adaugă o varietate de noi instrucțiuni și facilități deasupra și dincolo față de ce oferă nivelul ISA.
- ❖ Uzual, sistemul de operare este implementat în cea mai mare parte în software, însă nu există niciun motiv teoretic pentru care el nu ar putea fi plasat în hardware.
- ❖ Pe scurt, vom numi nivelul pe care îl implementează nivelul **OSM** (Operating System Machine).

SISTEM DE OPERARE

- ❖ SO este primul nivel software peste hardware
- ❖ Un SO performant
 - folosește facilitățile hardware
 - expune facilitățile hardware aplicațiilor
- ❖ Hardware-ul are nevoie de SO pentru a putea fi folosit (pentru a construi aplicații / software)
- ❖ SO evoluează pe măsură ce evoluează hardware-ul



FUNCTII SO

- ❖ Gestiunea resurselor sistemului (hardware)
- ❖ Interfața dintre aplicații și hardware
- ❖ Medierea accesului la hardware
- ❖ Securitatea și integritatea sistemului

CONCEPTE OS

- ❖ Subiecte importante în cadrul sistemului de operare:
 - Memoria virtuală, o tehnică utilizată de sistemul de operare pentru a face să pară că mașina are mai multă memorie decât în realitate.
 - Fișierele de I/E, un concept de nivel superior studiat în cadrul cursului anterior.
 - Prelucrarea paralelă, modalitatea prin care mai multe procese pot fi executate, pot comunica și se pot sincroniza.

MEMORIA VIRTUALĂ

- ❖ La începutul apariției calculatoarelor, memoriile erau mici și scumpe. Unul dintre primele compilatoare de ALGOL 60 a fost scris pentru un calculator cu doar 1024 de cuvinte de memorie.
- ❖ Un program putea rula doar în memoria disponibilă existentă.
- ❖ Soluția tradițională a acestei probleme era folosirea memoriilor secundare, de tipul discului.
- ❖ Un programator împărțea programul într-un număr de componente numite faze (overlays), fiecare dintre acestea încăpând în memorie.

MEMORIA VIRTUALĂ

- ❖ Un proces are un spațiu virtual (inexistent de fapt) de memorie
 - Procesul are impresia că toată memoria îi aparține
- ❖ Memoria virtuală decuplează vederea procesului de memoria sistemului
- ❖ Procesele lucrează cu adrese virtuale (din spațiul fizic de adrese)
 - Adresele fizice sunt adrese din memoria fizică
- ❖ SO mapează / asociază spațiul virtual al proceselor cu memorie fizică

MEMORIA VIRTUALĂ

- ❖ Memoria virtuală este o proprietate arhitecturală al cărei scop este să permită programelor să folosească un spațiu de adrese mai mare decât memoria fizică a mașinii, sau să ofere un mecanism consistent și flexibil pentru protecția memoriei și pentru partajare.
- ❖ Acest mecanism se poate implementa prin paginare, segmentare sau o combinație a acestora.
- ❖ La paginare, spațiul de adrese este spart în pagini virtuale de dimensiuni egale. Numai unele dintre acestea sunt puse în corespondență cu cadre de pagini fizice.
- ❖ O referință la o pagină care are corespondent în spațiul fizic este transformată de MMU într-o adresă fizică corectă. O referință la o pagină care nu are corespondent va cauza o eroare de pagină.

PAGINARE

- ❖ Se separă conceptele de spațiu de adrese și locații de memorie.
- ❖ Această tehnică se numește paginare (**paging**), iar bucățile de program citite de pe disc se numesc pagini (**pages**).

PAGINARE

- ❖ Se poate folosi și un mod mai sofisticat de realizare a corespondenței între spațiul de adrese și adresele de memorie disponibile.
- ❖ Vom numi adresele la care programul face referire ca **spațiu de adrese virtuale** (*virtual address space*), iar memoria hardware existentă o vom numi **spațiu de adrese fizice** (*physical address space*).
- ❖ O **hartă de memorie** (*memory map*) sau un **tabel de pagini** (*page table*) stabilește corespondența între adresele virtuale și cele fizice.

THREAD-URI

- ❖ Un proces poate avea mai multe thread-uri
- ❖ Thread-urile partajează spațiul virtual de adrese al procesului
- ❖ Utile să faci mai multe lucruri cu aceleași date (din spațiul virtual de adrese al procesului)
- ❖ Permit folosirea facilităților hardware multi-procesor

CONCURENȚĂ ȘI SINCRONIZARE

- ❖ Thread-urile sau procesele pot concura la achiziția de resurse (date în memorie)
- ❖ Accesele concurente pot duce la date inconsecvente sau blocaje
- ❖ Sincronizarea garantează accesul consecvent și ordonat la date

PROCESE

- ❖ Iau naștere dintr-un program executabil
- ❖ Program în execuție, entitate dinamică
 - Noțiunea de “runtime” (timpul rulării) se referă la proces
- ❖ Date și cod în memorie, se execută pe CPU
- ❖ Are asociate resurse: spațiu de adrese de memorie, fișiere deschise, socketi
- ❖ SO oferă protecție și comunicare inter-proces
- ❖ Ierarhie de procese la nivelul SO

FIȘIERE

- ❖ Unități de stocare
- ❖ Deschise și folosite de procese
- ❖ Fișier pe disc (static) și fișier deschis (dinamic, în cadrul unui proces)
- ❖ Fișier pe disc: nume, dimensiune, permisiuni, timestamp-uri
- ❖ Fișier deschis: handle de fișier, cursor de fișier, drepturi de deschidere, operații pe fișier

FIȘIERE

- ❖ Cea mai importantă abstractizare a I/E prezentă la acest nivel este fișierul.
- ❖ Un fișier constă dintr-o secvență de octeți sau înregistrări logice care pot fi citite sau scrise fără a se cunoaște cum lucrează discurile, benzile magnetice sau dispozitivele de I/E.
- ❖ Accesul la fișiere poate fi secvențial, direct după numărul de înregistrare, sau direct după o cheie.
- ❖ Cataloagele pot fi folosite pentru a grupa fișiere.

EXEMPLE DE SISTEME DE OPERARE

- UNIX și Windows sunt două sisteme de operare sofisticate.
- oferă suport pentru paginare și pentru punerea în corespondență a fișierelor cu zone de memorie.
- oferă suport pentru sisteme ierarhice de fișiere.
- oferă suport pentru procese și fire de execuție și furnizează mijloace de a le sincroniza.

REZUMAT

- **Sistemul de operare** poate fi privit ca un interpretor pentru anumite proprietăți arhitecturale care nu se găsesc la nivelul ISA.
- Cele mai importante sunt: memoria virtuală, instrucțiunile de I/E virtuale și facilitățile pentru prelucrări paralele.

RESURSE



- Andrew S. Tanenbaum, Arhitectura calculatoarelor.
- Operating Systems Basics