

# PROCESUL de LOGIN PENTRU UTILIZATOR

BIOS/UEFI

init( PID=1)

seteaza modul de operare  
si porneste serviciile de baza

porneste getty

login

verifica parola in /etc/passwd,  
/etc/shadow

daca e OK atunci seteaza mediul  
utilizatorului

utilizatorul primeste un UID,  
GID, si ii sunt asociate variabile  
de mediu (HOME, SHELL, USER)

porneste shell-ul(bash, c shell etc.)

# CLI ( COMMAND LINE INTERPRETER)

un program care  
interpreteaza  
comenzi si le  
"executa"

USER  
(tasteaza o comanda  
in terminal)

TERMINAL  
primeste inputul de la tastatura  
afiseaza output text pe ecran

SHELL  
citeste comanda introdusa  
o interpreteaza  
cauta programul in \$PATH  
creeaza un proces nou(fork)  
apeleaza exec() pentru a-l rula

PROGRAM  
ruleaza ca proces separat  
trimite rezultatele inapoi catre shell

KERNEL  
ofera apeluri de sistem(read, write, fork etc)  
gestioneaza procesele si resursele  
transmite output ul catre shell

OUTPUT PE ECRAN  
rezultatul comenzii este afisat  
shell-ul asteapta alte comenzi

PROCES (comanda, fisier etc.)		
fd	descriere	legat de
0	stdin (input)	tastatura
1	stdout (output)	ecran (terminal)
2	stderr (erori)	ecran (terminal)

Un file descriptor este un numar intreg folosit de kernel pentru a identifica un canal de I/O deschis

## REDIRECTAREA OPERATIILOR DE I/O

### output

"> filename" suprascrie filename

">> filename" adauga la finalul fisierului

"2> filename" trimite erorile stderr in fisier

"2>&1" redirect stderr catre stdout

### input

"sort < filename" citeste din fisier

"sort < filename > filename2" combinare i/o

Pipe-uri= este un canal UNIDIRECTIONAL de comunicare intre doua procese, fara a necesita fisiere temporare pentru a lega doua programe

### sintaxa

comanda1 | comanda 2

asta inseamna ca comanda1 scrie in pipe iar comanda2 citeste din pipe si scrie rezultatul in terminal stdout sau pipe

Aceasta comunicare este gestionata de Kernel printr-un buffer intern

FLAG= optiune care modifica comportamentul unei comenzi