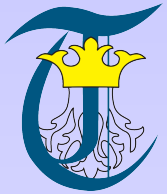


Universitatea *Transilvania* Brașov
Facultatea de Inginerie Electrică și Știința Calculatoarelor
Departamentul de Electronică și Calculatoare

Sisteme de operare

– curs 1 –

2013.02.27

ș.l. dr. ing. Kertész Csaba-Zoltán

Contact

- web:
  - http://etc.unitbv.ro/~csaba.kertesz/so
- e-mail:
  - csaba.kertesz@etc.unitbv.ro

Structura Disciplinei

- 2 ore curs / săptămână
  - prezența (>10) punct bonus la examen
- 1/2 ore laborator
  - prezența obligatorie
  - teme de casă
- examen
  - 30 întrebări / 30 minute contra-cronometru
- nota finală
  - 70% examen + 30% teme de laborator
  - minim 5 la amândouă

Structura cursului

- Capitolul 1. Introducere
  - Ce este un SO ? Istoria sistemelor de operare. Concepte de baza ale SO. Procese. Fișiere. System calls. Shell-ul. Structura SO. Sisteme monolitice. SO organizate pe nivele. Mașini virtuale. Modelul client-server.

Structura cursului

- Capitolul 2. Procese.
  - Implementarea proceselor. Comunicația între procese (IPC). Condiții de concurență Secțiuni critice. Excluderea mutuala. Sleep & wakeup. Semafoare. Contoare de evenimente. Monitoare. Message passing. Probleme IPC clasice. Problema cinei filozofilor. Problema scriitorilor și cititorilor. Planificarea proceselor. Round Robin scheduling. Priority scheduling. Multiple queues. Shortest Job First. Planificarea garantata. Politici și mecanisme.Thread-uri.

Structura cursului

- Capitolul 3. Managementul memoriei.
  - Managementul memoriei fără utilizarea paginării și swapping-ului. Multiprogramarea si utilizarea memoriei. Swapping. Multiprogramare utilizând partiții variabile. Managementul memoriei utilizând bit maps. Managementul memoriei utilizând buddy system. Alocarea spațiului pentru swap. Analiza sistemelor cu swapping. Memoria virtuala. Paginarea. Page tables. Exemple de hardware pentru paginare. Memoria asociativa. Algoritmi de înlocuire a paginilor. Algoritmul de înlocuire optim. Not Recently Used page replacement algorithm. FIFO page replacement algorithm. Second Chance page replacement algorithm.

Structura cursului

- Clock page algorithm. Last Recently Used algorithm. Simularea software a algoritmului LRU. Modelarea algoritmilor de înlocuire a paginilor. Anomalia lui Belady. Algoritmi de tip stiva. Predicția ratei page fault error. Considerații privind proiectarea sistemelor de paginare a memoriei. Modelul working set. Politici de alocare locale și globale. Dimensiunea paginilor. Considerații privind implementarea sistemelor de paginare a memoriei. Segmentarea.

Structura cursului

- Capitolul 4. Sisteme de fișiere
  - Fișiere. Numele fișierelor. Structura fișierelor. Tipuri de fișiere. Accesul la datele dintr-un fișier. Atributele unui fișier. Operații cu fișiere. Directoare. Implementarea sistemelor de fișiere. Implementarea fișierelor. Implementarea directoarelor. Fișiere partajate. Managementul spațiului de pe disc. Fiabilitatea sistemului de fișiere. Performantele sistemului de fișiere. Securitatea sistemului de fișiere. Probleme celebre în securitatea sistemelor. The Internet worm. Atacuri generice. Autentificarea utilizatorilor. Mecanisme de protecție. Domenii de protecție. Liste de control al accesului. Capabilități. Modele de protecție.

Structura cursului

- Capitolul 5. Managementul I/O
  - Hardware-ul I/O. Dispozitive I/O. Device controllers. Direct Memory Access. Software I/O. Cerințele software-ului I/O. Interrupt handlers. Device drivers. Device independent I/O software. User space I/O software. Discuri. Disk Arm Scheduling algorithms. Timers. Clock hardware. Clock software. Terminale. Hardware-ul unui terminal. Memory mapped terminals. Input software. Output software.

Bibliografie

- Silberschatz, P.B. Galvin, Operating System Concepts, Addison Wesley, 1998
- A.S. Tanenbaum, Modern Operating Systems, Prentice Hall, 1992
- W.R. Stevens, Advanced Programming in the UNIX Environment, Addison Wesley, 1992
- D. Grosu, Sisteme de Operare - Indrumar de Laborator, Universitatea Transilvania Brașov, 1998
- V. Cristea, A. Panoiu, E. Kalisz, I. Athanasiu, L. Negreanu, S. Calinoiu, F. Baboescu, UNIX, Teora, 1994
- Guido Gonzato, DOS/Windows to Linux HOWTO

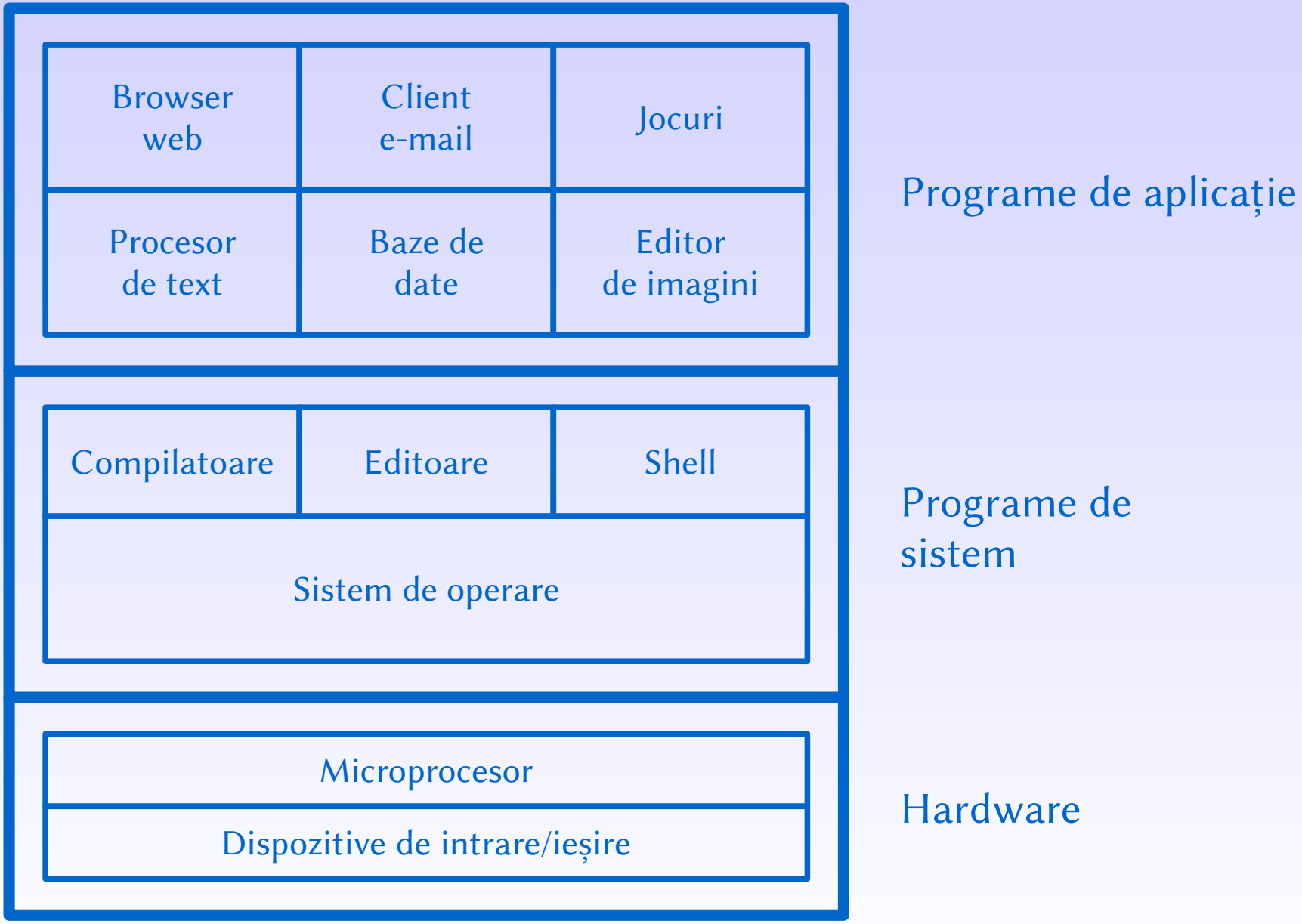
Capitolul 1. Introducere

Cuprins

1.1. Ce este un SO?

- Software:
  - programe de sisteme (controlează activitatea sistemului)
  - programe utilizator (rezolvă problemele utilizatorilor)
- SO reprezintă o componentă de bază a clasei programelor de sistem
- SO controlează toate resursele calculatorului și oferă acces la acestea programelor utilizator într-un mod convenient și sigur

Software


Browser web
Client e-mail
Jocuri
Procesor de text
Baze de date
Editor de imagini
Programe de aplicație
Compilatoare
Editoare
Shell
Sistem de operare
Programe de sistem
Microprocesor
Dispozitive de intrare/ieșire
Hardware

Software

- microprocesorul controlează dispozitivele IO prin registre speciale
- SO ascunde acest nivel, oferind o interfață comună pentru programele aplicație
- compilatoare, editoare, shell-uri sunt furnizate împreună cu SO, dar nu fac parte din aceasta

Rolul SO

- realizează o mașină extinsă:
  - funcția SO este de a prezenta utilizatorului un echivalent a unei mașini extinse, mai ușor de programat decât hardware-ul direct
- administrează resursele:
  - urmărește cine și ce resurse utilizează și mediază conflictele de acces

1.2. Istoria SO

- Evoluția SO poate fi împărțită în 4 generații în strânsă corelație cu evoluția calculatoarelor

Generația I.
(1945 – 1955)

- calculatoare cu tuburi electronice, fără SO
- programare în limbaj mașină prin legături fizice
- nu există limbaje de programare
  - de la 1950 s-a început folosirea cartelelor perforate
- nu există SO

Generația II.
(1955 – 1965)

- calculatoare cu tranzistoare, SO: sisteme batch
- joburi rulate de pe cartele perforate
- programe se scriau în asamblare sau FORTRAN (și „compilate” în cartele perforate)
- în scopul creșterii eficienței mașinii au apărut sistemele batch, care eliminau operațiile manuale între joburi
  - mai multe joburi se scria pe bandă magnetică
  - de pe bandă mașina prelua joburile pe rând și oferea rezultate pe o bandă de ieșire

Generația III.
(1965 – 1980)

- calculatoare cu IC, SO: multiprogramare
- multiprogramarea era introdusă pe familia de calculatoare OS/360
- utilizare eficientă a mașinilor:
  - în timpul execuției unui job, aceasta trece prin faze care folosesc intensiv procesorul și faze care lucrează cu dispozitive IO lente
  - s-ar putea executa două joburi simultan fără să se deranjeze dacă aceste se află în faze complementare

Generația III.

- soluție:
  - partiționarea memoriei în câteva partiții având fiecare câte un job
  - când un job așteaptă terminare unei operații IO, procesorul este oferit unui alt job din memorie
- creștere a utilizării timpului de procesor
- mecanisme de protecție între joburi

Generația III.
tehnici SO

- spooling (Simultaneous Peripheral Operation On-Line)
  - după terminarea fiecărui job, SO încarcă un nou job de pe disc în partiția goală
- time-sharing
  - joburile concurente pe sistem pot deține controlul procesorului maxim o perioadă prestabilită, după care controlul este oferit altui job

Generaţia IV.
(1980 – ...)

- integrare de nivel înalt, PC-uri, SO complexe
- software user-friendly
- multe procese rulate în paralel
- SO dominante:
  - DOS/Windows
  - UNIX/Linux
  - MacOS X