

Sisteme de operare 1

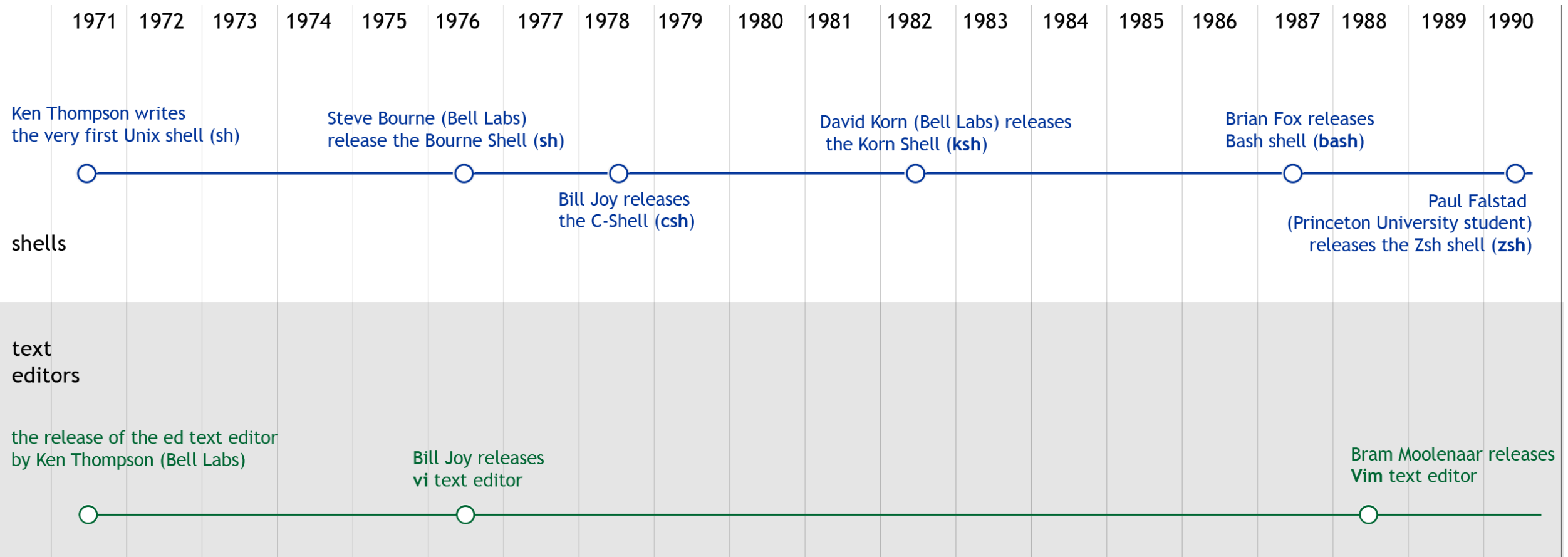
Curs 2

Titular curs,
Dr. Dragoş Sanda Maria

SHELL

- Principala interfață Unix-utilizator
 - Limbaj de comandă
 - Interpretor de comenzi
 - Limbaj de programare
 - Concepte de variabilă, constantă, expresie, structuri de control și subprogram

Interpretoare de comenzi *shell*



Cum functioneaza Shell?

CâtTimp (nu s-a închis sesiunea)

- Afișează prompter;

- Citește linia de comandă;

- Dacă (linia se termina cu '&') atunci

 - Creează un proces și-i dă spre execuție comanda

 - Nu așteaptă ca execuția să se termine

- Altfel

 - Creează un proces și-i dă spre execuție comanda

 - Așteaptă ca execuția comenzii să se termine

- SfDacă

SfCâtTimp

Gramatica limbajului sh

comanda:

comandaElementara

(listaCom)

{ listaCom }

if listaCom **then** listaCom [**elif** listaCom **then**
listaCom]* [**else** listaCom]? **fi**

case cuvnt **in** [cuvnt [| cuvnt]*) listaCom ;;]+
esac

for nume **do** listaCom **done**

for nume **in** [cuvnt]+ **do** listaCom **done**

while listaCom **do** listaCom **done**

until listaCom **do** listaCom **done**

comandaElementara:

[element]+

listaCom:

legarePipe [separator legarePipe]*
[terminator]?

legarePipe:

comanda [| comanda]*

Gramatica limbajului sh (cont).

element:

cuvant

nume=cuvant

>cuvant

<cuvant

>>cuvant

<<cuvant

>&cifra

<&cifra

>&-

<&-

separator:

&&

||

terminator

terminator:

;

&

LEGENDA

[]? - cel mult odată

[]+ - cel puțin odată

[]* - de 0 sau mai multe ori

cuvant - secvență de caractere diferite de caracterele albe (spațiu, tab)

nume - secvență ce începe cu literă și continuă cu litere, cifre sau _

cifra - una din cele 10 cifre zecimale

Sucesiuni de comenzi

Op	Ex	Explicație
&&	c1 && c2	c2 se execută doar în cazul în care c1 s-a terminat cu succes
	c1 c2	c2 se execută doar în cazul în care c1 s-a terminat cu insucces
;	c1; c2; c3	c1, c2 și c3 sunt executate secvențial
&	c1 & c2	c1 se va executa în background iar c2 în foreground

Exemplu fișier de comenzi

compilari


```
#!/bin/sh
for fis in *.c
do
    vi $fis
    gcc $fis
done
```

```
$ chmod 755 compilari
```

```
$ ./compilari
```

```
$ sh compilari
```


Mecanisme de substituție

- ``comanda`` - ieșirea comenzii
 - `nume=cuvânt` - definirea unei variabile
 - `$nume`
 - `${nume}`
 - `${nume-cuvânt}`
 - `${nume=cuvânt}`
 - `${nume+cuvânt}`
 - `${nume?cuvânt}`
- 
- substituția valorii variabilei

Variabile SHELL predefinite

- **HOSTNAME** - numele mașinii (*linux.scs.ubbcluj.ro*)
- **HOME** - directorul gazdă al utilizatorului (*/home/scs/an2/gr321/snmr0123*)
- **PATH** - căile de căutare ale fișierelor executabile
- **LOGNAME** - numele sub care utilizatoru și-a deschis sesiunea de lucru
- **SHELL** - tipul interpretorului de comenzi folosit (*/bin/bash*)
- **TERM** - tipul terminalului folosit (*xterm*)
- **MAIL** - fișierul care conține poșta electronică a utilizatorului (*/var/spool/mail/snmr0123*)
- **IFS** - separatorii shell pentru cuvinte
- **PS1** - prompterul principal Unix (*[\u@ \h \W] \ \$*)
- **PS2** - prompterul secundar Unix (*>*)

Linia de comandă

\$	comanda	arg1	arg2	...	arg9	arg10	...argn
	^		^		^		^
	\$0		\$1		\$2		\$9

- \$# - întoarce numărul argumentelor comenzii
- \$* - indica toate argumentele, privite ca un singur șir
- \$@ - indica toate argumentele, ca succesiune de șiruri
- \$- - indica argumentul ce conține opțiunile
- \$? - întoarce codul de terminare al comenzii precedente
- \$! - indica PID-ul ultimului proces lansat în background
- \$\$ - indica PID-ul procesului părinte

COMENZI utile în context shell

- **shift** [n]
- **read** lista_de_nume
- **sleep** n
- **exit** n
- **cut** [-b lista | -c lista | -f lista [-d delim]?] [fis]
- **echo** text
- **test**

Comanda *test*

test *expr* sau [*expr*]

- Evaluează valoarea expresiei și returnează exit status 0 sau 1.

- **Comparări de întregi**

[*int1* -eq *int2*] # *int1* = *int2*

[*int1* -ge *int2*] # *int1* >= *int2*

[*int1* -gt *int2*] # *int1* > *int2*

[*int1* -le *int2*] # *int1* <= *int2*

[*int1* -lt *int2*] # *int1* < *int2*

[*int1* -ne *int2*] # *int1* != *int2*

Comanda *test (cont.)*

- **Comparări/testări de fișiere**

[*file1* -ot *file2*] # Adevărat dacă *file1* este mai vechi decât *file2*

[*file1* -nt *file2*] # Adevărat dacă *file1* este mai nou decât *file2*,
luând în considerare timpul de modificare

[-d *file*] # Adevărat dacă *file* este un director.

[-e *file*] # Adevărat dacă *file* există.

[-f *file*] # Adevărat dacă *file* există și este un fișier regulat.

[-L *file*] # Adevărat dacă *file* este un link simbolic.

[-r *file*] # Adevărat dacă *file* este un fișier din care poți citi.

[-w *file*] # Adevărat dacă *file* este un fișier în care poți scrie.

[-x *file*] # Adevărat dacă *file* este un fișier pe care îl poți executa.

Comanda *test* (*cont.*)

- **Comparări/testări de șiruri**

[-z *string*] # Adevărat dacă *string* este șir vid.

[-n *string*] # Adevărat dacă *string* este șir nevid.

[*string1* = *string2*] # Adevărat dacă *string1* este la fel cu *string2*.

[*string1* != *string2*] # Adevărat dacă *string1* nu este la fel cu *string2*.

- **Compunerea expresiilor *test***

[!*E*] # negarea expresiei *E*

[*E1* -a *E2*] # compunerea *AND* a expresiilor *E1* și *E2*

[*E1* -o *E2*] # compunerea *OR* a expresiilor *E1* și *E2*