

# Instrumente si Tehnici de Baza in Informatica

Semestrul I 2025-2026

Vlad Olaru

# Curs 7 - outline

- perspectiva utilizator (recapitulare)
- perspectiva de sistem
- servicii

# Perspectiva utilizator

- accesul utilizatorului in sistem
  - sub controlul *init*
  - *getty* aloca un terminal unui user
  - *login* autentifica utilizatorul si lanseaza un interpretor de comenzi conectat la terminalul alocat anterior
- interpretorul de comenzi
  - executa comenzi sau scripturi
  - comanda: program executabil, in executie este un proces copil ale interpretorului
  - procesele sunt imaginea din memorie a programelor de pe disc (fisiere executabile)
  - comenzile apeleaza serviciile sistemului de operare pentru:
    - gestiunea fisierelor si directoarelor
    - controlul proceselor
    - administrarea sistemului
    - *samd*

# Perspectiva de sistem

- serviciile sistemului de operare accesibile in doua moduri
  - direct, prin intermediul kernelului (interfata de apeluri sistem)
  - indirect, prin intermediul unor programe specializate implementate in userspace
    - in cele din urma apeleaza tot la kernel
- comenzile shell
  - apeleaza direct serviciile kernelului, *sau*
  - contacteaza *servere* din userspace pt efectuarea serviciului
- servere
  - programe specializate care furnizeaza servicii ale sistemului de operare in spatiul utilizator
  - s.n. si *demoni* (terminologie Unix)
  - accesibile in diferite moduri printr-o forma sau alta de IPC (Inter-Process Communication)
    - memorie partajata (shared memory)
    - schimb de mesaje (message passing)
  - ex: servicii de retea, firewall, imprimare, gestiune a timpului, securitate, etc

# Servicii

- programe de sistem pornite la bootarea sistemului de operare
  - fie servere/demoni, fie programe care contribuie la buna functionare a sistemului si la asigurarea mediului de executie pentru programele utilizator
  - tehnic, procese pornite de *init*
  - furnizeaza diverse servicii de sistem utilizatorului
    - direct, prin servere/demoni
    - indirect, prin asigurarea unui mediu de executie corespunzator programelor utilizator
  - in general, ruleaza atata vreme cat sistemul e in functiune (eventual restartate automat, daca apar erori)
  - configurate pt fiecare runlevel in parte
  - operatii standard: pornire, oprire, reincacarea configuratiilor, afisarea starii
- multiple interfete de acces si gestiune
  - *System V* (Unix)
  - *Upstart* (implementare Linux pt *init*)
  - *Systemd* (varianta recenta Linux)

# Interfata System V

- configurata in fisierul */etc/inittab* in sistemele Unix (si versiuni mai vechi de Linux)
  - *init* citește aici runlevel-ul default (*initdefault*)
- servicii grupate in directoare de tip *rc* (*run commands*), cate unul pentru fiecare runlevel + unul pt etapa initiala, imediat dupa boot

*/etc/rc0.d, /etc/rc1.d, /etc/rc2.d, ..., /etc/rc6.d, /etc/rcS.d*

(in Unix grupate in */etc/rc.d/*)

- fiecare director *rc* contine link-uri simbolice catre scripturi din */etc/init.d* care contin serviciile executate la intrarea sistemului in runlevel-ul respectiv
  - ordinea executiei data de ordinea lexicografica a numelor de servicii
- dezactivare serviciu intr-un anumit runlevel
  - redenumirea link-ului catre script a.i. noul nume incepe cu K+2 cifre  
`$ ln -s ../init.d/ssh K01ssh`
- activare serviciu intr-un anumit runlevel
  - redenumirea link-ului catre script a.i. noul nume incepe cu S+2 cifre  
`$ ln -s ../init.d/rc.local S05rc.local`

# Exemplu rc3.d

```
$ runlevel
N 3
$ pwd
/etc/rc3.d
$ ls -l
total 4
lrwxrwxrwx 1 root root 13 nov 16 2024 K01ssh -> ../init.d/ssh
lrwxrwxrwx 1 root root 16 mar 1 2023 K01xinetd -> ../init.d/xinetd
-rw-r--r-- 1 root root 677 feb 5 2016 README
lrwxrwxrwx 1 root root 16 oct 18 2018 S01apport -> ../init.d/apport
lrwxrwxrwx 1 root root 17 oct 18 2018 S01rsyslog -> ../init.d/rsyslog
lrwxrwxrwx 1 root root 29 oct 18 2018 S01unattended-upgrades -> ../init.d/unattended-upgrades
lrwxrwxrwx 1 root root 15 oct 18 2018 S01uuidd -> ../init.d/uuidd
lrwxrwxrwx 1 root root 15 oct 18 2018 S02acpid -> ../init.d/acpid
lrwxrwxrwx 1 root root 17 oct 18 2018 S02anacron -> ../init.d/anacron
lrwxrwxrwx 1 root root 13 nov 11 2024 S02atd -> ../init.d/atd
lrwxrwxrwx 1 root root 14 oct 18 2018 S02cron -> ../init.d/cron
lrwxrwxrwx 1 root root 14 oct 18 2018 S02dbus -> ../init.d/dbus
lrwxrwxrwx 1 root root 20 oct 18 2018 S02irqbalance -> ../init.d/irqbalance
lrwxrwxrwx 1 root root 20 oct 18 2018 S02kerneloops -> ../init.d/kerneloops
lrwxrwxrwx 1 root root 15 oct 18 2018 S02rsync -> ../init.d/rsync
lrwxrwxrwx 1 root root 27 oct 18 2018 S02speech-dispatcher -> ../init.d/speech-dispatcher
lrwxrwxrwx 1 root root 18 oct 18 2018 S02thermald -> ../init.d/thermald
lrwxrwxrwx 1 root root 18 oct 18 2018 S02whoopsie -> ../init.d/whoopsie
lrwxrwxrwx 1 root root 22 oct 18 2018 S03avahi-daemon -> ../init.d/avahi-daemon
lrwxrwxrwx 1 root root 19 oct 18 2018 S03bluetooth -> ../init.d/bluetooth
lrwxrwxrwx 1 root root 17 oct 18 2018 S03lightdm -> ../init.d/lightdm
lrwxrwxrwx 1 root root 14 oct 18 2018 S04cups -> ../init.d/cups
lrwxrwxrwx 1 root root 22 oct 18 2018 S04cups-browsed -> ../init.d/cups-browsed
lrwxrwxrwx 1 root root 15 oct 18 2018 S04saned -> ../init.d/saned
lrwxrwxrwx 1 root root 21 oct 18 2018 S05grub-common -> ../init.d/grub-common
lrwxrwxrwx 1 root root 18 oct 18 2018 S05ondemand -> ../init.d/ondemand
lrwxrwxrwx 1 root root 18 oct 18 2018 S05plymouth -> ../init.d/plymouth
lrwxrwxrwx 1 root root 18 oct 18 2018 S05rc.local -> ../init.d/rc.local
$
```

# Scripturi init.d

- includ la inceput, dupa specificarea interpretorului, un header cu informatii de configurare (in Linux, definite cf. LSB, Linux Standard Base)

- tehnic sunt comentarii, dar interpretate la init
- cuprinse intre liniile urmatoare

*### BEGIN INIT INFO*

*### END INIT INFO*

- fiecare linie de tipul

*# {keyword}: arg1 [arg2 ...]*

- keyword-uri de pornire/oprire serviciu

- definesc runlevel-urile in care serviciile trebuie pornite/oprite implicit

*# Default-Start      2 3 4 5*

*# Default-Stop      0 1 6*



# Comenzi manipulare interfata SysV

- automatizarea lucrului cu directoarele *rc*
  - diverse comenzi in functie de SO: *chkconfig*, *update-rc.d/bum*, etc

Ex: Ubuntu, dryrun

```
$ update-rc.d -n ssh enable 2 3 4 5
```

- executia operatiilor serviciilor din */etc/init.d*

```
$ service --status-all
```

```
$ service ssh start
```

```
# echivalent /etc/init.d/ssh start
```

```
$ service ssh stop
```

```
# echivalent /etc/init.d/ssh stop
```

```
$ service ssh reload
```

```
# echivalent /etc/init.d/ssh reload
```

```
$ service ssh restart
```

```
# echivalent /etc/init.d/ssh restart
```

# Upstart

- implementare *init* in Linux
  - event-driven
  - evenimentele sunt mesaje trimise catre servicii
- procesele de serviciu s.n. *job*-uri si sunt configurate in */etc/init*
  - *job*-urile sunt actiuni executate ca urmare primirii unor mesaje (evenimente)
  - serviciile configurate au precedenta fata de serviciile din */etc/init.d*
  - modificarea fisierelor de configurare monitorizata cu *inotify*
  - fisierele de configurare includ informatii despre pornirea/oprirea joburilor

Ex:                    *start on runlevel [2345]*  
                      *stop on runlevel [!2345]*

- alte informatii de configurare: *respawn* (job repornit automat in caz de terminare anormala), *exec*

Ex, *tty1.conf*:

```
...  
respawn  
exec /sbin/getty -8 38400 tty1
```

# Upstart (cont.)

- evenimente standard:
    - *startup* (compatibilitate System V, porneste joburi cu evenimente tip *runlevel*) emis la pornirea *init*
    - la pornirea/oprirea job-urilor, *init* emite *starting*, *started*, *stopping*, *stopped*
  - functionare
    - (1) incarca din */etc/init* configuratia job-urilor
    - (2) pt eveniment de start, ruleaza jobul corespunzator
    - (3) joburile create genereaza noi evenimente care la randul lor pornesc alte joburi
    - (4) continua acest ciclu pana cand se termina toate joburile necesare
  - spre deosebire de System V, permite si gestiunea unor sesiuni utilizator
    - *user session mode*
    - $PID > 1$
    - fisierele de configurare servicii:  $\$XDG\_CONFIG\_HOME/upstart$ ,  $\$HOME/.init/$   
 $\$XDG\_CONFIG\_DIRS/upstart$ ,  $/usr/share/upstart/sessions$
- Ex:   \$ upstart -user*

# Initctl

- comanda de lucru cu *upstart*
- vizualizare joburi

*\$ initctl list*

# lista de joburi si starea asociata

*\$ initctl status networking*

# vizualizare job specific

- stare job: pereche *scop/stare curenta*
  - scopul defineste operatia dorita (start, stop, etc)
  - perechea se actualizeaza pe masura ce joburile evolueaza
- pornire/oprire joburi

*\$ initctl start networking*

*\$ initctl stop networking*

- emitere manuala eveniment (mesaj)

*\$ initctl emit some\_event*

# Systemd

- versiunea recenta de *init* pt Linux
- foloseste tinte (*targets*) in loc de runlevels pentru a porni serviciile sistem
  - target-urile au dependente care trebuie indeplinite
  - sistemul de dependente gestioneaza 12 tipuri de unitati (*units*)
- **unit**
  - incapsuleaza diferite obiecte relevante pentru bootare si mentenanta sistemului
  - descrise in fisiere de configurare
  - pot fi *active/inactive*, similar cu procesele (au si stari intermediare)
- **ex tipuri de unitati**
  - *service units* – pornesc si controleaza demonii
  - *target units* – grupeaza unit-uri sau furnizeaza puncte de sincronizare la bootare
  - *device units* – expun echipamentele in *systemd* si activarea bazata pe echipamente
  - *mount units* – controleaza mountpoint-urile din sistemul de fisiere
  - *timer units* – utile pt a activa alte unit-uri pe baza timerelor
  - *sample*.

# Systemd (cont.)

- serviciile nu sunt pornite intr-o ordine anume ca in System V, ci pe baza dependentelor unitatilor
- la fel ca si *upstart*, poate fi pornit in mod utilizator (PID > 1)

Ex:      *\$ systemd -user*

- compatibil cu System V in mare masura
  - scripturile *init.d* sunt suportate si folosite ca alternativa de configurare
  - furnizeaza interfata */dev/initctl* sau */run/initctl* (fisier FIFO pt comenzi *upstart*)
  - suport pentru tool-urile System V
- functionare *systemd*
  - incarca fisierele de configurare a serviciilor din */etc/system/system* sau */lib/systemd/system* (si/sau */usr/lib/systemd/system*)
  - determina tinta de boot (*boot target*), uzual *default.target*
  - determina dependentele boot target si le activeaza

# Systemd targets

- echivalenta runlevels
  - *poweroff.target* – shutdown
  - *rescue.target* – single user mode
  - *multi-user target* – multi-user + networking
  - *graphical.target* – multi-user + networking + GUI
  - *reboot.target* – reboot
  - *default.target* – tinta implicita de boot, link simbolic cate una din tintele de mai sus
- ex: bootare in default-target
  - activeaza unit-urile dependente: *networking.service*, *crond.service*, etc
- Obs: tintele de bootare nu sunt singurele controlate de *systemd*

# Systemd units

- fișiere de configurare unități
  - codifică informații despre servicii, echipamente, mountpoint-uri, etc
  - inspirate de XDG Desktop Entry Specification (gen fișiere *.desktop*)
  - conțin mai multe secțiuni: [Unit], [Install] + secțiuni specifice (e.g. [Service] pt service units)
- [Unit]
  - descriere unit
  - controlează ordinea și/sau timpul activării
- [Service]
  - aici se pornesc, opresc, reincarcă serviciile
- [Install]
  - folosită pentru exprimarea dependentelor



# Exemplu fisier configurare unit ssh.service

## [Unit]

Description=OpenBSD Secure Shell server

After=network.target auditd.service

ConditionPathExists=!/etc/ssh/sshd\_not\_to\_be\_run

## [Service]

EnvironmentFile=-/etc/default/ssh

ExecStartPre=/usr/sbin/sshd -t

ExecStart=/usr/sbin/sshd -D \$SSHD\_OPTS

ExecReload=/usr/sbin/sshd -t

ExecReload=/bin/kill -HUP \$MAINPID

KillMode=process

Restart=on-failure

RestartPreventExitStatus=255

Type=notify

## [Install]

WantedBy=multi-user.target

Alias=sshd.service

# Comenzi systemd

- implementate de *systemctl*

- listare unitati

*\$ systemctl list-units*

- pornire/oprire/repornire unitati

*\$ systemctl start ssh.service*

*\$ systemctl stop ssh.service*

*\$ systemctl restart ssh.service*

- activare/dezactivare unitati

*\$ systemctl enable ssh.service*

*\$ systemctl disable ssh.service*