

# Introducere

SO: Curs 1

# Cuprins

- De ce (un curs) de SO?
- Despre cursul de SO
- Despre sisteme de operare
- Sisteme de calcul. Hardware
- Concepte importante în SO
- Structura unui SO

**DE CE (UN CURS DE) SO?**

# Un mic joc

- Scrieți pe o foaie de hârtie o aplicație (desktop, mobile, web) pe care o folosiți sau care vi se pare utilă/interesantă
- Participați la discuția cu titularul de curs

# Sisteme de operare

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

# De ce sisteme de operare?

- Independent de job/limbaj de programare/tip de aplicație
  - Depanare
  - Performanță
  - Securitate

# De ce sisteme de operare? (2)

- Sunt peste tot
- Gestiune eficientă
- Aplicații performante, eficiente
- Folosirea resurselor sistemului
- Folosirea interfețelor SO
- Înțelegerea sistemului, modului de funcționare
- Programare de sistem, programare low-level, programare embedded
- Software engineering: conexiuni, bune practici
- Concepte: structuri de date complexe, concurență, multithreading, I/O

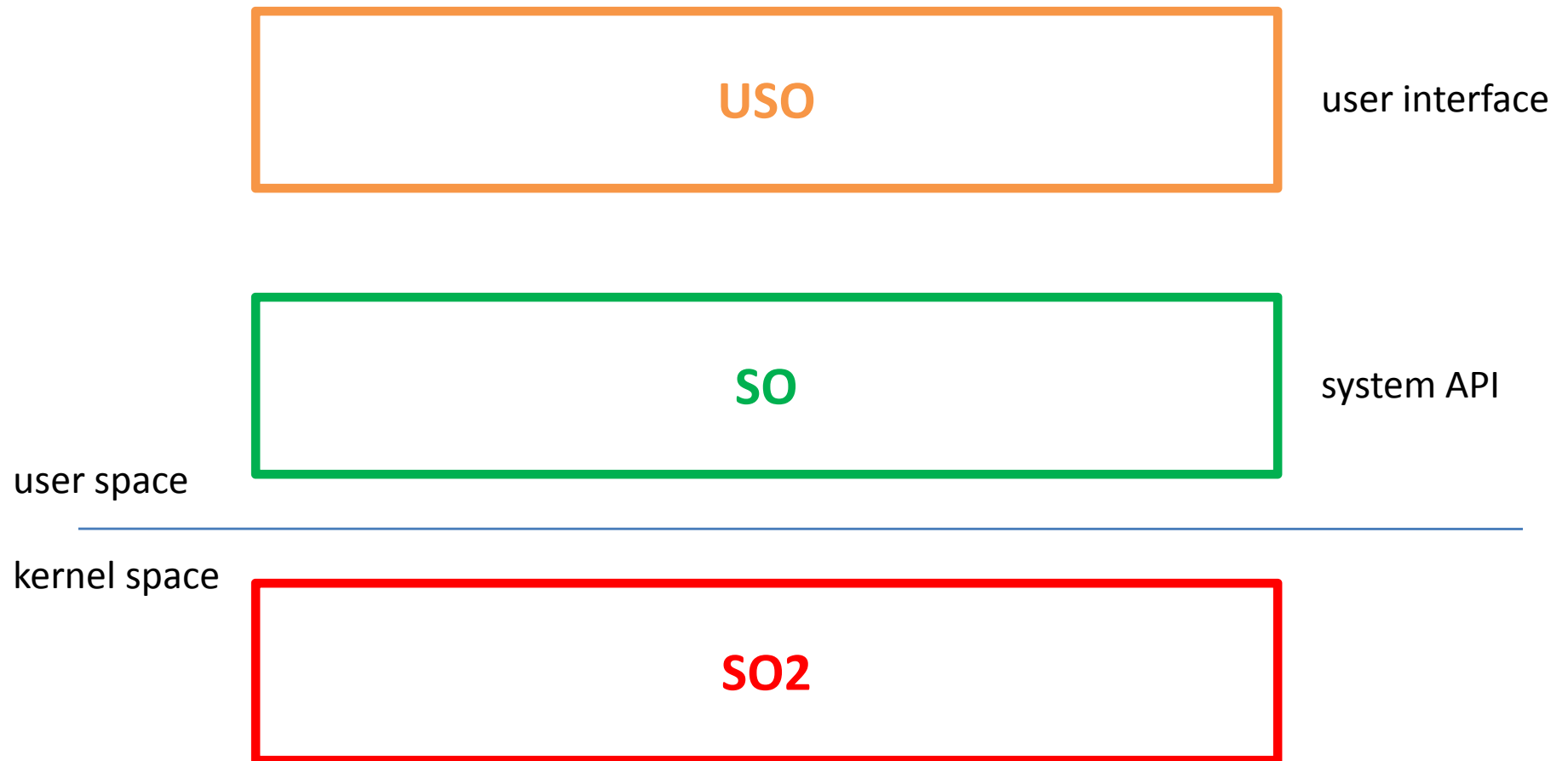
# De ce un curs de SO?

- Conceptele de bază în lumea sistemelor (systems)
- Cunoașterea funcționării sistemelor pentru a fi un bun profesionist în domeniu
- Realizarea de corelații între concepte, extragerea aspectelor relevante într-o situație
- Tehnici avansate de programare
- Folosirea unei interfețe de programare standard (system API)
- Acomodare cu realizarea de programe mai complexe



# **DESPRE CURSUL DE SO**

# Cursuri de SO



# Legături cu alte discipline

## **Cunoștințe utile din**

- USO
- Programare
- SD
- IOCLA
- CN
- PC

## **Cunoștințe utile pentru**

- SO2
- CPL
- SI
- APP
- SPRC

# Cine suntem

- Curs
  - Costin Raiciu, Marius Zaharia, Răzvan Deaconescu
- Laborator
  - Flavius Anton, Mihai Bărbulescu, Costin Carabaș, Mihai Carabaș, Cristian Condurache, Sergiu Costea, Răzvan Crainea, Roberta Dobrescu, Laura Gheorghe, Costin Lupu, Adrian Stanciu, Laura Vasilescu
- Colaboratori
  - Vladimir Diaconescu, Alexandra Săndulescu

# Resursele cursului de SO

- echipa
- wiki: <http://ocw.cs.pub.ro/courses/so/>
- listă de discuții
- slide-uri cursuri
- catalog și calendar Google
- mașini virtuale
- interfața vmchecker
- documentație
- pagina de Facebook

# Comunitatea SO

- întrebări, discuții, sugestii, feedback
- live (cursuri, laboratoare), listă, Facebook
  - să parcurgeți indicațiile de comunicare pe listă
- colaborare în timpul laboratorului
  - încurajăm să discutați și vă ajutați între voi

# Conținut

## Cursuri

- 13 cursuri
- slide-uri
- suport bibliografic
  - foarte important de parcurs
- bogat în informații
- interactiv

## Laboratoare

- 12 laboratoare
  - corelate cursurilor
  - după curs
- practice
- joc, discuții, exerciții
- tutorial, task-based, learn by doing
- încurajăm colaborarea

# Teme

## Conținut

- Cross-platform development
- Mini-shell
- Demand pager/swapper
- Planificator de thread-uri
- Server de fișiere

## Dezvoltare

- Linux și Windows
- aprofundare API
- 8-20 de ore pe temă
- teste publice
  - testare pe vmchcker
- întrebări și suport pe listă



# Despre rezolvare teme

- Temele te ajută să înțelegi mai bine conceptele
- Temele îți dezvoltă abilitățile de programare
- Pot fi dificile, dar e “rewarding” când îți ies
- Temele sunt **individuale**
  - Puteți discuta între voi în limitele bunului simț
  - Întrebați cu încredere pe lista de discuții
  - Parcurgeți regulile de realizare de teme
- Folosiți GitLab sau BitBucket sau GitHub (privat)

# Bibliografie

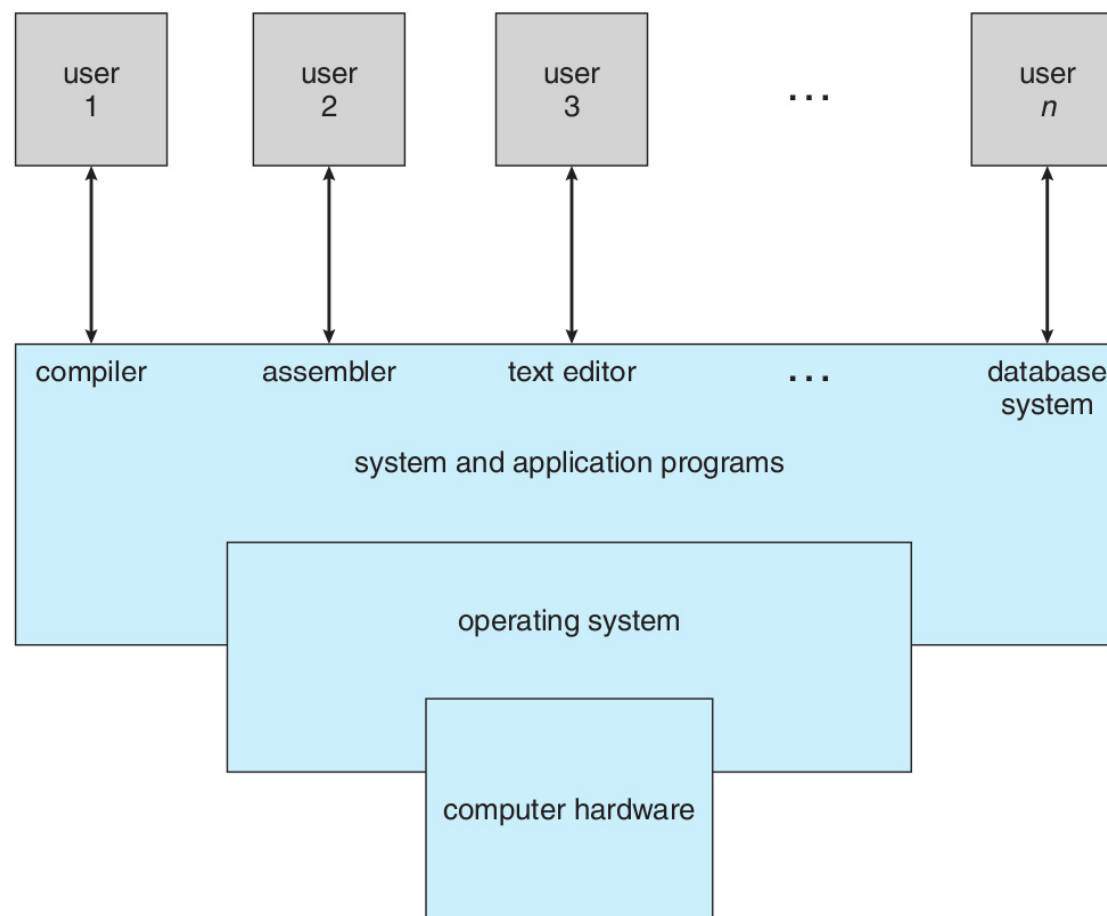
- Curs
  - Operating System Concepts
  - Modern Operating Systems
- Laborator
  - The Linux Programming Interface
  - Windows System Programming

# Bibliografie suplimentară

- Beginning Linux Programming
- Advanced Programming in the Unix Environment
- Linux System Programming
- Win32 Programming
- Programming Windows
- Linkers and Loaders

# **DESPRE SISTEME DE OPERARE**

# Ce este un SO?



OSCE, fig. 1.1, pg. 4

# Ce este un SO?

- un set de programe
- vedere top-down: extensie a mașinii fizice
- vedere bottom-up: gestionar al resurselor fizice
- scris în general în C
- relativ transparent utilizatorului (“trebuie să meargă”)

# Istoria SO

- Strâns legată de evoluția hardware
- UNIX, 1970, PDP-11
- CP/M, MS-DOS, VAX VMS, 1980
- Versiuni de UNIX: Solaris, HP-UX, AIX, anii '80-'90
- Windows: domină zona desktop anii '90, 2000 ...
- Windows, Mac OS X, Linux, \*BSD: desktop + servere moderne (virtualizare, cloud)
- Android, iOS, Windows Mobile: mobile
- Linux, RTOS-uri: sisteme embedded, IoT

# Legătura SO - hardware

- SO este primul nivel software peste hardware
- Un SO performant folosește facilitățile hardware
- Un SO performance 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



# **SISTEME DE CALCUL. HARDWARE**

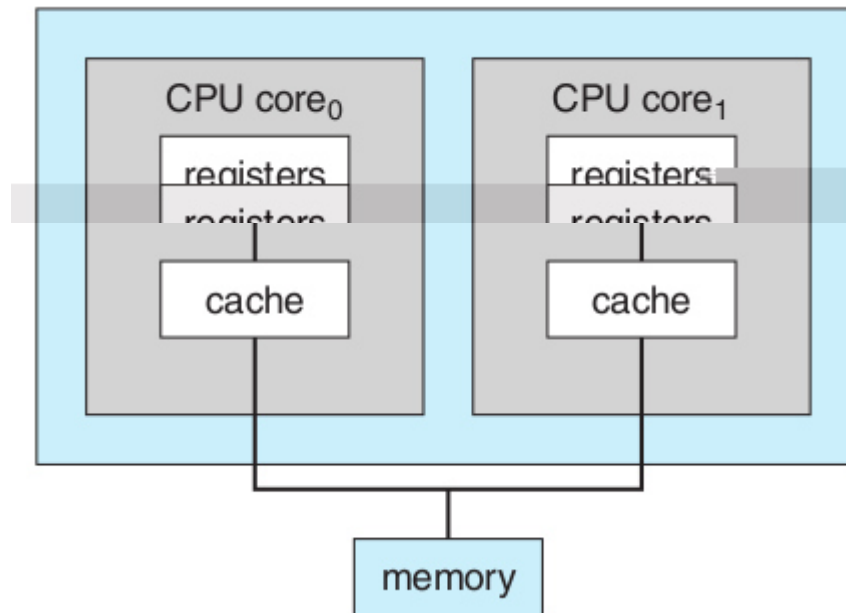
# Sistem de calcul

- Sistemul fizic (hardware) + sisteme de operare și aplicații
- Baza pentru a dezvolta aplicații, a folosi aplicații, a construi sisteme
- În general interconectat cu alte sisteme
- De la servere cu mii de core-uri la dispozitive mici din zona IoT (Internet of Things)

# Hardware-ul unui sistem de calcul

- Procesor (CPU)
  - rulează codul (instrucțiunile) proceselor
- Memorie de lucru (RAM)
  - stochează datele și codul proceselor
- Magistrale
  - leagă CPU, module de memorie, dispozitive de I/E
- Dispozitive periferice (de intrare/ieșire, I/E, I/O)
  - comunicarea cu exteriorul: utilizator, alte sisteme de calcul, alte dispozitive
- Spațiu de stocare (disc-uri, flash, ROM, NVRAM)
  - Programe (din care vor lua naștere procese)
  - Date pentru procese
  - Informații pentru utilizator (fișiere)

, /

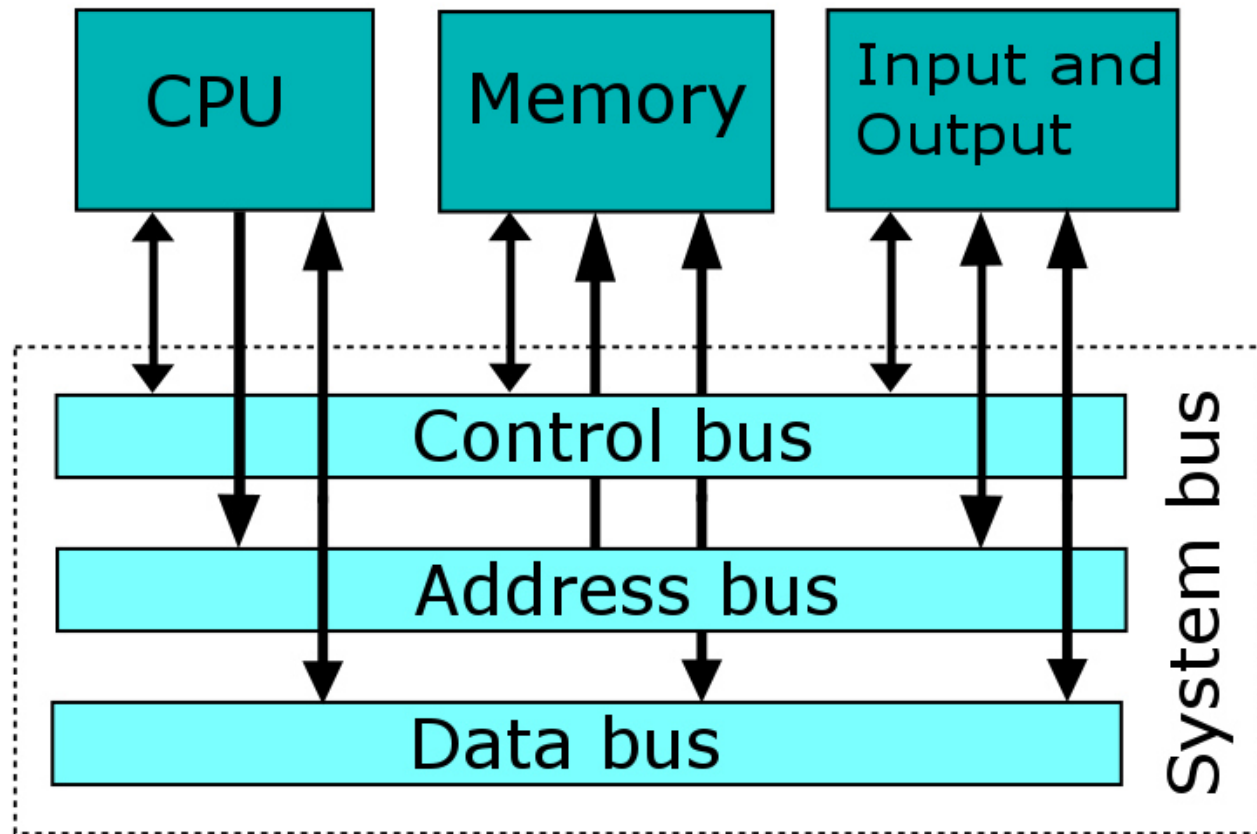


l' # 1 9' 1 3

# Memorie

OSCE, fig. 1.11, pg. 28

# Magistrale



[https://en.wikipedia.org/wiki/Bus\\_%28computing%29](https://en.wikipedia.org/wiki/Bus_%28computing%29)

# **CONCEPTE IMPORTANTE ÎN SO**

# Shell

- Interfața utilizator-sistem de operare
- CLI sau GUI
- Bash vs. Windows Explorer / GNOME / KDE
- Un proces care permite pornirea de alte procese/aplicații
  - “shell spawns a process”

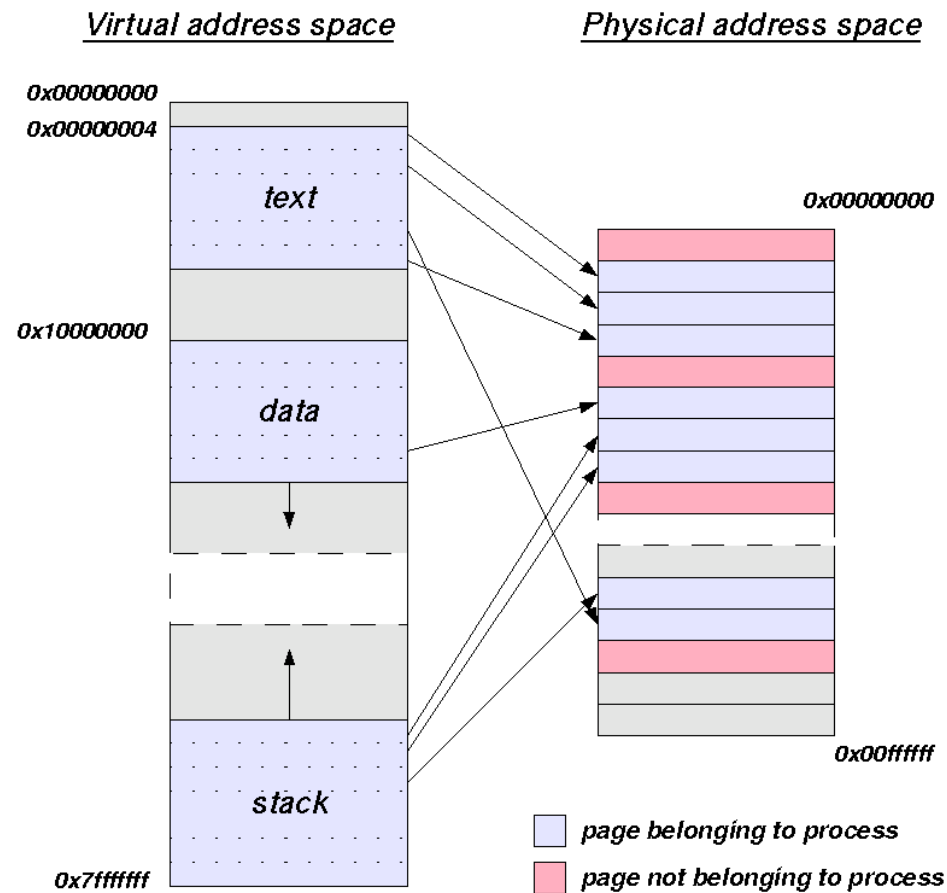


# Procese

- 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

Mai multe la cursurile 3 și 4

# Memoria unui proces



Mai multe la cursurile 3 și 5

[https://en.wikipedia.org/wiki/Virtual\\_address\\_space](https://en.wikipedia.org/wiki/Virtual_address_space)

SO: Curs 1: Introducere

# Memorie 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

Mai multe la cursurile 6 și 7

# 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

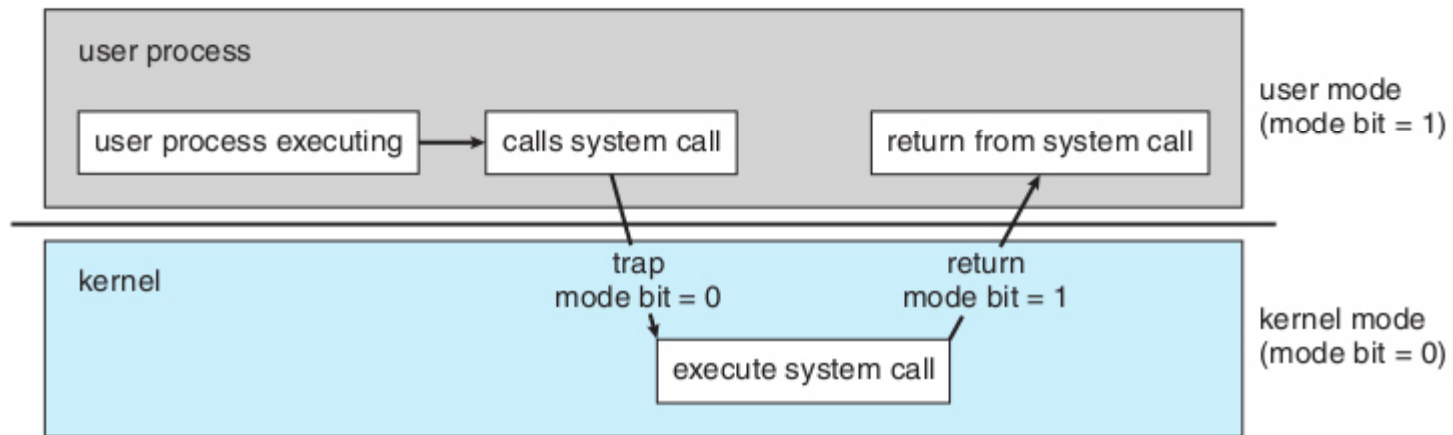
- Thread-urile sau procesele pot concura la achiziția de resurse (date în memorie)
- Accesele concurente pot duce la date inconsecvente sau blocaje
- Sincronizarea garantează accesul consecvent și ordonat la date

# Moduri de execuție

- Procesorul are un mod privilegiat: supervisor mode, system mode, kernel mode
  - Pentru acțiuni privilegiate
  - Aici rulează sistemul de operare
- Aplicațiile rulează în modul neprivilegiat: user mode
  - Nu pot comunica cu hardware-ul sau cu alte procese
  - Pot doar acționa asupra spațiului de memorie propriu
- Nucleul intermediază accesul proceselor la hardware sau la resursele altor procese
- Separația modurilor asigură securitatea sistemului
- Tranziția user mode -> kernel mode = apel de sistem

Mai multe la cam toate cursurile următoare ☺

# Traziția user mode – kernel mode

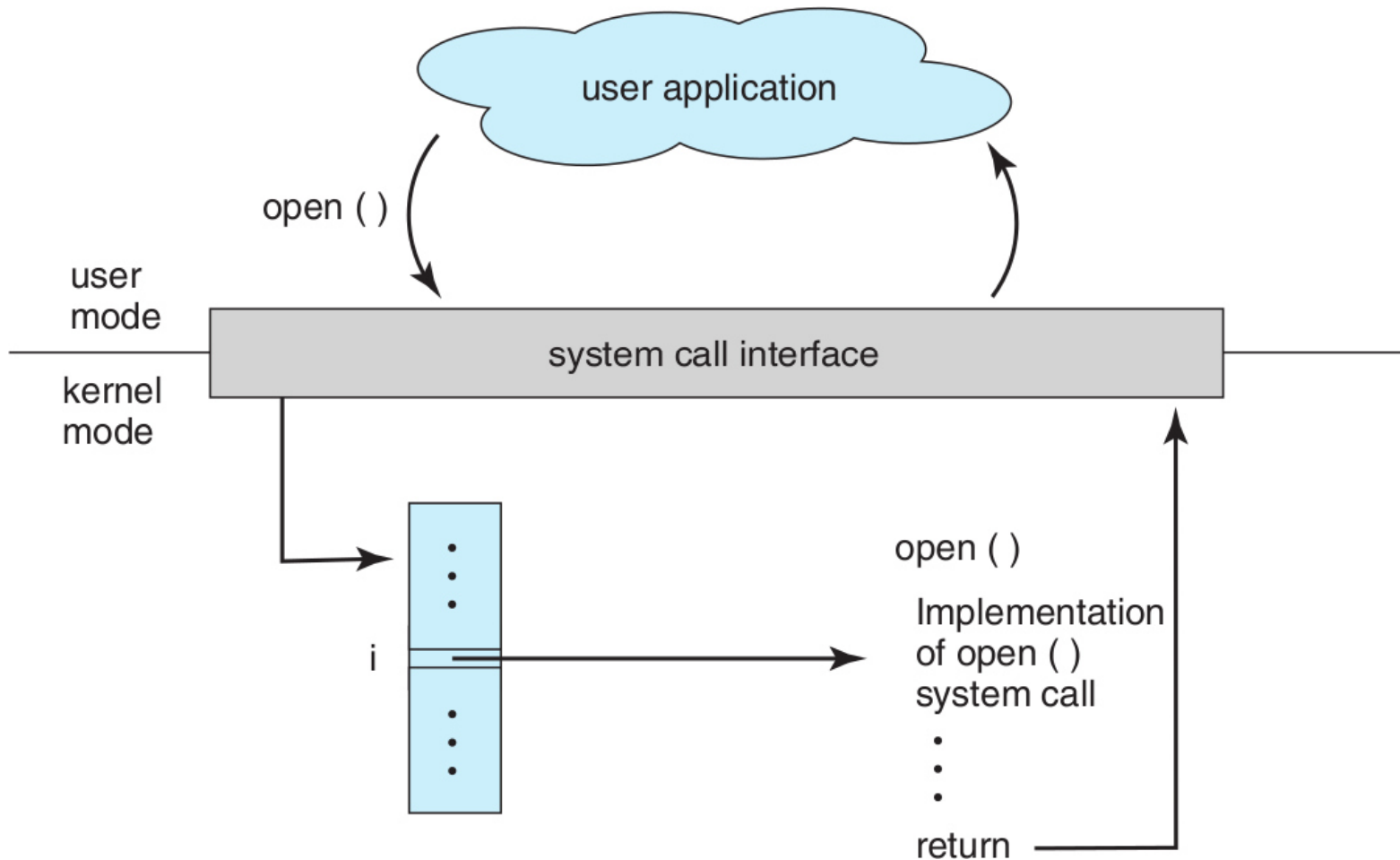


OSCE, fig. 1.10, pg. 22

Mai multe la cam toate cursurile următoare 😊

SO: Curs 1: Introducere

# Apeluri de sistem



Mai multe la cam toate cursurile următoare 😊

SO: Curs 1: Introducere

OSCE, fig. 2.6, pg. 56

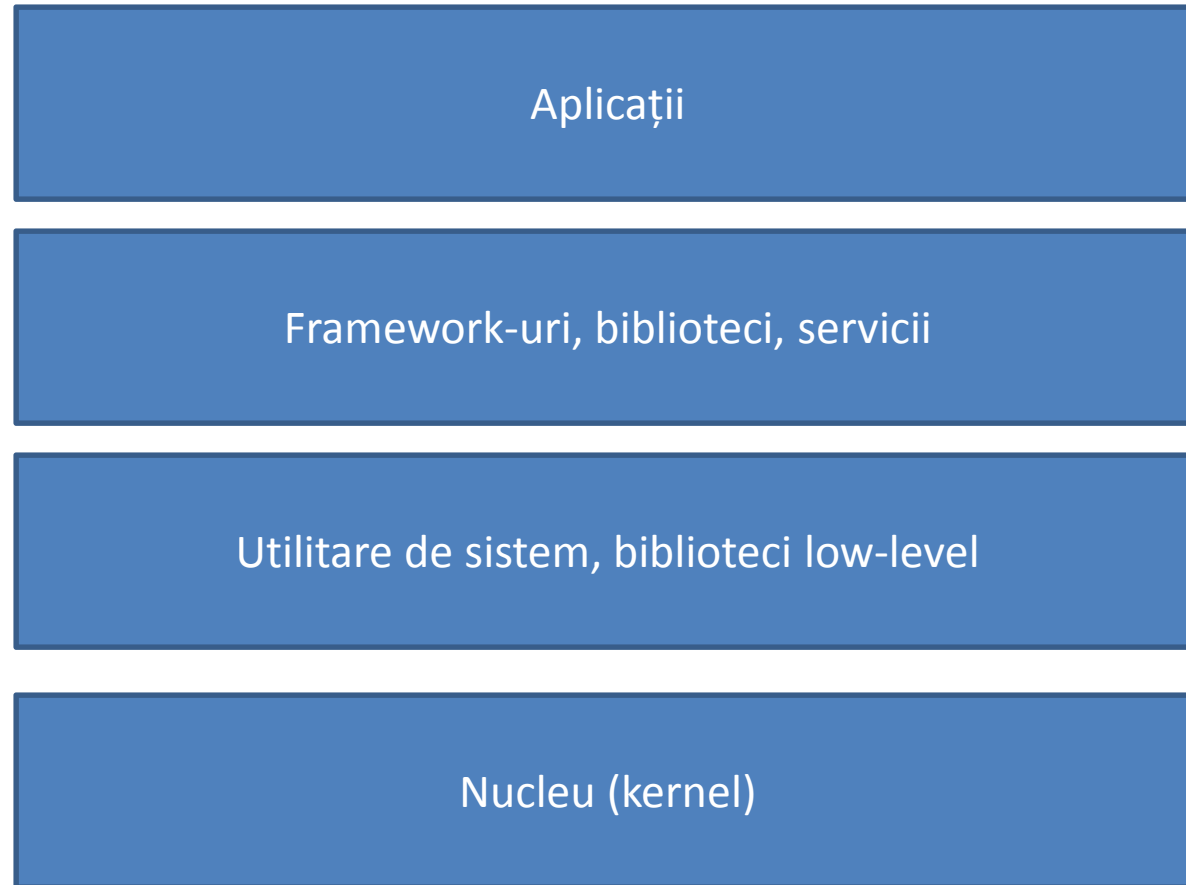


# Fișiere

- 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

# STRUCTURA UNUI SO

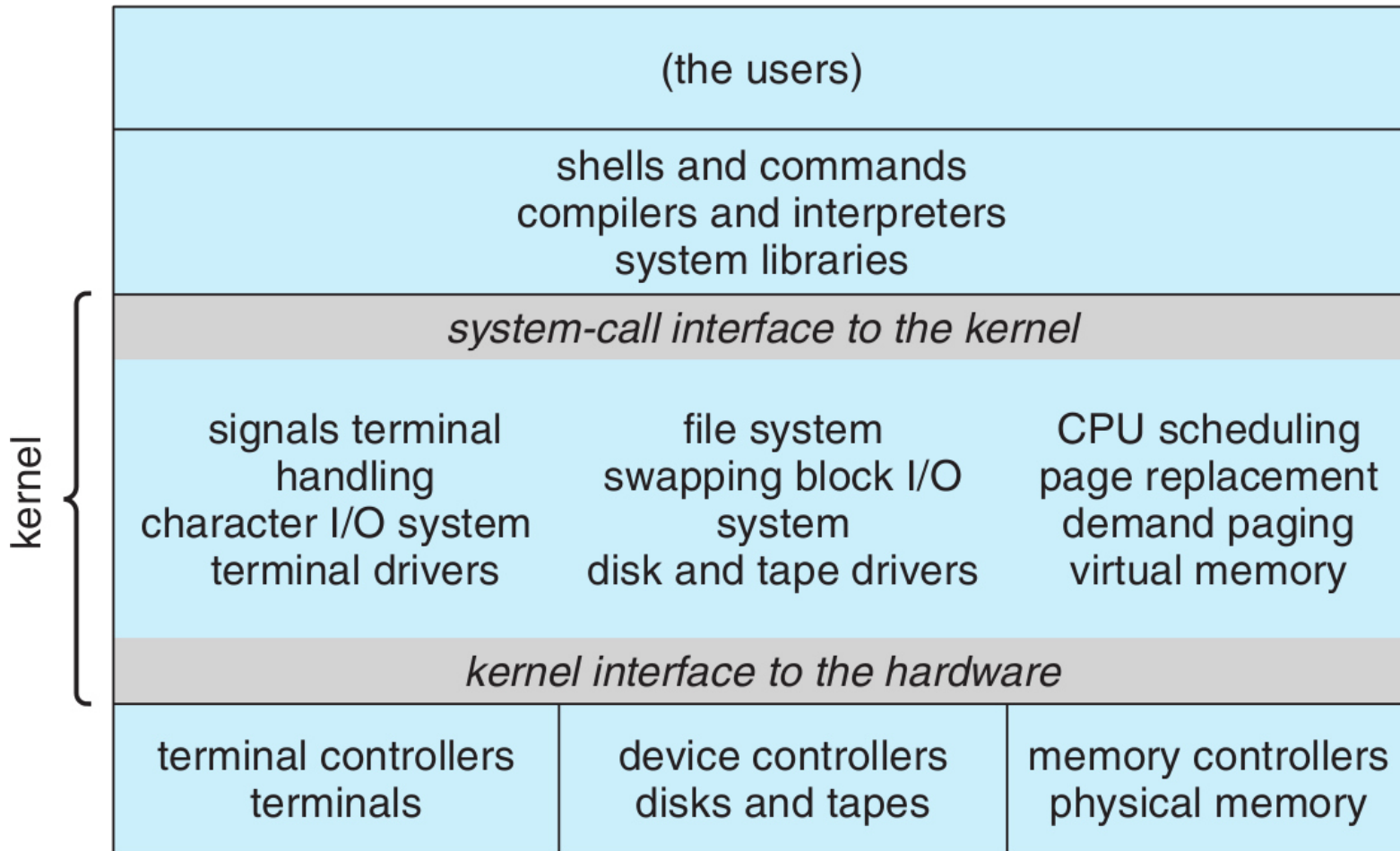
# Stiva software pentru un sistem de calcul



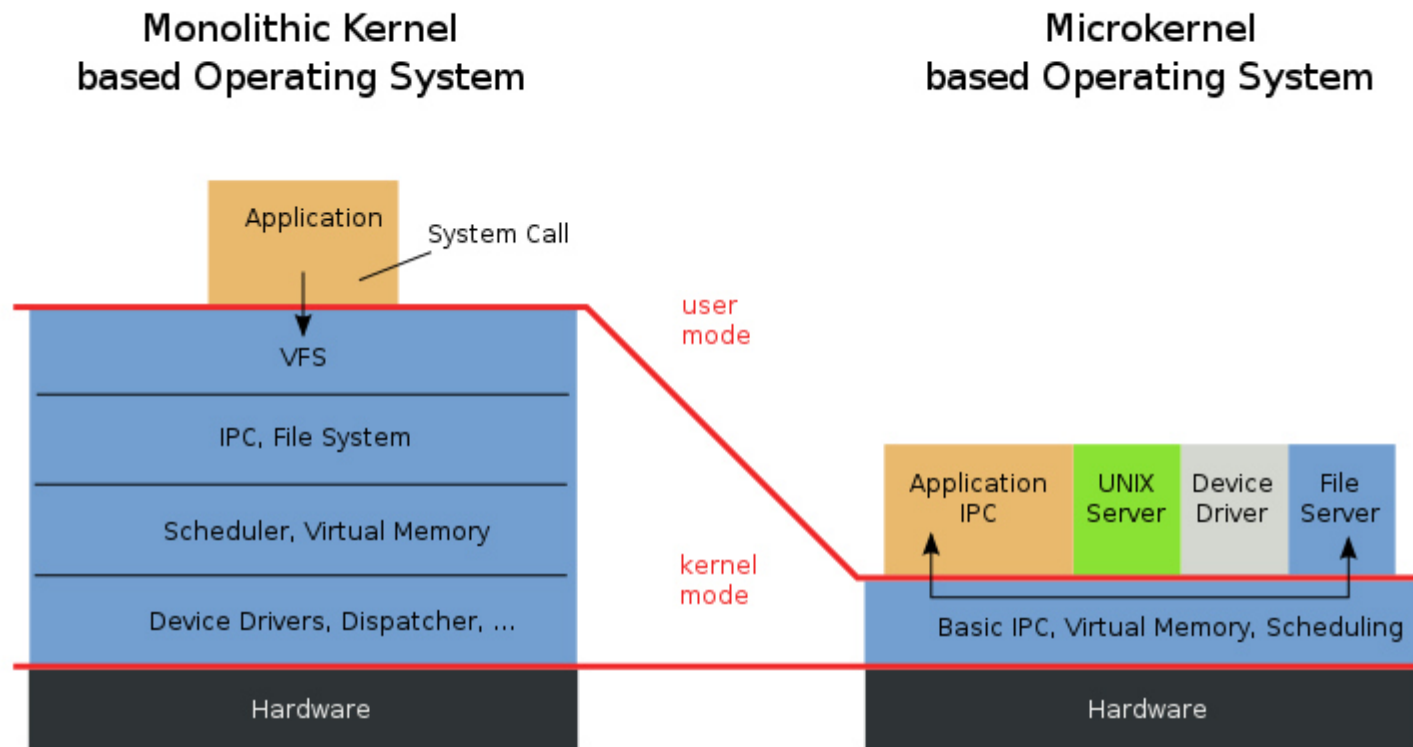
# Nucleul (kernel-ul)

- Strict tehnic, nucleul este sistemul de operare
  - Windows are ca nucleul WindowsNT din imaginea ntoskrnl.exe
  - Linux este nucleul unei distribuții GNU/Linux și Android
  - Mac OS X și iOS auca nucleu XNU
- Încărcat la bootare apoi pornește primele aplicații și gestionează hardware-ul
- Răspunde apelurilor de sistem ale proceselor
- Gestionează resursele hardware
- Garantează integritatea sistemului

# Structura uzuală a SO (monolithic)



# SO monolithic vs. SO microkernel



<https://en.wikipedia.org/wiki/Microkernel>

# SO monolithic vs. SO microkernel (2)

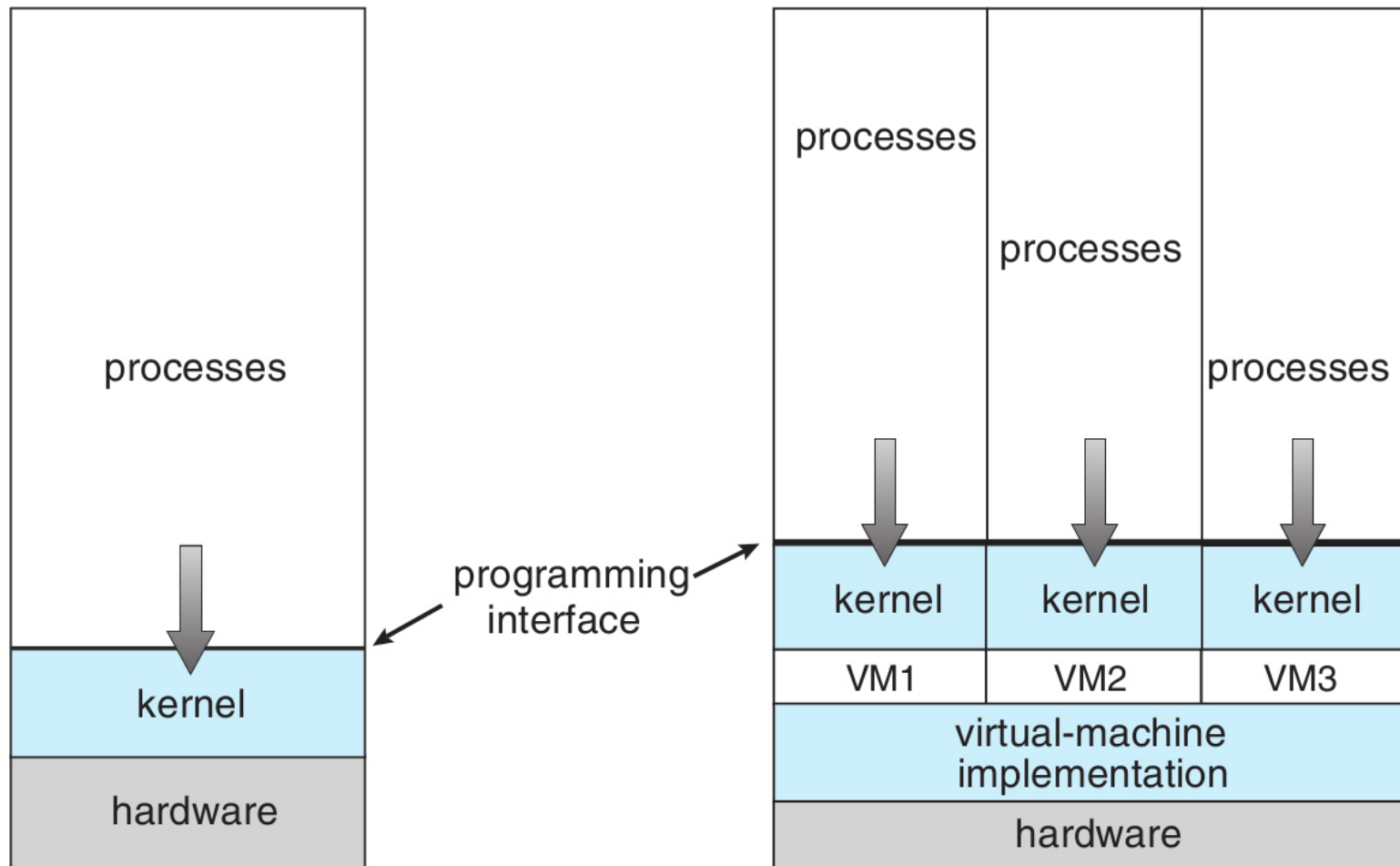
## **Monolithic**

- Eficient
- Coeziunea codului/datelor
- Mai puțin flexibil
- TCB (Trusted Computing Base) mai mare (design mai puțin sigur)

## **Microkernel**

- Mai lent (comunicare între servicii)
- Componentizabil, flexibil, modular
- TCB redus (design mai sigur)

# Maşini virtuale





# Tendințe curente în SO

- Securitate
- Dispozitive de mici dimensiuni (tinification)
- Scalare (CPU, memorie, disc), mașini virtuale
- Performanță, suport hardware pentru operații specifice

# Reminder: Resursele cursului de SO

- echipa
- wiki: <http://ocw.cs.pub.ro/courses/so/>
- listă de discuții
- slide-uri cursuri
- catalog și calendar Google
- mașini virtuale
- interfața vmchecker
- documentație
- pagina de Facebook

# Concluzie

- Cunoștințele de SO sunt utile pentru
  - securitate
  - depanare
  - performanță
- Conceptele de SO sunt relativ complexe
- Sunt foarte importante legăturile între concepte
- Vom insista pe fiecare concept în următoarele cursuri
- Sistemul de operare este strâns legat de hardware
- Nucleul SO gestionează hardware-ul și asigură securitatea sistemului

# Cuvinte cheie

- sistem de operare (SO)
- comunitatea SO
- sistem de calcul
- hardware
- procesor
- memorie
- procese
- spațiu virtual
- memorie virtuală
- mod de execuție
- kernel mode
- user mode
- apel de sistem
- fișiere
- nucleu/kernel
- SO monolitic
- SO microkernel
- mașină virtuală