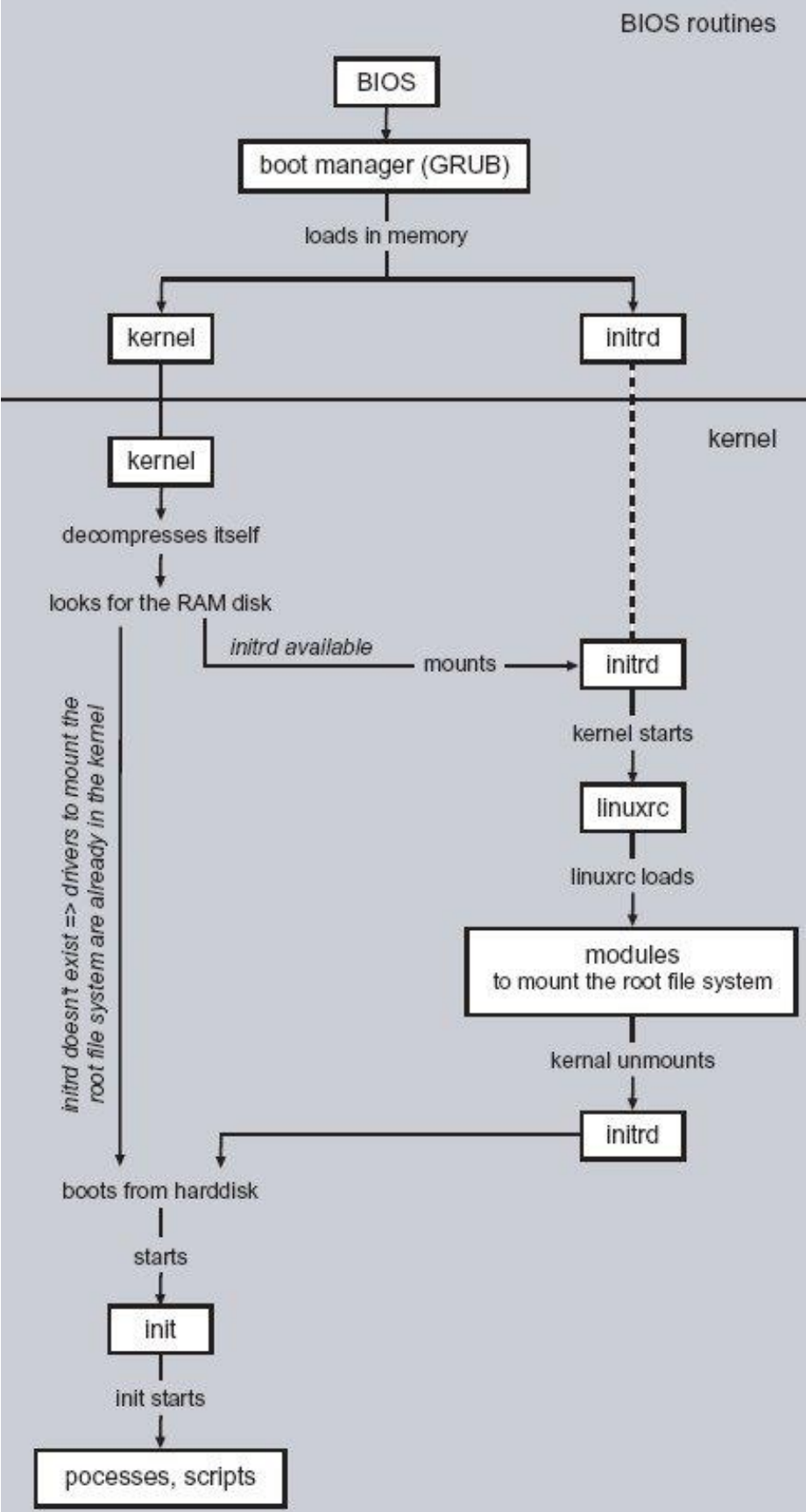


Diverse

[10 Diverse](#) » [10.1 Managementul serviciilor si Procesul de Butare](#)

Managementul serviciilor si Procesul de Butare

Linux Boot Process



O sarcina importanta a oricarui administrator de sistem este de a stabili ce servicii pornesc automat la butare, ce servicii trebuie oprite etc.

Etaple butarii unui calculator care ruleaza Linux sunt urmatoarele:


1) BIOS executa POST (Power on Self Test) si conform cu configuratia sa (de unde este configurat sa buteze: Hard Disk, CD-ROM, DVD-ROM etc) citeste MBR-ul care reprezinta primul sector al hard diskului (primii 512 bytes).

2) In MBR (Master Boot Record - primul sector al Hard diskului) se gaseste un boot loader (GRUB sau LILO) care incarca in memorie de pe disk, kernelul si initrd inainte sa monteze root file system. Functionalitatea interna a lui GRUB permite sa citeasca de pe disk kernelul si initrd si sa le incarce in memorie inainte sa monteze partitia pe care acestea se gasesc. Kernelul se decompima si foloseste modulele din initrd (monteaza initrd) pentru a buta de pe disk. Daca nu exista initrd functionalitatea pt. a buta de pe hard disk trebuie sa fie compilata in kernel.

Nota

initrd reprezinta un RAMDISK


Fizic acesta este un fisier. Exemplu: /boot/initrd.img-2.6.22-14-server

 initrd contine module necesare kernelului in prima faza a procesului de butare inainte sa monteze root file system.

Exemplu: initrd ar putea contine un modul care reprezinta driver pentru SATA si care nu a fost compilat in kernel. Kernelul are nevoie de acest driver pentru a putea monta hard diskul.

3) Primul program pe care kernelul il executa de pe disk este init (parintele tuturor proceselor).

Nota

 Ubuntu nu mai foloseste initializarea clasica a sistemului folosind doar init ci foloseste si Upstart. [Detalii](#).

4) Se executa /etc/rc.d/rcS precum si toate scripturile din directorul /etc/rcS.d/

5) In functie de runlevelul in care intra sistemul (Ubuntu ruleaza in runlevelul 2) se executa /etc/rc*.d/*

* (asterix) reprezinta runlevelul in care intra sistemul. Poate fi 0,1,2,3,4,5 sau 6.

Exemplu:

Daca se intra in runlevel 2 (Ubuntu) se executa scripturile din /etc/rc2.d/

Daca se intra in runlevelul 5 (alte distributii precum Fedora) se executa scripturile din /etc/rc5.d/

Pentru fiecare serviciu exista in /etc/rc[0-6].d/ un fisier de forma SnrNumeServiciu si KnrNumeserviciu.

Exemplu: S10network si K10network (optional).

Fisierele care incep cu litera S se executa la intrarea in runlevelul respectiv iar cele care incep cu litera K se executa la iesirea din runlevelul respectiv.

Ordinea de executie a fiecarui serviciu este importanta si se stabileste in functie de numarul care urmeaza dupa litera S sau K.

Fisierele din /etc/rc[0-6].d/ sunt de fapt symlinkuri catre fisierele din /etc/init.d/ - directorul

global cu startup files. Fiecare fisier din /etc/init.d/ trebuie sa poata primi ca parametru start|stop|reload|restart.
Exemplu: executia fisierului /etc/rc3.d/S10network este echivalenta cu executia scriptului /etc/init.d/network start.

Nota

 rc = Runlevel Script / resource control

Managementul serviciilor

Pentru a stabili ce servicii pornesc la butare avem mai multe optiuni:

1) Varianta 1 (manuala)

- a) se creaza un script in /etc/init.d/ care sa porneasca serviciul atunci cand primeste parametrul start si sa-l opreasca atunci cand primeste parametrul stop.
- b) se creaza un symlin in /etc/rc2.d/ catre /etc/init.d/nume_script. Acesta trebuie sa fie de forma SnrNume. Exemplu: S99firewall


2) Varianta 2

Se foloseste comanda update-rc.d

Exemplu

1. Pornire serviciu ssh in mod default (cuvantul cheie default porneste serviciul in runlevel 2,3,4 si 5 si-l opreste in 0,1 si 6)
update-rc.d ssh defaults

2. Remove the symlink from /etc/rc[0-6]d/. Serviciul nu va mai porni in runlevelul specificat.

 update-rc.d ssh remove

3. Stabilire runlevel explicit in care se porneste/opreste serviciul
update-rc.d ssh start 0123 stop 456

Nota: ssh este numele scriptului din /etc/init.d catre care se vor vrea symlink-urile corespunzatoare.

Oprire/Pornire de servicii:

Pentru a opri/porni un serviciu exista in principiu 2 modalitati.

1. Folosind startup scripts si pasand argumentul start, stop sau restart.

Exemplu:

/etc/init.d/ssh start

/etc/init.d/network restart

2. Folosind comanda service

Exemplu:

service sendmail start

```
service apache2 stop  
service ssh restart
```

Comanda service ruleaza scriptul pe care il primeste ca parametru (Exemplu: sendmail) cu optiunile aferente (start, stop etc). Scriptul trebuie sa se gaseasca in /etc/init.d/

Nota



service se gaseste in directorul /sbin sau /usr/sbin si este un script de shell

Daca nu dorim sa cream cate un script pentru fiecare serviciu putem adauga comenzi in fisierul /etc/rc.local care este executat dupa terminarea tuturor celorlalte scripturi de initializare.

Trebuie sa existe un symlink catre el in /etc/rc[0-6].d/

Administrarea pachetelor software

[10.2.1 deb & apt-get](#)

[10.2.2 rpm](#)


Exista 2 posibilitati principale pentru managementul noilor programe (instalare/dezinstalare/update etc).

1. Instalarea de programe folosind fisiere binare, care au fost deja compilate pentru o anumita distributie si platforma hardware
2. Compilarea surselor respectivelor programe. Marea majoritate a programelor din Linux sunt programate in C (Exemplu: Kernelul, aproape toate serverele etc) si se compileaza folosind gcc (GNU C Compiler)

Instalarea/dezinstalarea/update-ul de programe direct prin compilare este in general greoi fiindca nu exista un management centralizat care sa tina evidenta fiecarui fisier cu care a venit programul, locatia lui pe hard disk etc. Avantajele compilarii sunt in principal cresterea flexibilitatii (optiunile de instalare) si performantei. Administratorul poate compila un server doar cu optiunile necesare task-ului sau si-l poate optimiza pentru o anumita platforma.

Exista distributii de linux bazate pe compilare (Exemplu: Gentoo, Slackware, LFS) precum si distributii bazate integral pe fisiere binare deja compilate (Exemplu: SuSE, RedHat, Fedora, CentOS, Debian, Ubuntu/Kubuntu, Knoppix, Mandriva).

Nota

 Compilarea de programe se poate realiza inclusiv pentru distributiile bazate pe fisiere binare, iar cunoasterea modului de compilare a programelor pe Linux este **OBLIGATORIE** pentru orice Admin chiar si daca se foloseste o distributie precum SuSe, RedHat sau Ubuntu.

Important

In Linux notiunea de instalare a unui program este echivalenta cu copierea fisierelor componente in anumite locatii. Dezinstalarea unui program este echivalenta cu stergerea fisierelor care fac obiectul programului respectiv.

deb & apt-get

Un format binar foarte des intalnit si folosit este deb. Acesta isi are originea in Debian si se foloseste acum pe orice distributie Debian Based (Debian, Ubuntu, Kubuntu, Knoppix, Damm Small Linux, Mepis, Linex, Xandros etc).

Nota



Numele Debian a fost dat de creatorul acestei distributii Ian Murdock si reprezinta o combinatie intre numele sotiei sale Debra si numele sau Ian.

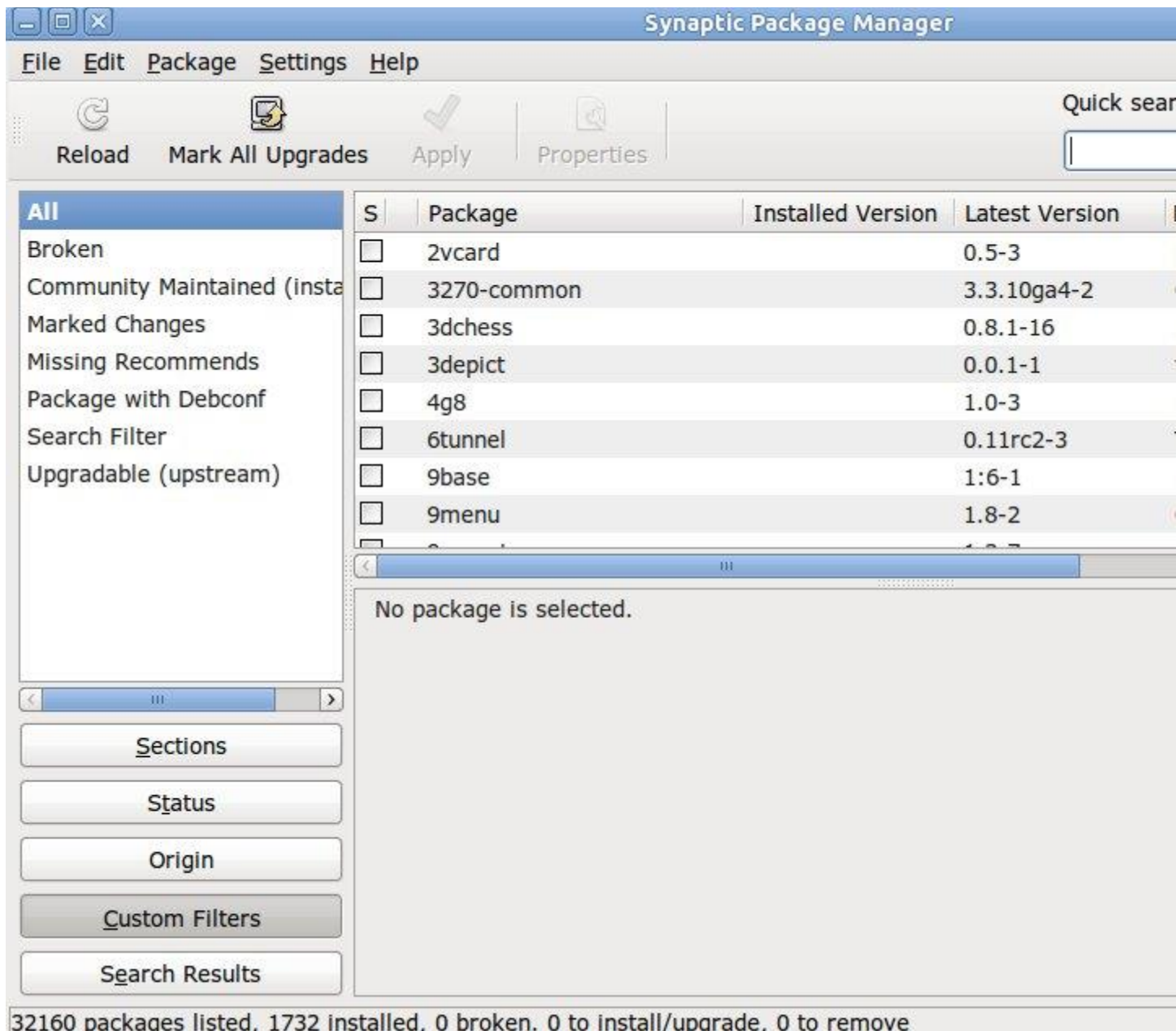
Un fisier deb este de fapt o arhiva Unix de tip [ar](#) care contine 2 alte arhive gzip sau bzip. Intr-una se gasesc informatii metadata (versiune, release, dependinte etc) iar in cealalta binarele efective ale programului care se gasesc in deb.

Nota



root este singurul user care poate instala/dezinstala/actualiza aplicatii. Comenzile specifice se ruleaza de catre root (anterior s-a executat su) sau se ruleaza de catre un user neprivilegiat si sunt prefixate cu sudo.

Programul folosit pentru instalarea, stergerea sau modificarea programelor instalate din deb-uri se numeste dpkg. De multe ori acesta se foloseste cu ajutorul unui alt program numit apt. Synaptic este o interfata grafica pentru apt. Pentru a instala synaptic (in cazul in care nu exista) se ruleaza: `sudo apt-get install synaptic`



apt-get

apt-get este o aplicatie in linie de comanda folosita pentru managementul pachetelor deb si este considerat user back-end tool pentru apt

Mod folosire:

apt-get [options] [command] [package ...]

Comenzi:

update = se foloseste pentru sincronizarea caracteristicilor pachetelor software instalate in sistem cum ar fi versiunea acestora cu sursele acestora din repositoarele specificate in /etc/apt/sources.list . Se recomanda rulara comenzii apt-get update inainte de rulara lui apt-get folosind optiunea upgrade. Cu ajutorul acestei comenzi apt-get ia la cunostinta despre ultimele versiuni existente in repozitoare.

upgrade = se foloseste pentru instalarea automata a ultimei versiuni a tuturor pachetelor instalate in sistem din repositoarele specificate in /etc/apt/sources.list. Pachetele sunt automat downloadate din repozitoare si upgradate la ultima versiune. NU se instaleaza pachete noi si nu se dezinstaleaza pachete deja instalate.

dist-upgrade = pe langa procesul de upgrade, optiunea dist-upgrade are grija in mod inteligent si de dependintele pachetelor care se upgradeaza.

install nume-pachet1 nume-pachet2 etc = se foloseste pentru instalarea de noi programe din repozitoare impreuna cu dependintele necesare.

remove nume-pachet1 nume-pachet2 etc = se foloseste pentru dezinstalarea respectivelor pachete fara a sterge si fisierele de configurare. Daca se doreste stergerea inclusiv a fisierele de configurare se foloseste optiunea purge.

check = reprezinta un tool de diagnostic folosit pentru update-ul cache-ului pachetelor precum si verificarea dependintelor cu probleme

clean = sterge pachetele downloadate din repozitor (cache-ul pachetelor) si care au fost sau nu deja instalate.

Optiuni:

-m = ignora pachetele lipsa

-d = doar downloadeaza pachetele, dar nu le si instaleaza

-f = repara "broken dependencies"

Exemplu: apt-get -f install bind9

-y = se asuma raspunsul yes la orice intrebare. Folositor pentru upgrade automat.

-h = printeaza help-ul comenzii

-V = printeaza versiunile pachetelor care se instaleaza

Exemplu: apt-get -V install bind9

Nota

1. Daca se doreste generarea unei liste cu toate pachetele instalate in sistem se poate rula:
dpkg --get-selections > installed-software



2. Daca se doreste afisarea tuturor fisierele continute intr-un pachet se ruleaza dpkg -L NUME_PACHET.

Exemplu: dpkg -L coreutils

3. Dacă dorim să aflăm din ce pachet face parte un fișier se folosește opțiunea `-S` a lui `dpkg`.
Exemplu: `dpkg -S /bin/ls`

4. Dacă dorim să instalăm un pachet `deb` pe care-l avem local pe disk se rulează `dpkg -i NUME_PACKET`.

rpm

Nota



Toate exemplele si conceptele acestui curs au fost realizate si explicate pentru Ubuntu sau alte distributii derivate din Debian. Informatiile din acest sub-capitol sunt folositoare persoanelor care folosesc distributii bazate pe rpm-uri.

RPM reprezinta un acronim recursiv de la RPM Package Manager. Initial s-a numit Red Hat Package Manager si a fost dezvoltat de firma RedHat.

RPM se refera la 2 aspecte:

1. Un format de fisier binar care reprezinta un program/aplicatie
2. O aplicatie care ruleaza in consola in modul text folosita pentru instalare, dezinstalare, update si interogare a bazei de date cu toate programele RPM instalate.

RPM foloseste o baza de date care se gaseste in /var/lib/rpm. Este formata din fisierul Packages care contine informatii de tip metadata despre rpm-urile instalate precum si alte fisiere folosite cu scop de indexare.

Numele unui packet RPM este de cele mai multe ori conform unui "standard" si contine urmatoarele informatii:

- a) numele programului
- b) versiunea programului
- c) package release: identifica distributia pentru care a fost compilat (fc6, mdv, rhel4, suse100 etc)
- d) arhitectura pentru care a fost compilat (i386, i686, athlon, noarch). noarch se refera la pachete care nu sunt compilate pentru o arhitectura anume (Exemplu: fonturi, sau alte elemente folosite de alte aplicatii).

Exemplu: apache-1.3.33-6mdk.x86_64.rpm
lyx-1.4.3-3.fc4.i386.rpm

Un fisier rpm este format din urmatoarele componente:

- 1) Lead (identifica fisierul ca fiind rpm si contine header absolute)
- 2) Semnatura digitala
- 3) Header (meta information - package name, version, file list, architecture)
- 4) arhiva gzip cu binarele programului

Distributiile de Linux care folosesc RPM sunt urmatoarele (lista incompleta):

- RedHat Enterprise Linux
- Fedora Core
- Mandriva

- CentOS
- OpenSuSE
- Yellow Dog (Fedora pentru PowerPC)
- Turbo Linux

Comanda rpm

rpm OPTIONS package_name

Optiuni:

-i = install

-q =query

-e = erase

-F = freshen (upgrade doar daca o versiune mai veche exista deja)

-U = upgrade (instaleaza pachet daca pachetul nu exista sau upgrade pachet daca exista. Daca exista versiunea mai veche este stearsa)

-v = verbose

-f file_name = query package owning file

-l = list files in package

--force = forteaza instalarea chiar daca nu sunt satisfacute dependintele

Exemplu

1. Afiseaza toate fisierele documentatie ale pachetului samba

rpm -qd samba

2. Afiseaza numele pachetului din care face parte un fisier din sistem

rpm -qf /usr/bin/clamav

3. Afiseaza toate fisierele care fac parte dintr-un pachet

rpm -ql tightvnc



4. Reinitializeaza baza de date daca aceasta se corupe

rpm --rebuilddb

5. Afiseaza toate RPM-urile instalate in sistem

rpm -qa

6. Instalare programe din rpm-uri locale sau direct de pe Internet

rpm -ivh httpd-1.3.3.fc4.rpm

rpm -ivh ftp://rpmfind.net/linux/fedora/core/4/i386/os/Fedora/RPMS/gftp-2.0.18-2.i386.rpm