14.COMPILAREA KERNEL-ULUI LINUX. LUCRUL CU MODULE

14.1. Concepte

14.1.1. Versiuni de kernel

Kernel-ul Linux este responsabil cu managementul resurselor – procesor, RAM, hardware. El este dezvoltat de diversi programatori, sub coordonarea lui Linus Torvalds, creatorul sau. Kernel-ul este in permanenta dezvoltare, fiecare noua varianta de kernel fiind caracterizata printr-o versiune.

Versiunile de kernel au trecut prin diferite forme de-a lungul timpului. Iinainte de versiunea 2.6, ele se prezentau sub forma 2.4.28, unde semnificatia componentelor era:

- 2 – reprezinta versiunea de kernel
- *4* – reprezinta *major revision*
- 28 – reprezinta *minor revision* dar a carei schimbare introduce totusi incompatibilitati majore

Versiunile cu major revision par erau cele stabile, iar cele cu major revision impar erau cele de development, in care se incercau ultimele facilitati/drivere/etc.

Incepand cu 2.6, s-a eliminat conventia par/impar, iar versiunea de kernel este formata din 3 sau 4 secvente, dupa cum urmeaza:

- versiunile stabile au forma 2.6.32 sau 2.6.32.1 (ultimul caz se intalneste atunci cand sunt aplicate remedii urgente pentru bug-urile versiunii curente)
- versiunile instabile sunt asa-numitele “release candidates” si sunt de forma 2.6.32-rc8 (release candidate versiunea 8)

14.1.2. Ratiuni si modalitati de compilare

Recompilarea de kernel este o operatie avansata ce presupune un numar cunostinte de hardware si software. Ea se realizeaza doar pentru kernelurile open-source si poate fi dicatat de necesitati precum:

- includerea de drivere sau facilitati ce nu sunt prezente in kernelul livrat cu distributia dar sunt disponibile in sursele de kernel
  - includerea de drivere 3rd party
  - adaugarea de optiuni/facilitati legate de securitate
  - personalizarea kernelului in functie de necesitatile sistemului (optimizari etc)
- “curatarea” kernelului de facilitatile/driverele nefolosite

Kernelul Linux este gandit modular; atunci cand il compilam, putem decide, pentru majoritatea facilitatilor sale, daca vor fi incluse sau nu in forma finala, compilata, a kernelului. Includerea se poate realiza in doua moduri:

- includerea facilitatii dorite in fisierul kernel. Acesta se afla de obicei /boot si este acel executabil incarcat de catre boot loader in cadrul bootarii sistemului de operare
- includerea facilitatii dorite sub forma de modul. In acest caz va fi creat la compilare cate un fisier separat pentru fiecare dintre componentele compilate astfel. Fisierele modul sunt plasate in /lib/modules si pot fi atasate la kernel in timpul rularii acestuia (spre exemplu, putem incarca un driver pentru un dispozitiv USB numai cat timp dispozitivul este prezent in port)

14.2. Recompilarea kernelului Linux

14.2.1. Obtinerea si dezarhivarea surselor de kernel

Avem de ales intre 3 posibile proveniente pentru sursele de kernel:

- vanilla sources – sunt sursele lansate de catre Linus Torvalds si echipa sa, disponibile pe www.kernel.org sau orice mirror (majoritatea providerilor romani tin pe FTP-ul lor si surse de kernel)
- sursele de kernel cuprinse in distributia Linux folosita. Deseori distributiile aplica propriile patch-uri/optimizari de kernel de care este bine sa se tina cont. Aceste surse sunt in general disponibile de pe CD-urile/site-ul distributiei folosite sau de pe unul dintre mirror-urile sale; in functie de provenienta, sursele pot fi in format rpm (kernel-headers si kernel-sources) sau tar.gz/tar.bz2
- surse alternative – ramuri derivate din kernel-ul original, care au continuat dezvoltarea in alt mod

Dezarhivarea surselor se face de obicei in /usr/src. In exemplul urmator se va presupune folosirea versiunii 2.6.20:

```
cd /usr/src
tar jxf linux-2.6.20.tar.bz2
cd linux-2.6.20
```

In general se creeaza suplimentar un symlink */usr/src/linux* care pointeaza catre directorul surselor de kernel, astfel incat sursele kernelului sa poata fi usor gasite de aplicatii care au nevoie de ele (de exemplu, la compilarea unei aplicatii care are nevoie de headerele de kernel).

14.2.2. Configurarea surselor in vederea compilarii

Efectuata manual si de la zero, este cea mai de durata etapa din procesul de recompilare a kernelului; in acelasi timp cere si cele mai multe competente, deoarece comporta decizii importante in privinta multor parametri vitali pentru functionarea sistemului de operare. (vezi detalii mai jos sugestii si proceduri pentru recompilare)

Odata configurarea efectuata, ea se salveaza intr-un fisier numit *.config*, plasat in directorul surselor kernelului. Acestui fisier i se poate face o copie intr-un alt director, astfel incat, la o alta compilare de kernel, setarile facute sa poata fi importate. Kernel-urile mai noi suporta includerea acestui fisier in kernel-ul insusi si publicarea sa dedesubt-ul lui /proc pe parcursul rularii kernelului.

Configurarea surselor kernelului se poate realiza in mai multe feluri:

1. **generarea unei configuratii default**

```
make defconfig
```

   Comanda genereaza o configuratie default, care insa apoi trebuie customizata pentru sistemul in cauza.

2. **(optional) importarea unei configuratii salvate anterior**, daca dispunem de fisierul salvat. In acest fel, suntem scutiti de a mai trece inca o data prin intregul proces de configurare

```
# presupunem ca fisierul cu configurarea se afla in /boot si se numeste config_old
cp /boot/config_old .config
```

Setarile importate constituie baza pentru crearea unei noi configuratii, de aceea in cele mai multe cazuri aici se continua cu una dintre comenzile make menuconfig sau make xconfig.

3. **generarea manuala a unei configuratii** sau customizarea uneia importate – se poate face in 3 moduri:
   - **make config** – in mod text, varianta cea mai putin prietenoasa
   - **make menuconfig** – meniuri text-based; necesita biblioteca ncurses
   - **make xconfig** – front-end grafic, disponibil numai in prezenta serverului X

Dupa incheierea acestei operatiuni, setarile sunt salvate in fisierul *.config* din directorul surselor kernelului.

14.2.3. Compilarea

Comanda *make* realizeaza compilarea tuturor dependintelor, a kernelului si a modulelor (pentru facilitatile din kernel compilate ca modul). Ei i se poate pasa ca argument numarul de joburi de compilare rulate in paralel, pentru sporirea vitezei:

```
# numarul ce urmeaza dupa -j se recomanda a fi dublul numarului de procesoare pentru multi-core
make -j4
```

Dupa compilare, kernelul poate fi gasit sub forma unui fisier numit *bzImage* intr-unul dintre urmatoarele subdirectoare ale directorului surselor:

- *arch/i386/boot/bzImage* – pentru arhitecturi pe 32 de biti
- *arch/x86_64/boot/bzImage* – pentru arhitecturi pe 64 de biti

14.2.4. Instalarea modulelor

```
make modules_install
```

Modulele sunt instalate in /lib/modules/*nume_kernel*/, unde *nume_kernel* include si localversion-ul stabilit la inceputul procesului de configurare a surselor.

14.2.5. Instalarea noului kernel

Kernelul Linux este plasat de obicei in /boot:

```
cp arch/i386/boot/bzImage /boot/nume_dorit_kernel
```

Kernelul poate avea orice nume de fisier valid, atata timp cat la reconfigurarea boot managerului se specifica acelasi nume.

Alternativ, in unele distributii functioneaza si comanda *make install*, care copiaza kernel-ul la locul sau si actualizeaza configurarea boot manager-ului.

14.2.6. Creare initial ramdisk (pas optional)

Initial ramdisk reprezinta un sistem de fisiere temporar, care permite montarea sistemului de fisiere real. El poate contine executabile, module cu drivere etc. (ex: driverul pentru controllerul SATA la care este conectat hard disk-ul pe care se afla sistemul de fisiere real). *Initial ramdis*k-ul este util atunci cand kernel-ul nu contine driverele necesare pentru a putea boota (spre exemplu, lipseste driverul pentru controllerul de hard-disk sau pentru sistemul de fisiere de pe partitia /). In masura in care kernelul contine toata functionalitatea necesara bootarii si montarii sistemului de fisiere /, *initial ramdisk*-ul poate lipsi.

Comanda folosita pentru crearea *initial ramdisk*-ului este:

```
mkinitrd /boot/numeinitrd.img
```

Comanda va crea imaginea RAMdisk-ului – un fisier in care va copia toate modulele IDE, SCSI si drivere de sisteme de fisiere prezente in */etc/conf.modules*. Acest fisier va fi folosit de catre boot manager in timpul bootarii pentru a crea in memorie RAM disk-ul, ale carui resurse le va folosi kernel-ul.

14.2.7. Reconfigurarea boot managerului in vederea bootarii noului kernel

Atentie! NU stergeti kernelul vechi – nici din /boot, nici din meniul boot manager-ului! Veti dori sa reveniti la el in cazul in care noul kernel nu functioneaza corespunzator sau deloc.

Odata noul kernel instalat in */boot*, avem posibilitatea de a boota doua kerneluri diferite - insa cu acelasi sistem de fisiere. Ce ramane de facut este editarea fisierului de configurare al boot managerului:

- */boot/grub/menu.lst* in cazul GRUB
- */etc/lilo.conf* in cazul LILO

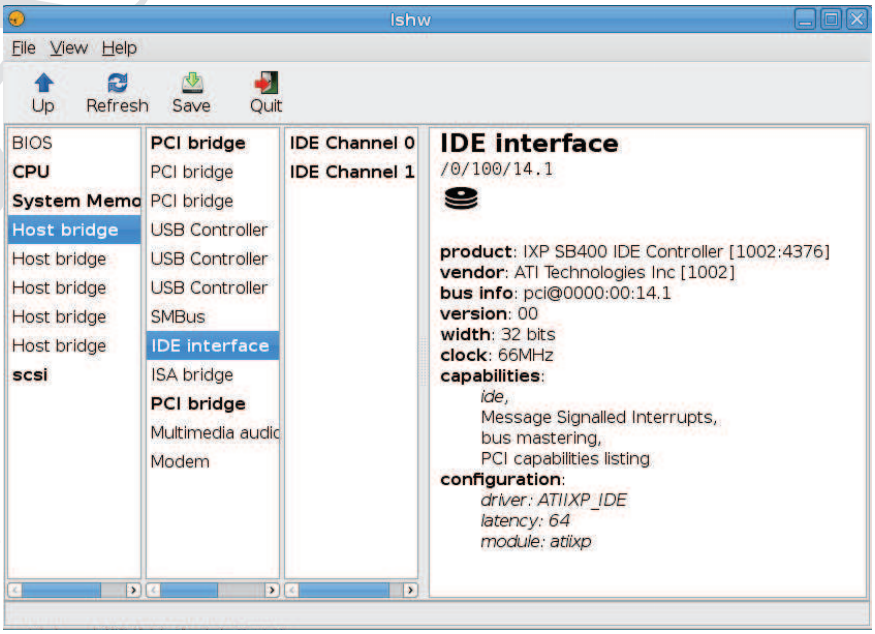
Iata cum ar putea arata sectiunea care trebuie adaugata in menu.lst in cazul lui GRUB:

```
title Linux Recompilat

# a se inlocui X cu numarul corespunzator partitiei Linux pe care se afla /
# ATENTIE! Numerotarea se face de la 0!
root (hd0,X)

# a se inlocui hda cu sda in caz de SATA sau SCSI. ATENTIE! Y=X+1!
# verificati ca numele kernelului sa fie acelasi cu cel folosit la copierea sa in /boot
kernel /boot/nume_dorit_kernel ro root=/dev/hdaY

# optional, daca este nevoie
initrd /boot/numeinitrd.img
```

14.2.8. Sugestii si proceduri pentru configurarea surselor in vederea compilarii

14.2.8.1. Determinarea hardware-ului prezent in sistem

In kernelurile mai noi (dupa 2.6.13), detectia hardware-ului si popularea directorului /dev este facuta automat de catre udev – un serviciu care trateaza evenimentele generate de */sbin/hotplug* la adaugarea sau eliminarea unei componente hardware din sistem. udev face disponibile detaliile tuturor dispozitivelor prin intermediul unui pseudo-sistem de fisiere aflat in /sys, in care dispozitivele sunt structurate ierarhic, pe categorii.



Exemple de subdirectoare din /sys:

- block – contine fisiere dispozitiv pentru block devices (ex: hard-disk-uri)
- bus – informatii despre magistralele de date prezente in sistem (PCI, PCI express, PCMCIA etc)

Tabelul de mai jos listeaza cateva modalitati de a determina hardware-ul prezent in sistem – pas necesar pentru o configurare corecta a surselor kernelului inaintea compilarii. Aceasta presupune ca rulam deja un Linux pe acea statie; desi ar putea parea restrictiv, sa nu uitam ca exista astazi atatea variante de live CD/DVD care, odata bootate, ne pot oferi informatiile necesare despre statia in cauza, chiar daca aceasta nu contine deja o instalare de Linux.

Procesor	cat /proc/cpuinfo
Memorie	cat /proc/meminfo
Dispozitivele de pe bus-ul PCI	lspci
Dispozitivele USB	lsusb
Hardware instalat	lshw SAU lshw -X (daca lshw e instalat)
Ierarhia de drivere folosite de catre un dispozitiv – ex: hdd	udevadm info -a -n /dev/hda|grep DRIVER
Drivere folosite - alternativa	lspci -v

Ultima comanda profita de faptul ca avem deja un kernel functional si listeaza toate driverele necesare dispozitivului cerut. In masura in care comanda afiseaza mai multe nume de drivere, toate acestea vor trebui incluse in kernel in etapa de configurare a surselor.

14.2.8.2. Optiuni si drivere vitale

Exista cateva drivere sau facilitati din kernel a caror absenta determina noul kernel sa nu poata boota. Nu uitati sa includeti:

- driver pentru controllerul de hard disk – toata lista de drivere raportata de comanda udevinfo aplicata hard-disk-urilor (vezi tabelul de mai sus)
- drivere pentru tipurile de sisteme de fisiere prezente pe partitiile Linux (ext2, ext3, reiserfs etc.)
- arhitectura procesorului trebuie sa corespunda cu cea raportata in */proc/cpuinfo*

14.2.8.3. Setare local version

Modulele de kernel sunt instalate in /lib/modules/*nume_complet_kernel*, iar numele complet este format din versiunea kernel-ului concatenata cu asa-numitul *local version.* Local version este un string ce poate fi setat in etapa de configurare a surselor in vederea compilarii. Setarea sa este importanta atunci cand compilam mai multe variante ale aceleiasi versiuni de kernel, pentru ca astfel se preintampina suprascrierea modulelor la fiecare make_install, permitand variantelor aceleisai versiuni de kernel sa coexiste.

14.3. BIBLIOGRAFIE

- Kernel HOWTO: http://www.faqs.org/docs/Linux-HOWTO/Kernel-HOWTO.html
- Ubuntu Kernel Compile: https://help.ubuntu.com/community/Kernel/Compile
- Initial ramdisk: http://www.ibm.com/developerworks/linux/library/l-initrd.html

14.4. ANEXA 1 (informativa): fisierul de configurare GRUB legacy

14.4.1. Prezentare

Fisierul de configurare al GRUB se gaseste de obicei in /boot/grub/menu.lst. GRUB isi citeste acest fisier la fiecare bootare (spre deosebire de LILO, care isi foloseste fisierul de configurare numai la rescrierea MBR-ului). Configurarile efectuate in menu.lst determina atat continutul meniului afisat utilizatorului la bootarea computerului (sau absenta acestui meniu!), cat si modalitatea exacta de bootare a fiecarui sistem de operare.

Fisierul menu.lst reprezinta de fapt un script, deoarece contine un ansamblu de comenzi ce pot fi date si in linia de comanda GRUB (prezentata mai jos in cadrul materialului). Comenzile disponibile se impart in doua categorii:

- comenzi generale (care nu se aplica doar unui anumit sistem de operare din meniul GRUB)
- comenzi per-element de meniu. Fiecarui element al meniului ii corespunde in menu.lst o grupare de comenzi ce debuteaza cu **title**, comenzile ce-i urmeaza aplicandu-se numai acelei intrari din meniu

14.4.2. Denumiri de hard disk-uri, partitii si elemente de meniu

In fisierul de configurare GRUB, totul se numeroteaza de la 0:

- **hard-disk-uri:** denumirile lor sunt (hdX), unde X incepe de la 0, ce desemneaza hard-disk-ul primar. Astfel, hard-disk-ul denumit in sistemul de fisiere Linux /dev/hda va fi referit in GRUB ca (hd0)
- **partitii:** sunt denumite (hdX,Y), unde X si Y incep de la 0. Daca ne referim la partiile de pe hard-disk-ul primar, atunci (hd0,0) pana la (hd0,3) se refera la partitiile primare, iar de la (hd0,4) in sus avem partitiile logice
- **elemente de meniu:** fiecare grupare de instructiuni ce incepe cu title primeste un numar incepand cu 0, in ordinea aparitiei in fisierul ce configurare. Acest numar poate fi folosit in cadrul fisierului pentru a referi un anumit element al meniului (vezi mai jos comanda *default*)

14.4.3. Comenzi generale

Sunt comenzi folosite pentru configurarea aspectului si comportamentului meniului afisat la bootarea computerului. Iata doua exemple:

- comanda **timeout** specifica numarul de secunde cat GRUB afiseaza meniul asteptand ca utilizatorul sa aleaga un sistem de operare. Daca pana la scurgerea acestui interval de timp este apasata o tasta, timeout-ul este anulat; in caz contrar, GRUB booteaza sistemul de operare default (vezi comanda urmatoare)
- comanda **default** precizeaza elementul ce va fi selectat din oficiu la afisarea meniului, si care va fi bootat automat daca utilizatorul nu apasa nici o tasta in intervalul specificat cu parametrul *timeout*. Elementele de meniu sunt numerotate de la 0, in ordinea in care apar in fisierul de configurare

Nota: *sunt disponibile diferite alte comenzi - pentru stabilirea culorilor meniului, a imaginii de fundal etc.*

14.4.4. Comenzi per-element

Acestea sunt comenzile ce se aplica numai cate unui element din meniul GRUB. Fisierul de configurare contine, pentru fiecare element al meniului, o sectiune de comenzi ce incepe cu **title**. Argumentul comenzii title este chiar textul ce va fi afisat in meniul GRUB corespunzator acelui element. Comenzile ce urmeaza lui title se aplica doar elementului respectiv.

In functie de rolul pe care il joaca GRUB pentru sistemul de operare corespunzator unui element din meniul sau, setul de comenzi per-element difera:

- atunci cand GRUB nu cunoaste sistemul de fisiere in cauza, el va opera numai ca boot manager, predand apoi stafeta catre boot loaderul acelui sistemul de operare. Iata comenzile cel mai des folosite in acest scop:
  - **rootnoverify (hdX,Y)** – indica lui GRUB pe ce partitie se afla sistemul de operare dorit, cerandu-i sa nu incerce sa recunoasca sistemul de fisiere de pe acea partitie. (X si Y se inlocuiesc cu numerele corespunzatoare pentru partitia respectiva)

  - **chainloader +1** – instruieste GRUB sa incarce primul sector de pe partitia specificata anterior si sa predea stafeta codului aflat acolo, in ipoteza in care la inceputul acelei partitii rezida boot loaderul sistemului de operare ce se doreste bootat
  - **makeactive** – seteaza partitia specificata ca activa (util in cazul sistemelor de operare care se asteapta sa booteze de pe partitia activa – ex Windows)
- atunci cand GRUB joaca rolul de boot loader (cazul Linux-ului), trebuie sa i se indice locatia kernelului si eventualele optiuni suplimentare legate de bootarea acestuia. Comenzi principale:
  - **root (hdX,Y)** – indica partitia pe care se afla directorul /boot, ce contine kernelul Linux
  - **kernel** ***/cale/catre/fisier_kernel*** **root=/dev/hdaX** ***alte_optiuni_kernel*** – indica locatia kernelului in sistemul de fisiere al partitiei specificate anterior. Tot ce ce urmeaza caii catre kernel sunt parametri pasati kernelului; optiunea *root* stabileste partitia ce contine radacina sistemului de fisiere, ce trebuie montata de catre kernel pentru a putea continua initializarea sistemului de operare
  - **initrd** ***/cale/catre/imagine.img*** – stabileste locatia fisierului ce joaca rolul de *initial RAM disk* – continutul acestui fisier se expandeaza in memorie la bootare sub forma unei pseudo-partitii ce contine module de kernel indispensabile bootarii (ex: driver pentru controllerul SATA sau RAID la care se conecteaza hard-disk-urile sistemului)

Iata un exemplu de fisier de configurare:

```
timeout 10
default 1

title Windows
rootnoverify (hd0,0)
makeactive
chainloader +1

title Mandriva 2008.1
root (hd0,4)
kernel /boot/vmlinuz root=/dev/sda5
initrd /boot/initrd.img
```

Meniul GRUB va avea doua elemente – "Windows" si "Mandriva 2008.1"; cel de-al doilea este selectat din oficiu la aparitia meniului (gratie comenzii default) si va fi bootat automat in 10 secunde (efectul lui timeout) daca utilizatorul nu apasa nici o tasta. Sistemul de operare Windows se afla pe prima partitie primara, care va fi setata ca activa la bootare. Distributia Linux se afla pe prima partitie logica a hard-disk-ului primar (/dev/sda5, sau (hd0,4) in terminologie de GRUB). Kernelul se numeste vmlinuz si se afla in /boot; el foloseste o imagine de RAM disk aflata in fisierul initrd.img din acelasi director.