

Sisteme de Operare 1 - Curs 1

Curs tinut in 2012-2013 de catre lector dr. Sanda-Maria Dragos

slide 2



slide 3



slide 4



slide 5



slide 6



slide 7



Ce este un Sistem de Operare?

Un Sistem de Operare este o interfata intre echipamentele hard (hardware) si aplicatii.
- actioneaza ca o "gazda" pentru aplicatii, preluand detaliile operatiilor cu echipamentele hard.

Caculatorul poate fi privit ca avand 3 componente principale: echipamentele hard, sistemul de operare si aplicatiile.

Echipamentele hard este tot ce e tangibil la un calculator: hard-disc-ul, procesorul, monitorul, tastatura, etc.

Aplicatiile sunt (motivul pentru care folosim calculatorul) folosite pentru diferite operatiuni (de ex. a juca un joc, a edita un documents sau a trimite un e-mail).

Sistemul de operare este componenta care pe de o parte gestioneaza si coordoneaza echipamentele hard, iar pe de alta parte gestioneaza aplicatiile (joaca rol de gazda pentru acestea).

Sistemul de operare contine rutine care se ocupa de initializare echipamentelor hard si accesare acestora de catre aplicatii, precum si rutine care se ocupa de executarea, cererea si deservirea resurselor pentru aplicatii, cat si terminarea executiei acestora.

slide 8



Scurt istoric al sistemelor de operare

slide 9

anii '40 - prima generatie de calculatoare

- » nu aveau SO
- » programele erau introduse ca si siruri de biti cu ajutorul unor comutatoare mecanice

anii '50 - a doua generatie de calculatoare

- » a aparut primul SO implementat de *General Motors Research Laboratories* pentru IBM 701
- » SO executau o singura operatie la un moment dat
- » erau numite "**single-stream batch processing**" deoarece programele si datele erau trimise in mod grupat spre procesare (pe cartela sau banda perforata)

anii '60 - a treia generatie de calculatoare

- » erau tot "**batch processing systems**" (gruparea proceselor astfel incat urmatorul incepe sa se execute imediat ce cel anterior s-a terminat), insa puteau rula mai multe operatii simultan fara interventie umana
- » conceptul de "**multiprogramming**"
 - » mai multe operatii se afla in memorie simultan
 - » un procesor trece de la operatie la alta pentru a le rula si a mentine perifericele functionale
- » conceptul de "**spooling**"
 - » transferarea datelor prin plasarea lor intr-un spatiu de lucru temporar spre a fi accesate ulterior. Este folositor datorita faptului ca perifericele acceseaza datele cu viteze diferite. Cea mai obisnuita aplicatie de spooling este tiparirea: folosirea simultana a perifericelor prin interpunerea unui dispozitiv cu viteza mare de acces ("high-speed"; de ex HDD-ul) intre un program in rulare si un dispozitiv cu viteza reduse de acces ("low-speed"; de ex imprimanta) pentru operatiile de intrare-iesire, prin folosirea unui buffer.
- » conceptul de "**time-sharing**"
 - » este o alta tehnica de multiprogramare prin folosirea unor timpi de asteptare (de ex. pentru accesarea HDD-ului sau al retelei) pentru executarea altor programe.

a patra generatie de calculatoare

- » reprezinta era dezvoltarii circuitelor LSI (Large Scale Integration) si a cipurilor, ceea ce a dus la aparitia calculatoarelor personale si a statiilor de lucru
- » doua SO au dominat scena calculatoarelor personale: MS-DOS, dezvoltat de Microsoft, Inc. pentru IBM PC si alte masini care foloseau procesoare Intel 8088 sau successorii acestora, si UNIX, care au fost initial folosite pe calculatoarele personale care foloseau procesoare din familia Motorola 68000.

slide 10

Sistemul de operare UNIX

Unix este un sistem de operare ce incepe sa devina popular. Initial a fost utilizat doar de catre comunitatea academica.

UNIX este un sistem de operare portabil si modular dezvoltat initial de un grup de programatori din cadrul AT&T Bell Labs, printre care si Ken Thomson, Dennis Ritchie si Douglas McIlroy.

Astazi, sistemele UNIX sunt dezvoltate de companii (precum IBM, HP sau SUN), indivizi si organizatii non-profit (precum Free Software Foundation).

Astazi, numele de UNIX nu mai reprezinta atat de mult numele unui singur sistem de operare, ci mai degraba numele unei clase de sisteme de operare compatibile desi nu impartasesc cod sursa comun, ci doar niste paradigme comune numite in general *filosofie unix*.

» SIMPLITATE, CLARITATE,
LONGEVITATE

» programe mici care fac un singur lucru BINE

» "small is beautiful"

» *apoi aceste programe simple pot fi combinate pentru a realiza lucruri mai complexe*

» CONECTIVITATE

» pentru a permite realizarea de sarcini complexe utilizand mai multe comenzi legate impreuna de utilizator, astfel iesirea unui program devine intrare pentru un alt program.

» conceptul de "**pipe**" inventat de Doug McIlroy: "Scrieti programe care fac un

slide 11

singur lucru dar il fac bine. Scrieti programe care comunica intre ele. Scrieti programe care folosesc siruri de text, pentru ca acestea sunt interfata universală."

» FLEXIBILITATE

» totul este un tratat ca un fisier (inclusiv procesele rulant sau dispozitivele fizice - de la tastatura pana la memoria RAM),

» Orice program e un filtru: produce un rezultat pentru anumite date de intrare.

» folosirea fisierelor text atat pentru configurare cat si pentru iesire,

» Nu exista tipuri de fisiere! UNIX nu face diferenta intre un fisier decat cand incerci sa rulezi un fisier. Continutul unui fisier e interpretat doar de comenzile care folosesc acel fisier. Asadar orice program poate accesa orice fisier, insa continutul fisierului poate fi descifrat doar de programele care inteleg formatul acelui fisier.

- » INTERACTIV: primeste comenzi de la terminal in regim conversational
- » TIME-SHARING: quante de timp alocate fiecarui proces in parte
- » MULTI-USER:
- » MULTI-TASK:

Familia sistemelor de operare UNIX

Sistemele UNIX actuale se bazeaza pe trei ramuri originale

1. UNIX System V (produs de cei de la AT&T la mijlocul anilor 1980),
2. BSD (ramura desprinsa din proiectul AT&T la sfarsitul anilor 1970) si
3. respectiv GNU/Linux (produs ca replica gratuita si open-source la celelalte doua ramuri - nucleul e dezvoltat de Linus Torvalds incepand cu 1991, iar o mare parte din aplicatii sunt date de Proiectul GNU).

slide 12



Fiecare distributie de astazi se bazeaza pe una din aceste ramuri, dar a preluat si anumite caracteristici de la celelalte.

Structura SO UNIX

UNIX include pe langa componentele traditionale a unui sistem de operare si un set de librarii si aplicatii. Multi cred ca aceste doua componente alcatuiesc SO UNIX, datorita faptului ca ele constituie interfata cu utilizatori.

Componenta UNIX care gestioneaza echipamentele hard si executa procesele se numeste **nucleu**.

Un alt element de FLEXIBILITATE e separarea kernelului de programele aplicatie etc. Tipic un sistem de tip Unix are o organizare pe trei nivele:

» *NUCLEU/KERNEL*: este baza sistemului de operare. Utilizatorul, prin intermediul nucleului, poate controla diverse componente ale calculatorului.

» *SHELL/Interpretor de comenzi*: este un intermediar intre utilizator si sistemul de operare (NUCLEU). Supravegheaza terminalele si preia comenzile utilizatorului. E un adevarat limbaj de programare.

» *APLICATII*

Componentele **nucleului** UNIX

- » SISTEMUL DE FISIERE (File System)
 - » Sistemul de fisiere nu face parte din Kernel! El se afla pe HDD! Insa toate rutinele care tin de gestiunea sistemului de fisiere (creare, modificare stergere, accesare fisiere sau directoare) se afla in nucleu.
 - » organizeaza si gestioneaza datele vehiculate in sistem si accesul la periferice (prin scrierea si citirea din fisiere)
 - » Implementeaza controale de securitate pentru a proteja siguranta datelor.
 - » UNIX - primul SO cu structura arborescenta de directoare
- » SISTEMUL DE GESTIUNE AL PROCESELOR (Process Control)
 - » gestioneaza procesele
 - » aloca resurse
 - » mediaza accesul la componentele hard

slide 13



slide 14



» CDEV, BDEV

» componente de legatura cu echipamentele hard/periferice

» CDEV - periferite de *tip caracter* (ex: tastatura, mouse-ul, terminale virtuale, modemuri seriale)

» BDEV - periferite de *tip bloc* (ex: HDD, drive-uri CD-ROM)

» Componente scrise in limbaj de asamblare

slide 15

Elemente de interfata exterioara UNIX

Structura unei comenzi, evitari si specificari generice

Sistemele de Operare de tip Unix ofera o mare diversitate de comenzi care presupun aceleasi reguli de scriere:



nume_comanda optiuni **expresii** fisiere

nume_comanda

numele comenzii

optiuni

un grup de una sau mai multe litere precedate de semnul - sau +. Fiecare litera reprezinta o optiune.

Pentru optiunile care accepta sa fie precedate de semnul + ceea ce reprezinta activare optiunii respective, semnul - reprezinta dezactivarea aceleiasi optiuni.

expresii

siruri de caractere folosite ca si argumente pentru comanda respectiva

fisiere

reprezinta unul sau mai multe fisiere specificate relativ sau absolut, sau folosind specificari generice.

```
ls -Rtl *.c ## -R - recursiv, -t - ordonat dupa data crearii
```

Orice comanda intoarce in urma rularii un COD DE RETUR (indicator de terminare).

= o la terminare cu succes

!= o la terminare cu insucces

Comenzilor UNIX le sunt asociate trei fisiere:

fisierul standard de intrare (stdin) 0

are ca input ceea ce s-a intriodus de la tastatura

fisierul standard de iesire (stdout) 1

avand ca output monitorul consolei de la care a fost lansata comanda

fisierul standard de eroare (stderr) 2

avand ca output monitorul consolei de la care a fost lansata comanda

slide 16



Aceasta asociere implicita a fisierelor standard cu monitorul si tastatura poate fi modificata

(redirectionata) putandu-se utiliza ca input/output un fisier specificat de utilizator, astfel:

comanda < fin

datele de intrare pentru comanda vor fi preluate din fisier

comanda > fout

comanda >> fout

iesirea standard va fi depusa in fisier prin rescriere, respectiv adaugare/concatenare

comanda 2> fout

comanda 2>> fout

erorile vor fi depuse in fisier prin rescriere, respectiv adaugare/concatenare

comanda1 | comanda 2

iesirea standard pentru prima comanda constituie automat intrare standard pentru cea de-a doua comanda

```
grep "ceva" A | sort > B;    ls -l *.c > A; sort < A > B
sort < A > B;               ls -Rl *.c | sort > B
```

Moduri de rulare: foreground, background

```
sort < A | sort > B &
cp C D &
gcc E.c
```

Numele unui fisier Unix

Un fisier Unix poate fi specificat printr-un sir de oricate caractere ASCII. Se recomanda folosirea literelor mari si mici, a cifrelor zeimale si/sau a caracterelor '.', '_'. Caracterul '.' poate sa apara oriunde si de ori cate ori. ATENTIE: sistemele de tip UNIX fac distinctie intre literele mari si cele mici!

Fiecare disc UNIX are un director principal/radacina/**root** din care pot descinde alte directoare si fisiere obisnuite. Din fiecare director pot descinde alte directoare s.a.m.d. Calea catre un fisier se poate specifica astfel:

```
[ / ] dir1/dir2/.../dirN/nume_fisier
```

specificare absoluta

incepand de la directorul radacina

specificare relativa

relativa la directorul curent

specificarea generica

*foloseste urmatoarele sintaxe (a se vedea la expresii regulate): *, ?, [sir], [!sir]*

Diferente intre MS-DOS si UNIX

	MS-DOS	UNIX
separator de cale	\	/
la PATH	;	:
sep. de linii in fis text	CR/LF	LF

Separarea zonelor comenzii se face prin spatii. Daca argumentul trebuie sa contina el insusi spatii sau orice alt caracter special, atunci expresia corespunzatoare lui se scrie intre apostrofuri sau ghilimele, astfel:

- » Daca in sir nu apare ' sau ", atunci delimitare se face cu '...' sau "...";
- » Daca in sir apare ' dar nu apare ", atunci delimitarea se face cu "..."; (*ex: echo "Tom's books"*)
- » Daca in sir apare " dar nu apare ', atunci delimitarea se face cu '...';
- » Daca in sir apare atat ' cat si ", atunci trebuie folosit caracterul de evitare \.

```
#Tom's "books". se va scrie "Tom's \"books\"."

echo "Tom's \"books\"."

grep "Tom's \"books\"." myfile
```

Componenta *shell* da o interpretare speciala caracterelor:

* ? < > . | \$ & ; \ []

Aceste caractere pot fi specificate intr-un sir prin folosirea caracterului de evitare \.

Primele comenzi UNIX


passwd

Se schimba parola. Daca comanda este folosita de superuser se va folosi nume_utilizator.

Terminarea unei sesiuni de lucru

```
#<CTRL/d>
exit
logout
```

slide 18



man

Deschide manualul pentru comanda respectiva

more

Afisare in mod paginat (cu oprire la fiecare umplere de ecran) a fisierului. \$<ENTER> - face trecerea la pagina urmatoare, **b** - face trecerea la pagina precedenta, **q** - termina listarea.

-n
*modifica la **n** dimensiunea implicita de 23 de linii a ecranului afisat;*

+n
*afisarea incepe cu cea de-a **n**-a linie a fisierului*

+/sablon
afisarea incepe de la linia care contine sirul de caractere specificat

more +100 fis

more +/331 fis

more +/nr=\"331\" fis