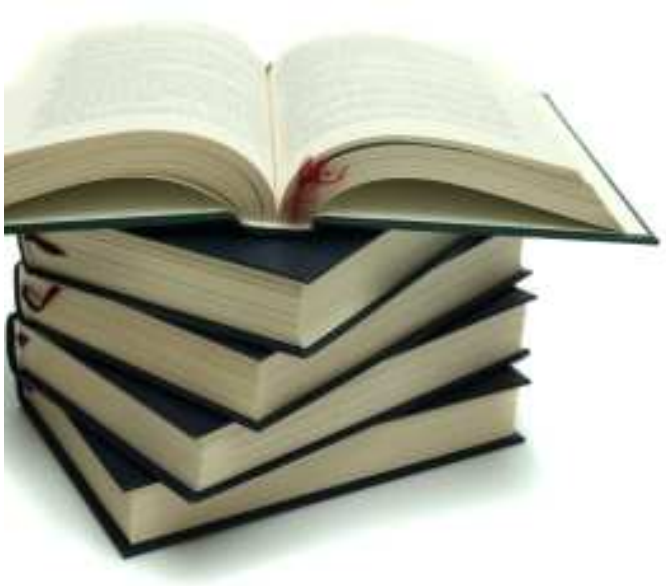
page crunch

Cursul 1



1

Introducere în sisteme de operare

25 februarie 2009

Cuprins

- Prezentare generală a cursului
- Ce este un sistem de operare?
- Istoria sistemelor de operare
- Clasificarea sistemelor de operare
- Concepte hardware de bază
- Concepte de bază în SO
- Componența și structura unui SO

Echipa

- Andrei Pitiș, Răzvan Deaconescu, Octavian Purdilă
- Mircea Bardac, Daniel Băluță, Mircea Gherzan, Lucian Grijincu, Andrei Ismail, George Milescu, Mihnea Muraru
- Mult succes în noul semestru!

Obiectivele cursului

- Make you a better engineer!

- Prezentarea mecanismelor puse la dispoziție de SO moderne pentru dezvoltarea de aplicații
  - Procese şi thread-uri
  - Comunicația între procese şi thread-uri
  - Accesul la memorie; gestiunea memoriei
  - Accesul la alte resurse puse la dispoziție de SO moderne (fişiere, dispozitive de I/E)

Obiectivele laboratorului

- Aplicarea conceptelor prezentate la curs
- Prezentarea și familiarizarea cu interfețele de programare de sistem (system API): POSIX & Win32
  - lucrul cu fișiere
  - lucrul cu procese/thread-uri
  - comunicația între procese/thread-uri
  - gestiunea memoriei
  - operații de I/E

Reguli şi notare

- site-ul cursului: http://cs.pub.ro/~so
- listă de discuții: http://cursuri.cs.pub.ro/liste/so
- curs – 3 puncte
  - lucrare – 1 punct (6 subiecte x 0.2 puncte) – 40 de minute
    - săptămâna a 8-a la curs
    - nu se reface
  - final – 2 puncte (11 subiecte x 0.2 puncte) – 70 de minute
    - sesiune
  - minim 1 punct din 3 pentru absolvirea cursului
- activitate laborator – 2 puncte
  - NU se punctează prezența

Reguli și notare (2)

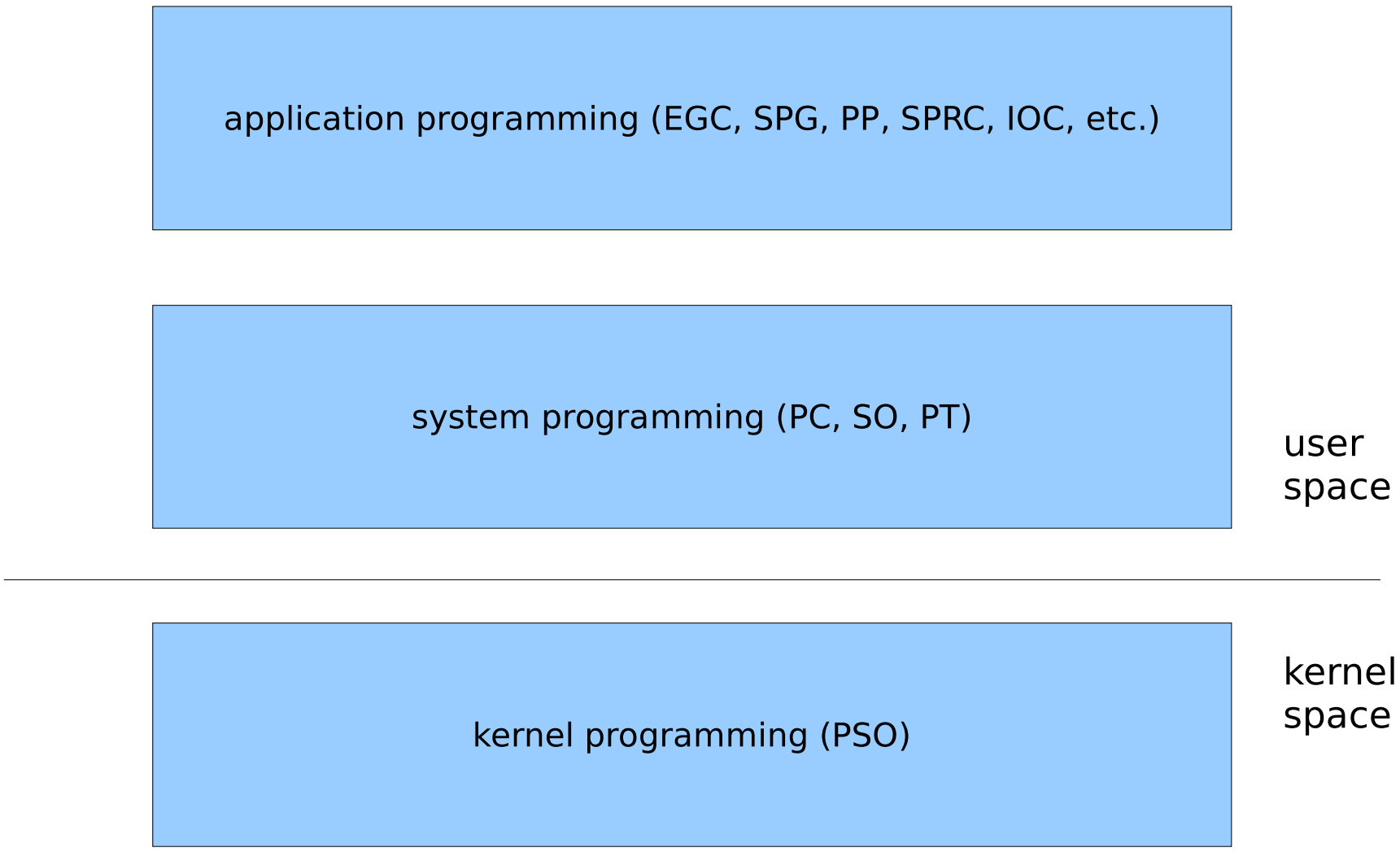
- 5 teme x 2 (Linux, Windows) – 10 puncte
  - o temă copiată -> punctaj 0 la toate temele
  - primele 5 teme (în ordinea punctajului) vor fi punctate integral
  - următoarele 5 teme vor fi punctate raportat cu nota de curs
  - corectare cu teste publice
- depunctare teme
  - -0.25p pe zi (din 10) timp de 12 zile
  - după 12 zile nota maximă pentru o temă este 7
  - deadline absolut: o săptămână înainte de examen
- punctajul de absolvire a cursului este 4.5
- după restanțe tot punctajul se resetează la 0

Legătura cu celelalte materii

- Prerequisites
  - USO
  - Programare, SDA
  - PLAS, CN
  - PC, RC
- Materii ce depind de SO
  - SO2
  - PC, SSC, APC, SPRC

Where do we stand?

Where do we stand? (programming)



Despre curs

- 12 cursuri
- interactiv
  - participați la discuții
  - întrebați atunci când nu ați înțeles

- destul de “dens”
- se recomandă călduros parcurgerea suportului bibliografic înainte și după curs
- slide-urile nu sunt suficiente pentru a înțelege materia

Despre laborator

- POSIX/Win32 API programming (C/C++)
- 20 min prezentare / 80 minute lucru
- se punctează activitatea
- learn by doing
- se punctează exercițiile rezolvate în laborator

Despre teme

- tema 1 – mini-shell
- tema 2 – sistem client-server cu IPC
- tema 3 – demand pager/swapper
- tema 4 – monitor generic
- tema 5 – server de fișiere

Despre teme (2)

- intense
- necesare: aprofundare API (laborator) și concepte (curs)
- estimare de timp: 8-20 ore pe temă
- teste publice
- suport de testare la submit - feedback imediat

Relevanță system programming

- Better programmer through intimate knowledge of how computers operate
- C, C++, C#, Java – doar limbaje de programare, operează cu acelasi concepte de multi-threading, I/O (async)
- Better jobs – better pay
- Voi ce credeți?

Bibliografie

- curs
  - Galvin, Silberschatz, Gagne – Operating System Concepts, 7th Edition
  - Andrew Tanenbaum - Modern Operating Systems, 2nd Edition
- laborator
  - Robert Love – Linux System Programming
  - Johnson Hart – Windows System Programming, 3rd Edition

Bibliografie suplimentară

- Mathew, Stones - Beginning Linux Programming, 4th Edition
- Stephens, Rago – Advanced Programming in the Unix Environment, 2nd Edition
- Rector, Newcomer – Win32 Programming
- Charles Perzold – Programming Windows, 5th Edition
- John Levine – Linkers and Loaders

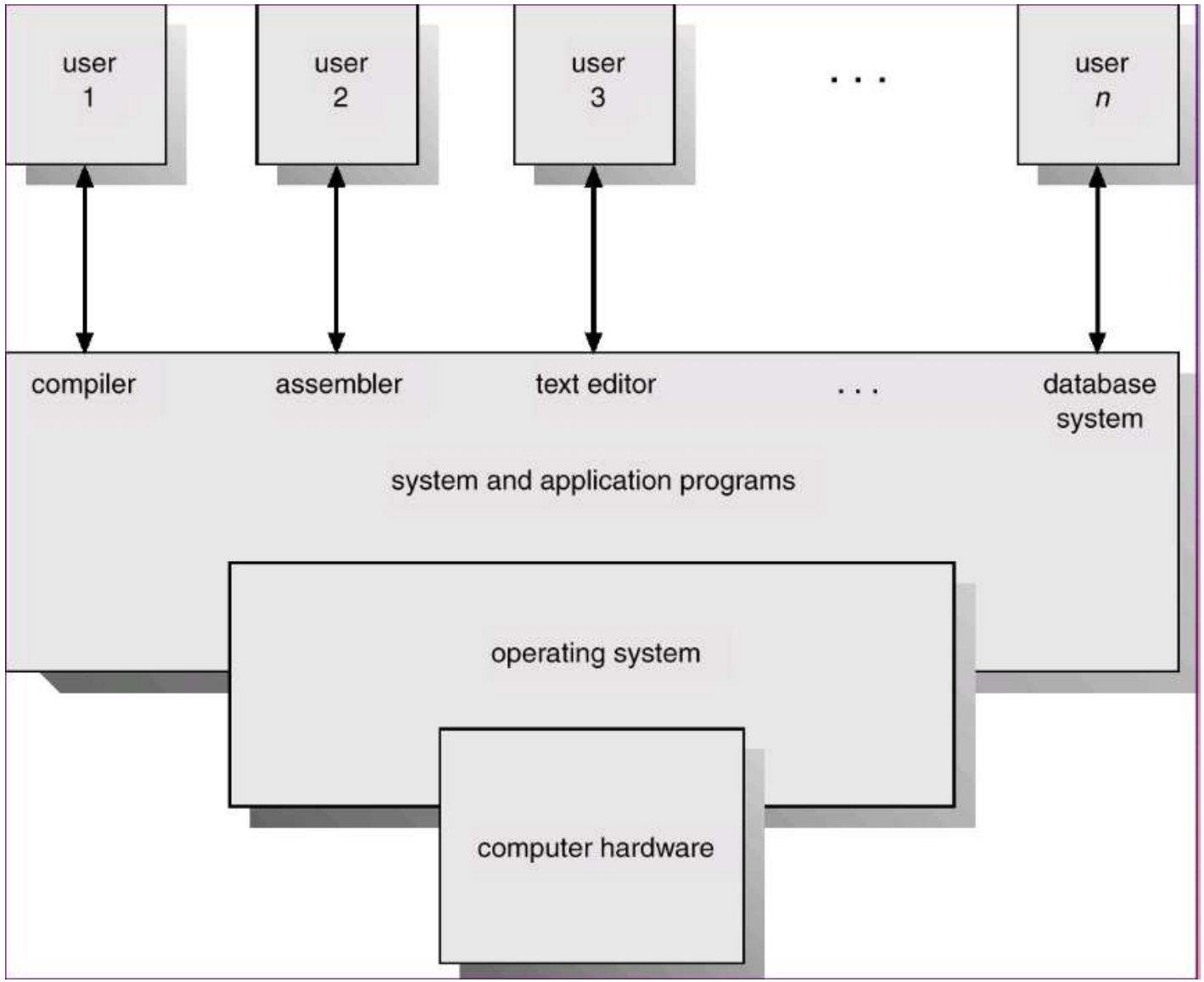
Suport curs 1

- OSC
  - Chapter 1: Introduction
  - Chapter 2: Operating-System Structures
- MOS
  - Chapter 1: Introduction

Ce este un SO?

- Extensie a maşinii fizice (vedere top-down)
  - abstractizează operațiile mașinii fizice în operați mai simple pentru ușurința utilizării
  - ex: accesul la fișiere
- Gestionar al resurselor mașinii fizice (vedere bottom-up)
  - utilizatorii accesează resursele comune ale sistemului
  - SO are rolul de multiplexare a accesului
  - ex: gestiunea procesorului, a memoriei

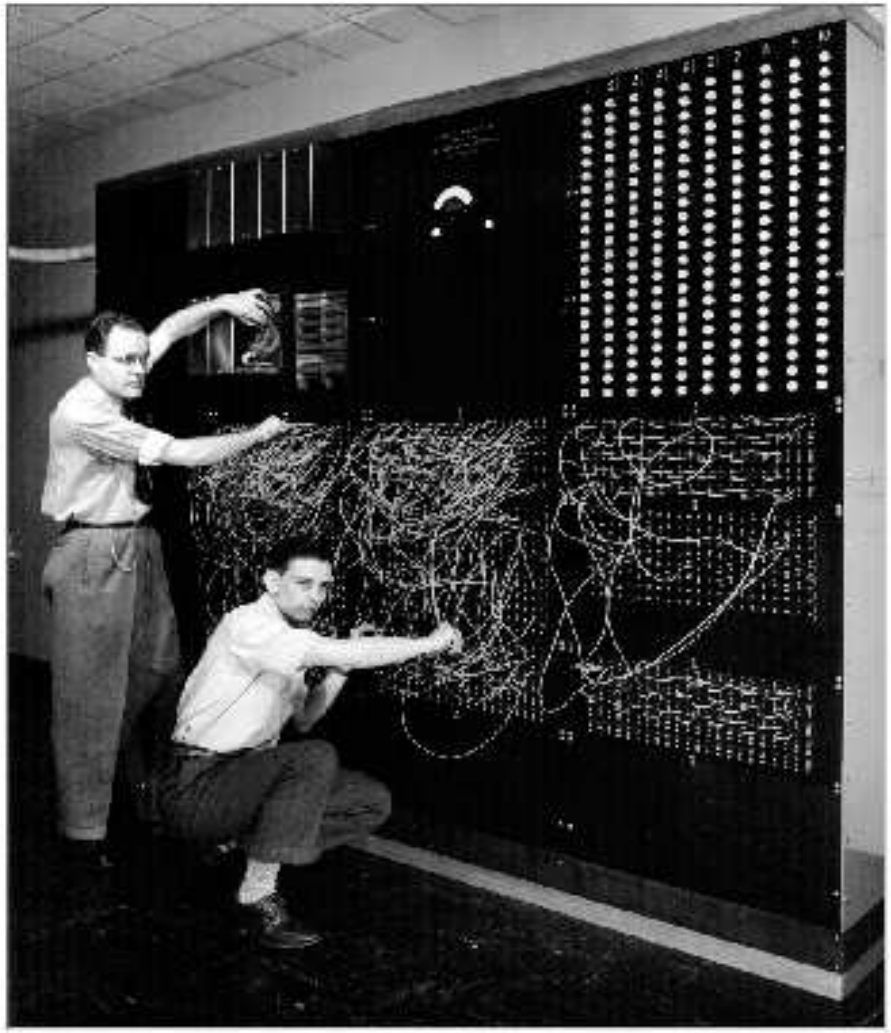
Ce este un SO? (2)



Istoria sistemelor de operare

- dezvoltarea sistemelor de calcul și a sistemelor de operare s-au influențat reciproc
  - dificultatea programării mașinilor a dus la adăugarea de facilități în SO
  - dificultăți ale implementării SO au dus la adăugarea de facilități în hardware (memoria virtuală)

Istoria sistemelor de operare (2)



Harvard Mark-I in use, 1944

- prima generație (1945 -1955)
  - primele calculatoare digitale construite: relee electromecanice, tuburi
  - programare se făcea manual, în limbaj mașină
  - nu existau compilatoare sau asambloare
  - nu existau sisteme de operare

Istoria sistemelor de operare (3)

- generația a doua (1955 -1965)
- tranzistoare, mainframeuri
- apare conceptul de batch
- sisteme de operare: FMS, IBSYS

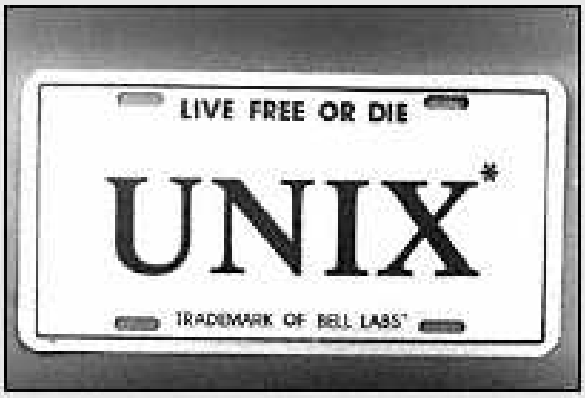
First bug

- On September 9th, Grace Hopper recorded the first actual computer "bug" — a moth stuck between the relays and logged at 15:45 hours on the Harvard Mark II.
- Hopper, a rear admiral in the U.S. Navy, enjoyed successful careers in academia, business, and the military while making history in the computer field.
- She helped program the Harvard Mark I and II and developed the first compiler, A-0. Her subsequent work on programming languages led to COBOL, a language specified to operate on machines of different manufacturers.

Istoria sistemelor de operare (4)

- generația a treia (1965 -1980)
- circuite integrate
- apare conceptul de familie de calculatoare (IBM System/ 360): aceeași arhitectură și set de instrucțiuni
- multiprogramare
  - partiționarea memoriei în mai multe segmente
  - cât timp un job așteaptă la I/O alt job se execută
- spooling
  - citirea joburilor de pe cartele perforate și păstrarea lor pe disc până la execuție

UNIX



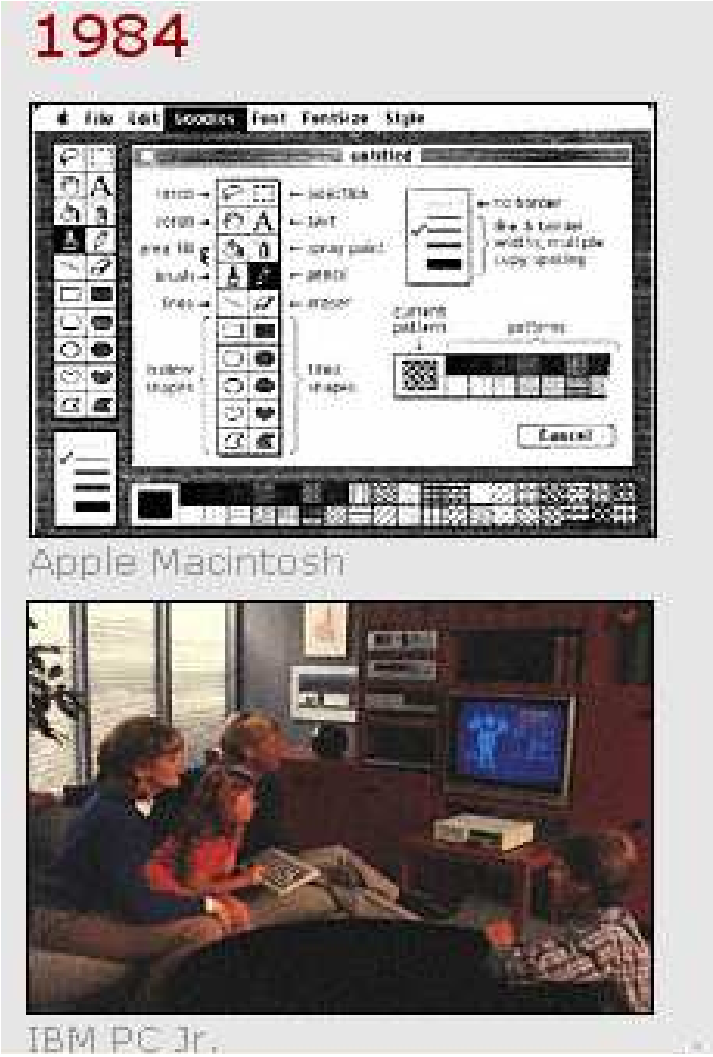
UNIX "license plate"

- AT&T Bell Laboratories programmers Kenneth Thompson and Dennis Ritchie developed the UNIX operating system on a spare DEC minicomputer. UNIX combined many of the timesharing and file management features offered by Multics, from which it took its name. (Multics, a projects of the mid-1960s, represented the first effort at creating a multi-user, multi-tasking operating system.) The UNIX operating system quickly secured a wide following, particularly among engineers and scientists.

Istoria sistemelor de operare (5)

- generația a treia (1965 -1980)
- multitasking (time-sharing)
  - CTSS (Compatible Time Sharing System)
  - MULTICS (Multiplexed Information and Computing Service)
    - MIT, Bell Labs, General Electric
    - lansat în 1960 are un succes comercial scăzut
    - influență masivă asupra dezvoltării ulterioare ale SO
  - UNIX
    - o versiune mult redusă a MULTICS
    - implementat de Ken Thompson
    - portabil (scris în C)
    - System V, BSD

Istoria sistemelor de operare (6)



- generația a patra (1980 -prezent)
  - microcalculatoare şi calculatoarele personale
  - CP/M
    - dezvoltat de Kidall pentru Intel 8080
  - MS-DOS
    - cumpărat de Microsoft de la Seattle Computer ($50, 000)
    - oferit împreună cu BASIC pentru IBM PC
  - Mac OS
    - Steve Jobs “fură” ideea de GUI de la Xerox
  - MS Windows: 3.11, 95, ..
    - influențat de Mac OS

Istoria sistemelor de operare (7)

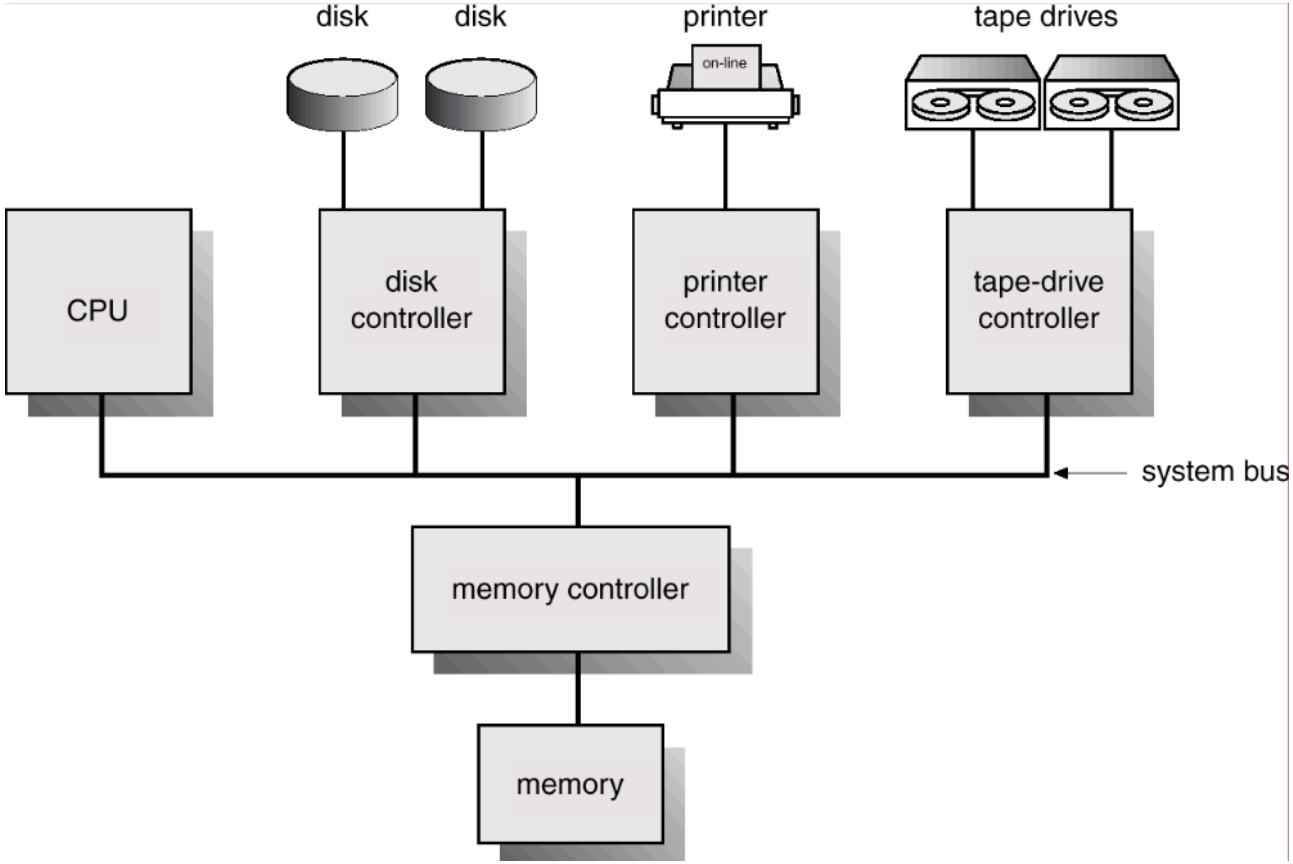
- Generația a patra (1980 -prezent)
  - MS Windows NT, 2000, XP, Vista
    - Scris de la zero, 32/64 biți
    - David Cutler (VAX VMS)
  - Solaris, IRIX, HP-UX, ULTRIX
  - Tru64
    - primul sistem de operare pe 64 biți (procesoare Alpha)
  - Linux
    - clonă UNIX
    - scris de la zero de Linus Torvalds
  - FreeBSD, OpenBSD, NetBSD

Clasificare SO

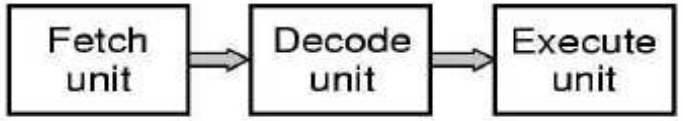
- SO pentru servere
- SO pentru sisteme multiprocesor
- SO pentru calculatoare personale
- SO pentru sisteme embedded
- RTOS

Concepte hardware de bază

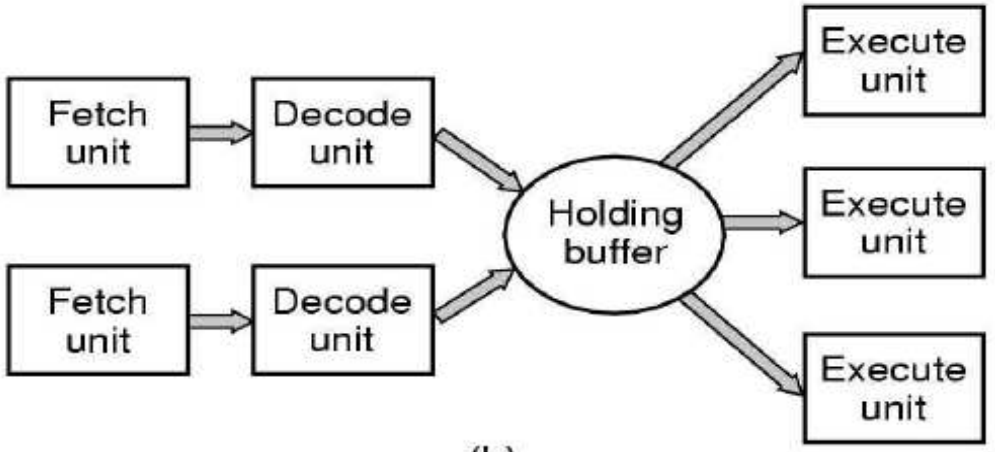
- SO interactionează cu hardware-ul la un nivel destul de scăzut
- sunt necesare cunoștințe despre hardware pentru a înțelege modul de funcționare a sistemului de operare



Procesoarele



(a)



(b)

- Arhitecturi
  - bandă asamblare (a)
  - superscalar (b)
  - VLIW/EPIC

- Arhitecturi
  - RISC
  - CISC

Memoria

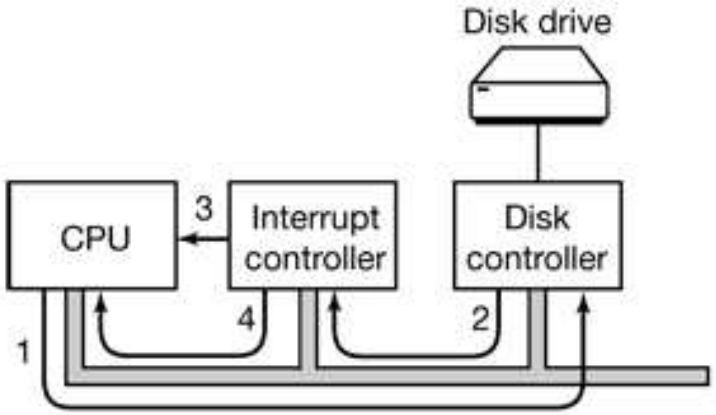
Nivel	1	2	3
Nume	registre	cache	memorie principală
Dimensiune	32/64/128 biți	~ MB	~ GB
Tehnologie	specializată	CMOS SRAM (on-chip sau off-chip)	CMOS DRAM
Timp de acces (ns)	0,25-0,5	0,5-25	80-250
Lățime de bandă (MB/s)	20.000-100.000	5000-10.000	1000-5000
Controlată de	compilator	hardware	sistemul de operare
Susținută de	cache	memoria principală	disc

Memoria cache

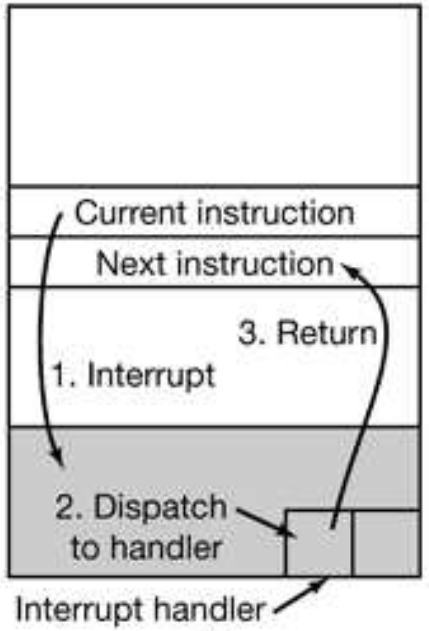
- Introdusă pentru a crea iluzia unei memorii ieftine, rapide și de capacitate mare
- Probleme de
  - consistență: sisteme multitasking
  - coerență: sisteme multiprocesor
- Tipuri de memorie cache
  - cu mapare directă
  - asociative total
  - asociative pe mai multe căi

Dispozitive de I/E

- în general sunt compuse din două părți
  - un controller
  - dispozitivul efectiv
- părțile din SO care controlează dispozitivele de I/E se numesc device drivere
- lente: mouse, tastatura
  - comunicația între dispozitivele de I/E lente şi procesor se poate face prin polling
- rapide: discuri, placa de rețea, placa video
  - pentru controlul acestor dispozitive se folosesc întreruperi și controllere DMA



(a)



(b)

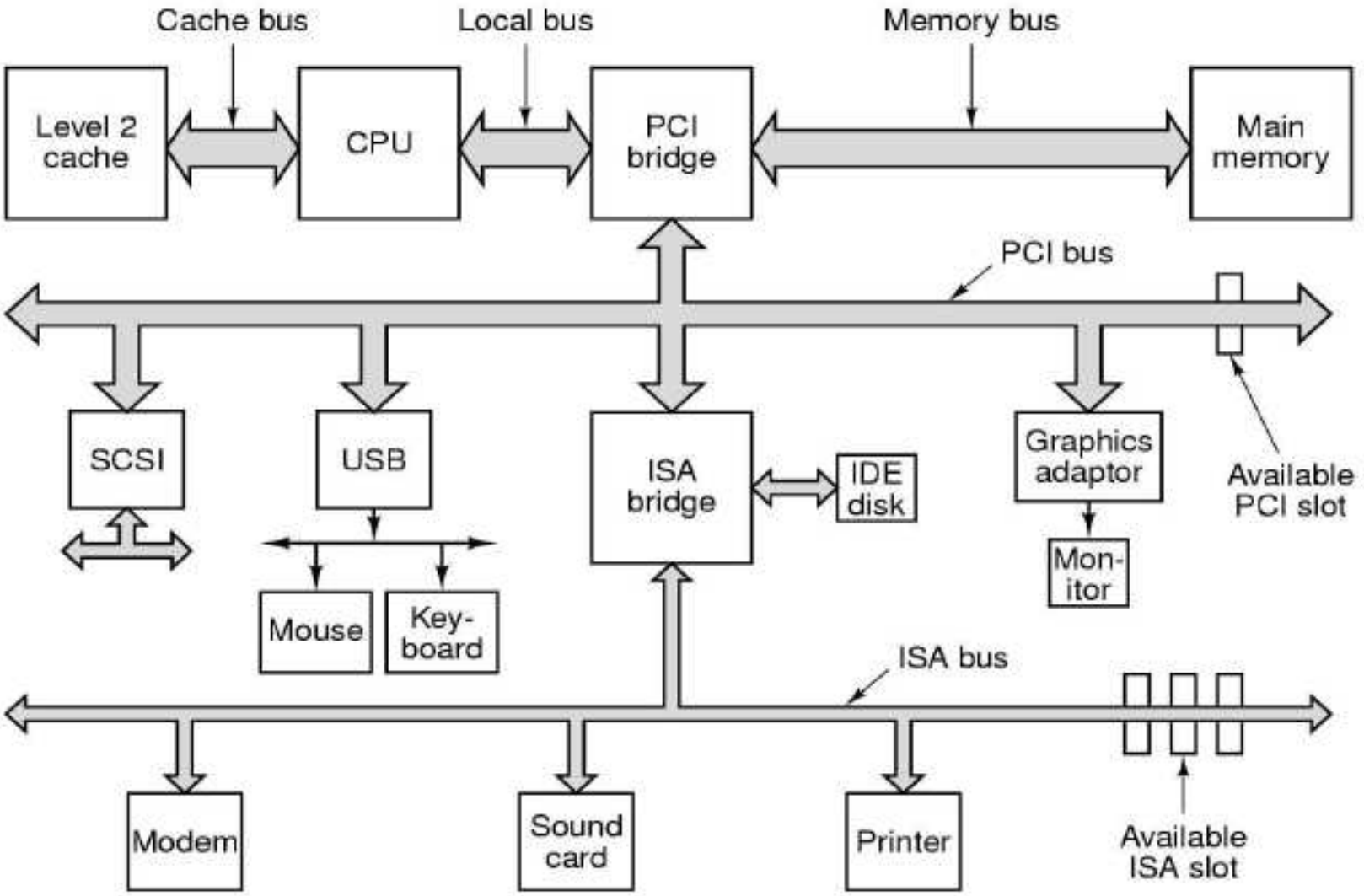
(a) activarea unui dispozitiv I/E și primirea unei întreruperi

(b) tratarea unei întreruperi

DMA

- folosit în cazul transferurilor mari de date între dispozitivul de I/E şi memorie
  - procesorul programează transferul
  - transferul este efectuat de un controller dedicat (DMA)
  - la încheierea transferului, controller-ul DMA emite o întrerupere

Magistrale


Cache bus
Local bus
Memory bus
Level 2 cache
CPU
PCI bridge
Main memory
PCI bus
SCSI
USB
ISA bridge
IDE disk
Graphics adaptor
Available PCI slot
Mon-itor
Mouse
Key-board
ISA bus
Modem
Sound card
Printer
Available ISA slot

Magistrale (2)

- Linii de
  - adresă (determină spațiul de adresă)
  - date (împreună cu frecvența magistralei determină lățimea de bandă)
  - control
- Lățimea de bandă
  - numărul de linii de date x frecvența magistralei

Concepte de bază

- procese, fire de execuție
- deadlock
- memorie virtuală
- sisteme de fișiere
- interpretorul de comenzi
- nucleul sistemului de operare
- kernel mode vs user mode
- kernel space vs user space
- apeluri de sistem

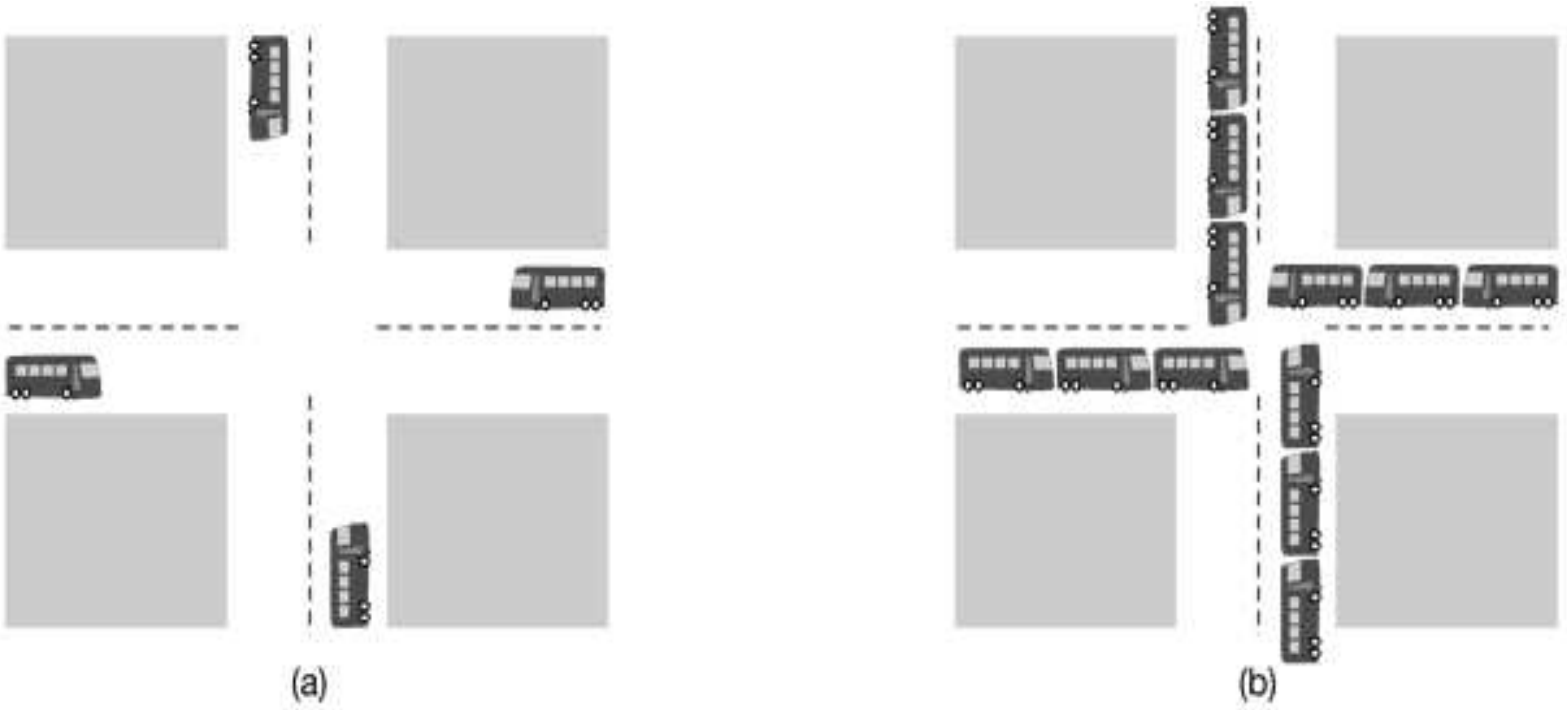
Procese

- un program în execuție
- are asociate mai multe resurse:
  - un spațiu de adrese
  - fișierele deschise
  - alte resurse (memorie partajată, socketi, etc)
- în general procesele sunt ierarhizate după relația părinte-copil
- SO oferă protecție dar şi comunicație interprocese

Fire de execuție

- un proces poate avea mai multe fire de execuție
- firele de execuție dintr-un proces partajează resursele acestuia (memorie, fișiere deschise etc.)
- fiecare fir de execuție are un context
  - context = informații despre starea thread-ului (stivă, registre generale, registre speciale)
- avantaje / dezavantaje
  - paralelism cu o comunicație extrem de facilă și rapidă
  - se pierde mai puțin timp când se face o schimbare de context
  - nu există protecție între firele de execuție

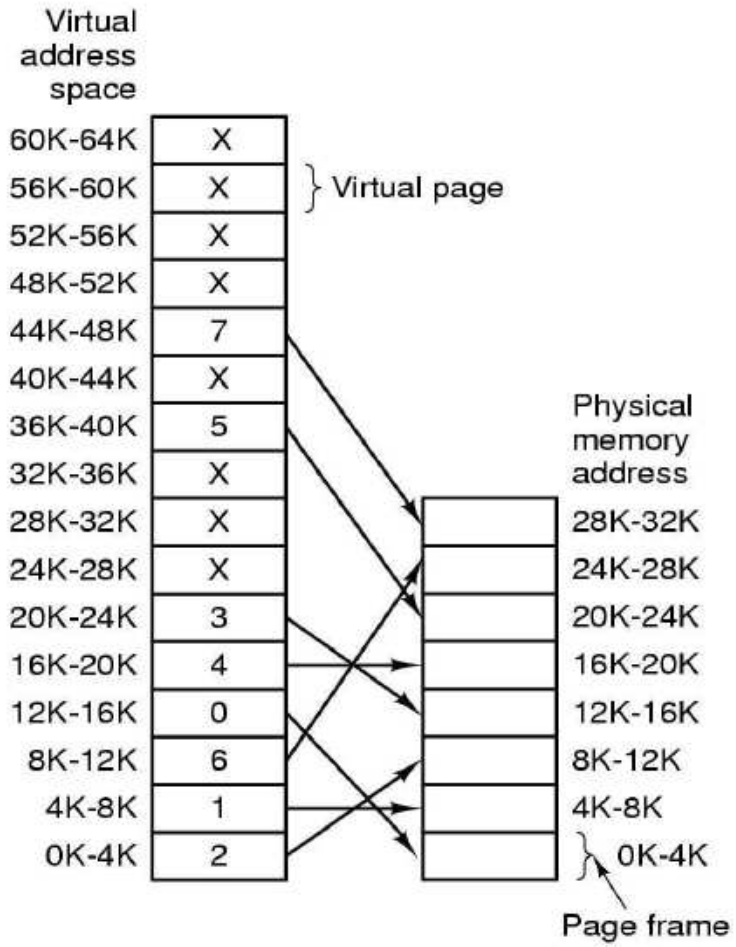
Deadlock-uri



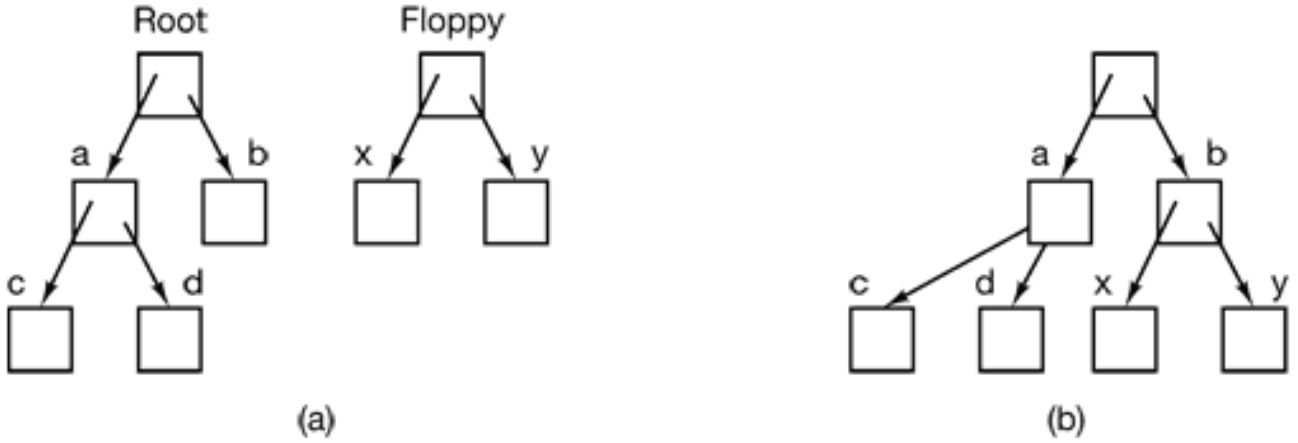
- deadlock potențial (a)
- deadlock (b)

Memorie virtuală

- adrese
  - virtuale
  - fizice
- MMU și SO fac translatarea din adrese virtuale în adrese fizice



- cale, director rădăcină, director de lucru
- descriptor de fișier / handle
- fișiere speciale
  - bloc
  - caracter
  - pipe-uri
  - link-uri



- sistem de fișiere, operații de montare/demontare
  - (a) înainte de montare
  - (b) după montare

Interpretorul de comenzi

- Programul de interfață între utilizator şi sistem
- Mod linie de comandă
  - avantaje: flexibilitate, puține resurse consumate
  - ex: sh , bash, korn
- Mod grafic
  - avantaje: mai ușor de utilizat
  - ex: explorer.exe în Windows

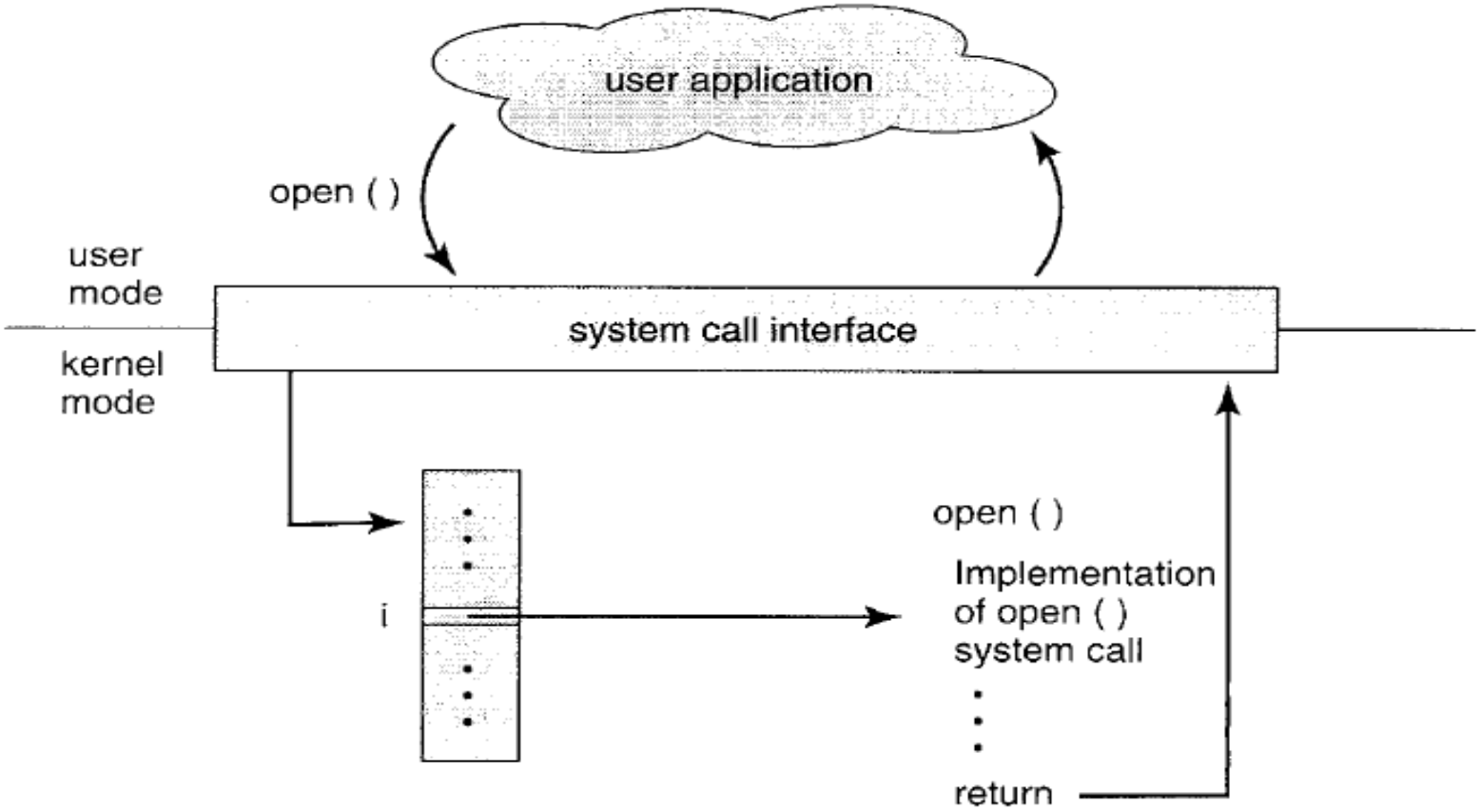
Nucleul sistemului de operare

- Nucleul SO are acces direct la hardware
- Părți din nucleu sunt permanent rezidente în memorie
- Imaginea nucleului
  - Linux: /vmlinuz, /boot/vmlinuz
  - Windows: %SystemRoot%\system32\ntoskrnl.exe
  - Mac OS X: Mach 3.0 + *BSD
- module / drivere

Kernel vs. User

- Nucleul rulează în mod privilegiat
  - kernel mode
  - kernel space
- Orice proces are un spațiu de adrese diferit
- Nucleul lucrează cu un spațiu de adrese diferit de cel al proceselor
  - user mode
  - user space

- Accesul la resursele sistemului se face prin apelarea serviciilor puse la dispoziție de nucleu



Componente ale SO

- gestiunea proceselor
- gestiunea memoriei
- gestiunea fișierelor
- gestiunea operațiilor de I/E
- gestiunea rețelei

Gestiunea proceselor

- crearea și terminarea proceselor
- suspendarea și repornirea proceselor
  - planificatorul de procese (scheduler)
- mecanisme de sincronizare
- mecanisme pentru comunicație inter-procese
- detectare/rezolvare deadlock-uri
- protecție

Gestiunea memoriei

- gestiunea memoriei fizice și virtuale
  - memorie virtuală, segmentare, paginare
  - swaping
- gestiunea memoriei folosite de nucleu
  - memorie rezidentă permanentă
  - memorie rezidentă temporară
- gestiunea spațiilor de adrese
  - malloc/free, mmap
- protecție

Gestiunea fișierelor

- translatarea operațiilor de acces asupra fișierelor (open, close, read, write, seek) în operații de citire și scriere pe disc
- caching și read-ahead
- compresie și criptare
- protecție

Gestiunea operațiilor de I/E

- interfață comună pentru device drivere
- caching, buffering
- întreruperi, DMA
- I/O scheduling

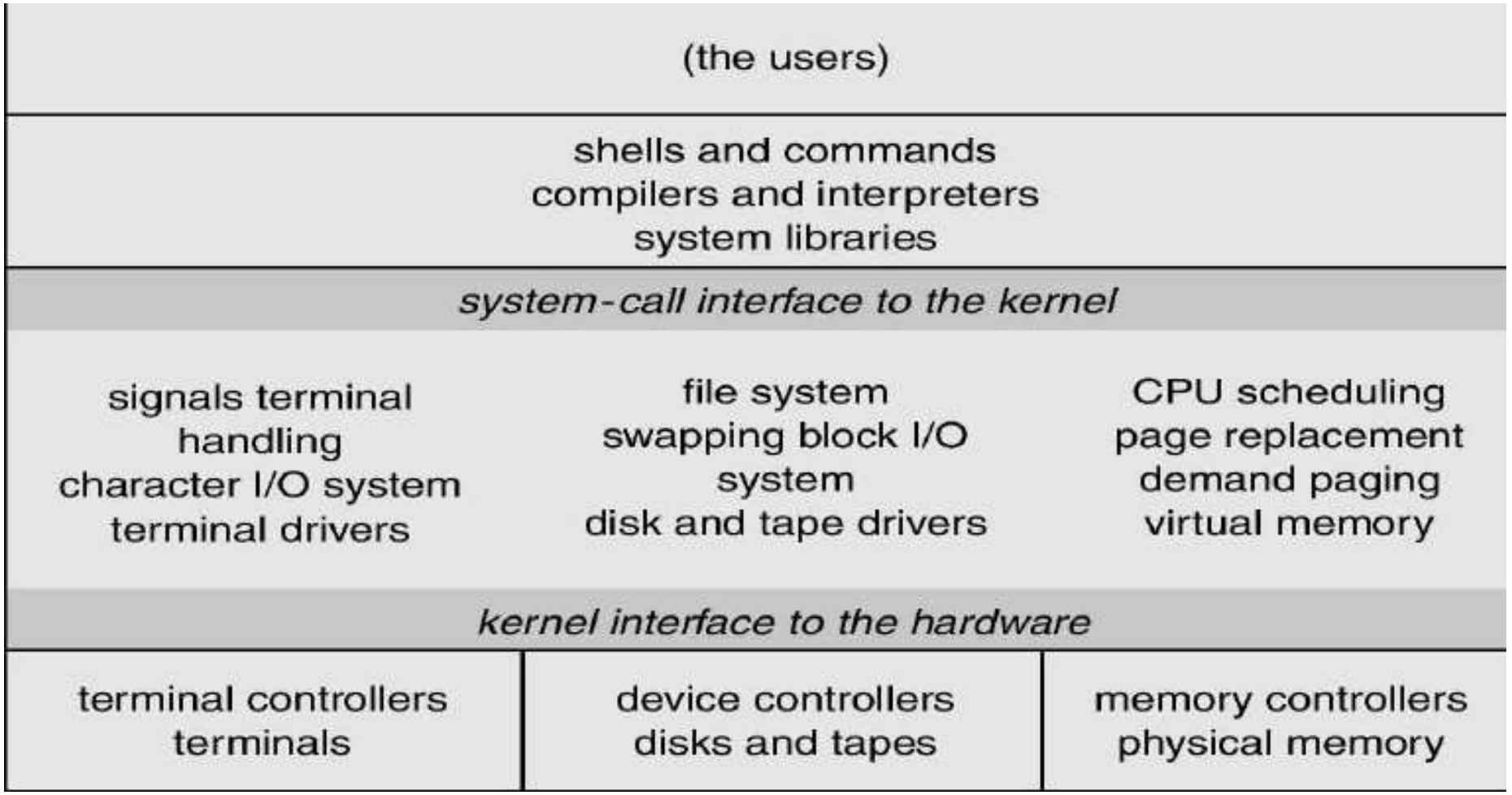
Gestiunea rețelei

- implementarea unor stive de protocol eficiente și sigure (secure)
  - implementare în kernel
- legături cu alte subsisteme ale SO: memorie, procese, scheduling, sisteme de fișiere

Structuri de SO

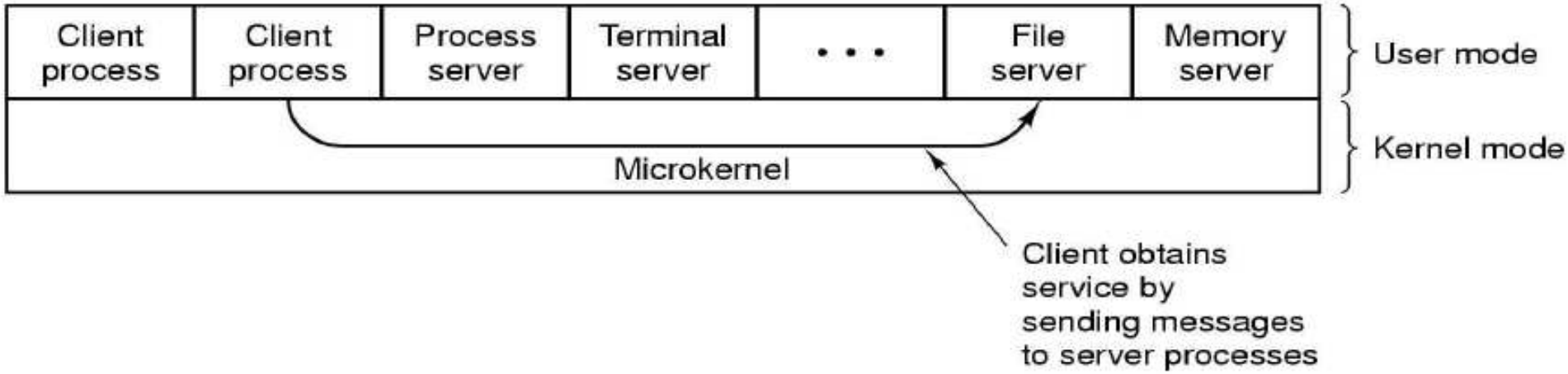
- SO monolitice
- SO microkernel
- Mașini virtuale
- SO exokernel
- SO stratificate

SO monolitice



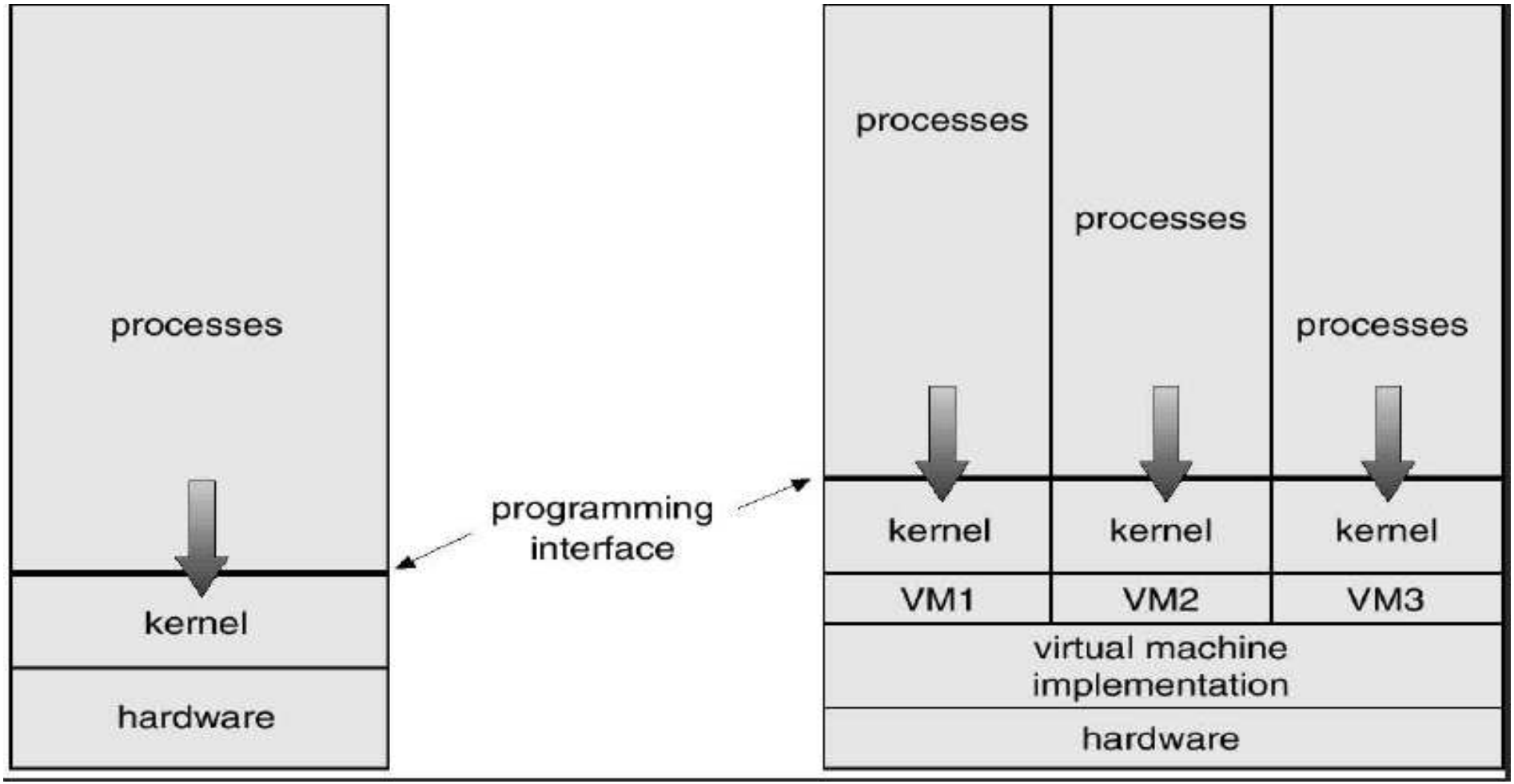
- UNIX și derivatele, Windows

Microkernel



- Minix, Amoeba
- QNX
- Mach

Maşini virtuale



Cuvinte cheie

- Sistem de operare (SO)
- Generații de SO/sisteme de calcul
- Unix
- Procesor
- Memorie
- Cache
- Întreruperi
- DMA
- Procese
- Fire de execuție
- Fișiere
- kernel/nucleu
- Apel de sistem
- Monolitic, microkernel
- Mașini virtuale

Intrebări

?