Procese si demoni

Introducere

- Linux este un sistem de tip multitasking, lucru care inseamna (evident) ca poate indeplini mai multe task-uri simultan.
- Task-urile sunt denumite **procese**
- Folositi comanda **top** pentru a vedea in timp real procesele care ruleaza.
- Ele sunt sortate in functie de timpul de procesare necesar si din aceasta cauza vor fi in continua miscare.

Informatii oferite de comanda top

- In partea superioara a display-ului, ne sunt oferite informatii privind starea sistemului: numarul de utilizatori log-ati, numarul total al proceselor si starea lor, cata memorie avem si cata este utilizata, etc.
- In coloana din stanga veti observa PID-ul procesului (Process ID, numar unic de identificare pentru fiecare proces), urmat de numele posesorului procesului.
- In coloana din dreapta apare comanda care a lansat acel proces. Veti observa ca si procesul **top** apare pe lista

Demoni

- parte din numele comenzilor care au lansat procese (cele din coloana situata in partea dreapta) au ultima litera **d**. Acest lucru semnifica ca acel proces este de fapt un **demon**.
- Demonii sunt procese care nu interactioneaza cu utilizatorul, adica ei sunt porniti de catre sistem.
- Demonii ne ofera sevicii cum ar fi email, conectare la internet, printare, etc.
- Folositi tasta **u** (in interiorul comenzii **top**), apoi introduceti numele utilizatorului pentru care doriti sa afisati procesele (numai procesele pentru care acel utilizator e owner).
- Folositi tasta **?** pentru a vedea comenzile posibile in **top**

Controlul proceselor

- Am mentionat mai sus ca fiecare proces primeste un PID in momentul in care este lansat, acesta fiind unic. PID-ul este setat pentru identificarea proceselor de catre bash. Procesele au o structura ierarhica, relatiile dintre ele putand fi de tipul tata-fiu. Procesul care are PID 1 este **init** , el fiind primul initializat. Un proces poate porni alte procese, acestea din urma numinduse fii pentru procesul care i-a pornit.

- un proces poate lucra in foreground (output-ul va fi display-ul, iar input-ul va fi tastatura), background (el nu mai interactioneaza cu utilizatorul, facand posibila lansarea altor procese).
- Pentru a lansa un proces in background, se adauga simbolul **&** (ampersand). Procesul va primi un PID si va rula in background.
- Pentru a trimite un proces in background, dupa ce el a fost lansat in foreground, se foloseste combinatia de taste CTRL+Z. Aceasta va trimite procesul in background, dar in va lasa intr-un stadiu inactiv (**Stopped**). Pentru a reactiva procesul in background, se foloseste comanda **bg** insotita de caracterul **%** si numarul **job-ului** corespunzator. Acest numar il aflam folosind comanda **jobs** (Ex.: **$ bg %5** daca numarul jobului este 5).
- Pentru a aduce un proces din background in foreground, vom folosi comanda **fg** cu aceeiasi sintaxa ca si **bg** (Ex.: **$ fg %6** daca numarul procesului este 6).
- Un proces poate fi terminat folosind comanda **kill** (Ex.: **$ kill 3434** , daca dorim sa terminam procesul cu PID=3434). Aceasta comanda poate fi utilizata cu mai multi parametrii, fiecare reprezentand un tip de semnal trimis catre sistem. Cand il rulam fara parametrii, un semnal de tip SIGTERM, care va permite salvarea buffere-lor si iesirea corecta din program. Totusi, unele procese pot fi setate sa ignore un astfel de semnal.
- Pentru a termina un proces fortat, se foloseste **kill -9 <PID>** . Acesta va trimite un semnal de tip SIGKILL, care va forta incheierea procesului.

Init

Introducere

- Init are rolul de a pune totul in functiune corespunzator.
- Ce inseamna asta?
    - Verifica integritatea sistemelor de fisiere si le monteaza, porneste daemonii pentru a inregistra mesajele de sistem, pentru a configura reteaua, comunicarea cu perifericele (mouse, tastatura), etc.
    - Totodata porneste procese de tip Getty, care afiseaza promptul de login la terminale, preia informatiile, adica utilizatorul si parola si le preda catre **login** care verifica fisierul /etc/shadow pentru a vedea daca sunt valide.

Despre Init

- Init preia datele din /etc/inittab, fisier care ii “spune” ce trebuie sa faca.
- In mod normal va rula in primul rand un scrip de initializare, acesta fiind interpretat de catre Bash, aceeasi interfata care asigura interpretarea comenzilor introduse in mod consola.
- Scriptul de initializare pentru Debian este /etc/init.d/rcS.
- Urmatorul script care va fi rulat de catre Init va fi specific runlevel-ului.
- Runlevel-ul default este setat tot in /etc/inittab.

- Ultimele lucruri importante care vor fi incarcate de catre Init sunt Getty-uri. Acestea vor fi repornite de catre Init ori de cate ori ele se opresc dintr-un motiv sau altul.
- kill -HUP 1

Fisierul /etc/fstab

- Acest fisier determina ce sisteme de fisiere trebuiesc montate.

Exemplu de linie din /etc/fstab: /dev/hda1 / ext3 defaults,errors=remount-ro 0 1

- Comanda mount -a este rulata in unul din scripturile lansate de catre Init. Aceasta comanda monteaza toate sistemele de fisiere conformt /etc/fstab. Pentru mai multe detalii asupra acestui fisier vezi man fstab, man inittab.

Runlevel

Introducere

- Intr-o maniera cat mai simplista, un runlevel determina care programe vor fi executate la pornirea sistemului.
- Exista 6 runlevel-uri (in mod uzual), dar putem avea mai multe!
  - 0 - system Halt
  - 1 - single user
  - 2 - full multi-user mode (default)
  - 3-5 - alte moduri multi-user similare cu runlevel 2
  - 6 - system reboot

Runlevel-uri speciale

- Runlevel 0 este runlevel-ul corespunzator inchiderii calculatorului.
- Runlevel 1 “single user” (cunoscut si sub numele runlevel S). Un mod de adresare mai exacta a acestui runlevel ar fi modul rescue sau troubleshooting. In acest runlevel nu mai este posibila logarea simultana a mai multor useri, fapt care impiedica si logarea din exterior a unui posibil user malitios.
- Runlevel 6 “system reboot” este similar cu runlevel 0, cu diferenta ca acesta va executa un reboot.

Unde gasim runlevel-urile?

- Dupa cum v-ati obisnuit deja, totul este configurat cu ajutorul fisierelor.
- Toate runlevel-urile se afla in director /etc/ in felul urmator:
  - /etc/rc0.d runlevel 0
  - /etc/rc1.d runlevel 1

- /etc/rc2.d runlevel 2
- /etc/rc3.d runlevel 3
- /etc/rc4.d runlevel 4
- /etc/rc5.d runlevel 5
- /etc/rc6.d runlevel 6

Idee generala

rcX.d este directorul corespunzator runlevel-ului X, unde X=0:6

Ce contin directoarele rcX.d?

#ls /etc/rc2.d

S10sysklogd S20dirmngr S20pcmcia S20xprint S99rmnologin S11klogd S20exim4 S20samba S89atd S99stop-bootlogd S14ppp S20inetd S20ssh S89cron S99xdm S20cupsys S20makedev S20xfs S99kdm

- Fiecare fisier din acest director este un link simbolic catre un script din /etc/init.d

Cum interpretam fisierele din rcX.d?

- Ca sintaxa ele sunt de forma urmatoare: [K | S] + nn + [String]
- "String" reprezinta numele scriptului care va fi apelat (cel din /etc/init.d) de catre link-ul simbolic.
- "nn" este un numar de 2 cifre (01 - 99) in functie de ordinea in care dorim sa fie pornite scripturile.
- "K" si "S", K=kill, S=start.
    - La comutarea intr-un nou runlevel, fisierele care incep cu "K" vor fi executate cu parametrul stop, iar cele care incep cu "S" vor fi executate cu parametrul start.
    - In Debian, daemonii care ruleaza nu vor fi reporniti la schimbarea runlevel-ului din motive de optimizare.
- Totul se face in ordine alfabetica, deci mai intai vor fi executate fisierele care incep cu "K" apoi cele care incep cu "S".
- Pentru a vedea corespondentul fiecarui link simbolic rulati comanda ls -l in directorul runlevel-ului dorit.
- Comanda pentru aflarea runlevel-ului curent si anterior este runlevel.

Exemplu:

- student:/etc/rc2.d# runlevel
- N 2

- N = nu a existat nici un runlevel anterior
- 2 = runlevel curent
- Schimbarea runlevel-ului se face cu ajutorul comenzii telinit (tell init to ...).

Exemplu: telinit 3 schimba runlevel-ul curent in runlevel 3.

Parametrii pe care ii poate primi un script din /etc/init.d

- start = pornire serviciul
- stop = oprire serviciul
- restart = oprire apoi repornire serviciu
- reload = configurarile unui serviciu sunt reincarcate fara a fi necesara oprirea acestuia
- force reload = la fel ca reload cu mentiunea ca daca serviciul este oprit el va fi pornit