



LPI-1. Examen 101. Objectius 101.2 i 102.2





LPI 101.2. Arrencada del sistema

LPI 102.2. Instal·lar un gestor d'arrencada

http://acacha.org/mediawiki/index.php/LPI_101.2

http://acacha.org/mediawiki/index.php/LPI_102.2

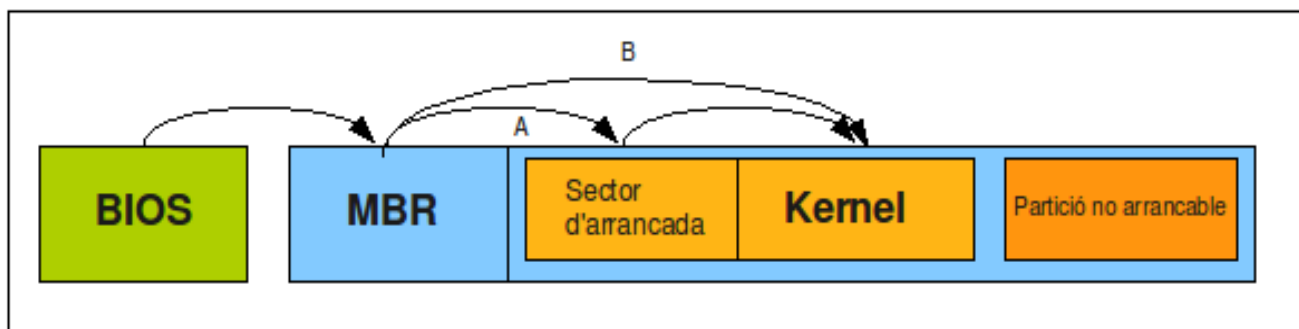
Objectius

| 101.2. Arrencada del sistema | |
|---|--|
|  | <ul style="list-style-type: none"> ▪ Objectiu: Els candidats han de ser capaços de guiar al sistema a través del procés d'engegada. ▪ Pes: 3 |
|  | <p>Àrees Clau de Coneixement:</p> <ul style="list-style-type: none"> ▪ Proporcionar les ordres típiques per al gestor d'arrancada i les opcions per al nucli del sistema al moment d'arrencar el sistema. ▪ Demostrar coneixement de la seqüència d'arrencada des de la BIOS fins al seu acabament. ▪ Comprovar els esdeveniments produïts durant l'inici del sistema als fitxers de registre. |
|  | <p>La següent és una llista parcial de fitxers, termes i utilitats utilitzades:</p> <ul style="list-style-type: none"> ▪ /var/log/messages ▪ dmesg ▪ BIOS ▪ bootloader ▪ kernel ▪ init |
|  | <p>Apunts: LPI 101.2. Arrencada del sistema</p> |

Altres objectius

| 102.2. Instal·lar un gestor d'arrencada | |
|---|---|
|  | <ul style="list-style-type: none"> ▪ Objectiu: Els candidats han de ser capaços de seleccionar, instal·lar i configurar un gestor d'arrencada. ▪ Pes: 2 |
|  | <p>Àrees Clau de Coneixement:</p> <ul style="list-style-type: none"> ▪ Proporcionar punts d'arrencada alternatius i opcions de còpia de seguretat. ▪ Instal·lar i configurar un gestor d'arrencada com GRUB. ▪ Interaccionar amb el gestor d'arrencada. |
|  | <p>La següent és una llista parcial de fitxers, termes i utilitats utilitzades:</p> <ul style="list-style-type: none"> ▪ /boot/grub/menu.lst ▪ grub-install ▪ MBR ▪ superblock ▪ /etc/lilo.conf ▪ lilo |
|  | <p>Apunts: LPI 102.2. Instal·lar un gestor d'arrencada</p> |

Procés d'arrencada



♦ Fases

- ♦ 1a Fase. BIOS
- ♦ 2a Fase. MBR (Gestor d'arrencada)
 - 3a Fase. Sector d'arrencada 1a partició
 - 3b Fase. Nucli del sistema operatiu
- ♦ 4a Fase. Procés scheduler
- ♦ 5a Fase. Procés init



Procés d'arrancada detallat

1. S'engega l'ordinador i s'executa la **BIOS**.
2. La BIOS realitza tasques específiques d'arrancada i **configuració del maquinari**.
3. Un cop el maquinari és reconegut i executat correctament, la BIOS carrega i executa el programa **Initial Program Loader** (IPL), també conegut com **stage1** o **fase 1** del gestor d'arrancada). Aquest programa es troba al Master Boot Record (sector 0) del dispositiu seleccionat com a dispositiu d'arrancada.
4. La fase 1 del bootloader **s'executa la fase 2**. Alguns gestors d'arrancada, entremig d'aquestes fases, executant la fase **1.5, fase opcional** i que **permet tenir accés a més sistemes de fitxers**.



Procés d'arrancada detallat

5. El gestor d'arrancada sovint ofereix al usuari un **menú amb diferents opcions de càrrega**. Un cop seleccionada una opció, s'executa el sistema operatiu mitjançant la **càrrega en memòria del kernel i la seva posterior execució**. Les primeres tasques del kernel són configurar funcions bàsiques d'accés al maquinari.
6. El kernel executa la funció **start_kernel()** que realitza la majoria de la configuració del sistema (interrupcions, gestió de memòria, inicialització de dispositius, drivers, etc.)
7. El kernel **executa dos processos, l'scheduler i el procés init** de forma separada.



Procés detallat d'arrencada

8. El procés **scheduler** pren el control del sistema i és l'encarregat de gestionar els processos i la multi-tasca del sistema. **El kernel queda inactiu (idle) a l'espera de peticions d'accés a l'espai del kernel.**
9. **El procés init s'executa en espai d'usuari** i executa els **scripts d'inicialització del sistema**. Aquest scripts configuren serveis que no són del sistema operatiu amb l'objectiu de crear un entorn d'usuari.
10. Finalment es proporciona a l'usuari una **pàgina de login**, que pot ser per **línia de comandes o per entorn gràfic**. També es possible configurar l'entorn per tal que entri sense necessitat de fer login.



1a Fase. La BIOS

♦ Basic Input/Output System (BIOS)

- ♦ És el firmware del PC
- ♦ Al principi eren memòria ROM (Read Only Memory)
- ♦ Actualment són memòries PROM, EPROM o FLASH (lectura i escriptura). Sistemes actualitzables
- ♦ Memòries encastades (embedded) a la placa mare
- ♦ En PCs, la BIOS disposa d'un programa de configuració
 - Accessible per combinació de tecles durant l'arrancada (supr o F2 o F8, depèn de la BIOS)
 - Determina configuracions del maquinari del sistema (Chipsets, busos de sistema, gestió del consum energètic, seguretat d'accés al sistema, etc.)
 - **Ordre d'arrancada dels dispositius del sistema**
- ♦ BIOS a la wiki del curs



2a Fase. Gestor d'arrencada

♦ Gestor d'arrencada o boot loader

- ♦ L'aplicació que es executada per la BIOS per iniciar la càrrega del sistema.
- ♦ El gestor d'arrencada es troba al MBR del disc (primer sector). A aquesta part del gestor de l'anomena **IPL**.
- ♦ 2 gestors d'arrencada importants a Linux:
 - **Linux Loader (LILO)**
 - **Grand Unified Boot Loader (GNU GRUB)**
- ♦ LILO és més antic que GRUB. Poc a poc està sent reemplaçat per GRUB com al gestor d'arrencada més utilitzat.



3a Fase. Sector d'arrancada 1a partició

♦ Fase 2 del gestor d'arrancada

- ♦ El MBR ocupa 512 bytes, dels quals només 446 corresponen al IPL
- ♦ No hi ha prou espai per oferir un programa amb opcions “avançades” menú, editor d'opcions, etc.
- ♦ Al primer sector de la partició que conté el gestor, es troba la fase 2 del MBR
 - Alguns gestors, com GRUB, utilitzant la zona de compatibilitat de DOS (resta d'espai de la pista 0 que no és MBR) per suportar més sistemes de fitxers
- ♦ Si no hi han opcions avançades n'hi ha prou amb 446 (p. ex. El NTLDR de Windows). **Fase 3a opcional!**



3a Fase. Sector d'arrancada 1a partició

♦ Fase 2 del gestor d'arrancada

- ♦ Conté la configuració avançada del gestor:
 - Menú d'opcions
 - Paràmetres d'execució del nucli
 - ...
- ♦ A la fase 1 en prou feines cap el codi per executar la fase 2.
- ♦ Si la fase 2 està en sistemes de fitxers que no són els habituals, es requereix d'una fase 1.5 per poder llegir tots aquests sistemes de fitxers.



Fase 3b. Nucli

♦ **Detalls de l'execució del nucli**

- ♦ El gestor d'arrancada executa el nucli amb paràmetres
- ♦ Fases
 - Execució del nucli: configuració del maquinari. Executar el planificador de processos
 - Execució del disc d'inici en RAM (initrd)
 - Execució del procés init
- ♦ També podem executar nuclis des d'eines de màquines virtuals com **qemu**



Fase 4. Kernel en mode sistema

♦ **Planificador de processos (scheduler)**

- ♦ El nucli es queda en execució dormint a l'espera de peticions d'accés a l'espai de sistema
- ♦ El planificar gestiona els processos
- ♦ Des de l'espai de sistema s'executa amb un fork el procés init



Fase 5. Kernel en espai d'usuari

♦ Procés **init**

- ♦ **/sbin/init**. Primer procés. PID 1
- ♦ S'executa en espai d'usuari.
- ♦ Els processos que s'executen durant l'arrencada del sistema estan controlats pel scripts d'inicialització System V
- ♦ Es pot carregar un altre procés mitjançant l'opció del nucli **init**.

Història d'UNIX, GNU i Linux.

♦ Finals anys 60

- ♦ Primer sistema Unix.

♦ Anys 70

- ♦ Les empreses venien grans màquines i el programari el regalaven.

♦ Anys 80

- ♦ S'abarateixen les despeses en maquinari i es comença a comercialitzar el programari.
- ♦ Richard Stallman, desenvolupador del MIT va veure que el secretisme en el desenvolupament del programari acabaria perjudicant a tothom i decideix crear el projecte GNU.



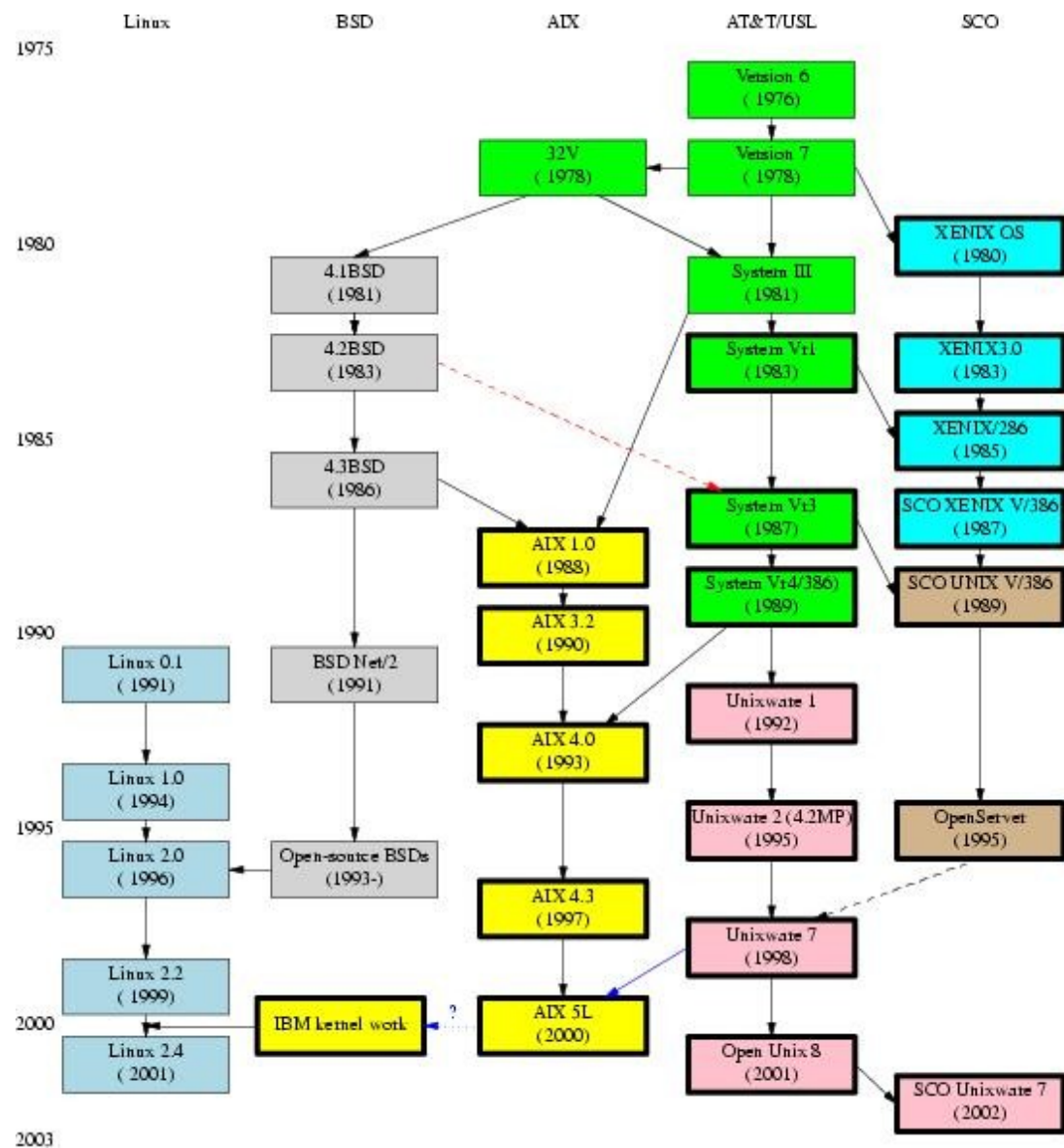
Història d'UNIX, GNU i Linux.

♦ Unix®

- ♦ És un sistema operatiu portable, multitasca i multiusuari desenvolupat a finals del 90 per un grup de treballadors dels laboratoris Bell de AT&T (Ken Thompson, Dennis Ritchie y Douglas McIlroy)
- ♦ Avui en dia Unix s'utilitza per referir-se
 - Família UNIX: Sistemes operatius Linux-Like. UNIX és refereix a una família de sistemes operatiu que comparteixen uns criteris de disseny i interoperabilitat comuns.
 - Més de 100 sistemes operatius desenvolupats en 20 anys
 - UNIX - el sistema operatiu original
 - UNIX - la marca registrada per "The Open Group"



Història d'UNIX, GNU i Linux.



Història d'UNIX, GNU i Linux.

♦ GNU



- ♦ ñu (Connchaetes) és un antílop de l'Àfrica del Sur.
- ♦ Acrònim recursiu: **GNU is Not Unix.**
- ♦ L'objectiu de GNU és crear un sistema operatiu basat completament en programari lliure que permetes als usuaris tornar a ser lliures com en els anys 60 i 70.
- ♦ Va ser anunciat el 27 de setembre de 1983 per Richard Stallman.
- ♦ El març de 1985 és publica el **GNU Manifesto**.
- ♦ L'octubre de 1985 es forma la Free Software Foundation (FSF).
- ♦ Paquets GNU: GNU Compiler Collection (GCC), les GNU Binary Utilities (binutils), la shell bash, la llibreria GNU de C library (glibc) i les GNU Core Utilities (coreutils).
- ♦ Actualment encara no hi ha una versió completa de GNU (Kernel GNU Hurd)

GNU i Richard Stallman

♦ Richard Stallman

- ♦ Fundador del projecte GNU i la FSF



- ♦ Creador de l'editor Emacs, el compilador GCC i el depurador GDB.
- ♦ Actualment és la figura més influent en l'establiment d'un marc de referència moral, polític i legal per al moviment del programari lliure.
 - Contínuament esta de gira promocionant com a Showman (a la foto San GNUcio) el programari lliure.
- ♦ Inventor del concepte de Copyleft



Linux i Linus Torvalds

♦ Kernel o nucli

- ♦ Part principal del sistema operatiu encarregada d'establir la comunicació entre el sistema operatiu i el maquinari.



♦ Linus Benedict Torvalds

- ♦ Desenvolupador finlandès que va iniciar la creació del Kernel a l'abril de 1991 amb 21 anys d'edat.
- ♦ Actualment és el coordinador del projecte [Linux Kernel](#)
- ♦ Linus es va inspirar en el S.O. Minix per desenvolupar un sistema operatiu Unix-Like que pugues ser executar en un PC (Intel 80386).
- ♦ Actualment Linux també funciona en altres architectures

GNU/Linux



- ♦ **Kernel + GNU = Sistema operatiu**

- ♦ El Kernel Linux va proporcionar el que faltava a GNU per poder crear un Sistema Operatiu complet.

- ♦ **Controvèrsia GNU vs Linux**

- ♦ La FSF i el seu líder Richard Stallman reivindiquen l'ús del terme **GNU/Linux** com a terme més just a l'hora de parlar de Sistemes Operatius Linux.



Nucli Linux

♦ Nucli (aka per l'anglicisme kernel)

- ♦ És la part fonamental del sistema operatiu, el programari responsable de facilitar a les aplicacions del sistema un accés segur al maquinari
- ♦ Gestionar els recursos compartits (memòria, espai de disc, perifèrics, etc.).

Accedir directament al maquinari pot ser una tasca realment complexa per a la majoria d'aplicacions.

El nucli s'encarrega d'**oferir una interfície més senzilla** d'accés al maquinari mitjançant una **abstracció del maquinari** que permet amagar (encapsular) la complexitat del sistema i proporcionar un accés senzill i controlat al maquinari. A aquesta interfície se l'anomena **màquina virtual**.

- ♦ [Wiki del curs](#)



Nucli Linux

♦ Linux kernel

- ♦ Segons FHS, trobareu els fitxers del nucli a la carpeta **/boot**

```
$ ls -l /boot
...
-rw-r--r-- 1 root root 3491696 2009-10-20 vmlinuz-2.6.28-16-generic
-rw-r--r-- 1 root root 3890400 2009-10-16 vmlinuz-2.6.31-14-generic
-rw-r--r-- 1 root root 3892224 2009-11-10 vmlinuz-2.6.31-15-generic
```

- ♦ Comparteix la carpeta amb GRUB (/boot/grub) i altres fitxers de suport al procés d'arrencada
- ♦ Podem tenir múltiples versions del nucli instal·lades
- ♦ <http://kernel.org>



Nucli Linux

♦ Nucli actual

- ♦ Trobareu normalment un enllaç simbòlic a l'arrel del sistema

```
$ ls -l /
lrwxrwxrwx 1 root... initrd.img -> boot/initrd.img-2.6.31-15-generic
lrwxrwxrwx 1 root... initrd.img.old -> boot/initrd.img-2.6.31-14-generic
...
lrwxrwxrwx 1 root... 2009-11-26 vmlinuz -> boot/vmlinuz-2.6.31-15-generic
lrwxrwxrwx 1 root... 2009-11-02 vmlinuz.old -> boot/vmlinuz-2.6.31-14-generic
```

- ♦ Ordre file:

```
$ file /boot/vmlinuz-2.6.31-15-generic
/boot/vmlinuz-2.6.31-15-generic: Linux kernel x86 boot
executable bzImage, version 2.6.31-15-generic (buildd@rothe,
RO-rootFS, root_dev 0x801, swap_dev 0x3, Normal VGA
```

```
$ uname -a
Linux portatil 2.6.31-15-generic #50-Ubuntu SMP Tue Nov 10
14:54:29 UTC 2009 i686 GNU/Linux
```



qemu

♦ Programari per a màquines virtuals

- ♦ Molt útil per a executar nuclis de sistema operatiu sense necessitat de tornar a iniciar.

- ♦ Instal·lació:

```
$ sudo apt-get install qemu
```

- ♦ No entra a LPIC

```
$ qemu -boot c -kernel /vmlinuz -initrd /initrd.img -hda  
image.img -append "root=/dev/sda clock=pit"
```

- ♦ Si dona un error amb el muntatge de l'arrel:

```
# mount -o remount,rw /  
# vi /etc/fstab
```

- ♦ Control:

```
Ctrl+Alt+2  
(qemu) stop  
(qemu) cont  
(qemu) help  
Ctrl+Alt
```



Paràmetres del nucli del sistema

♦ Els trobareu a la documentació:

```
$ sudo dpkg -L linux-doc | grep parameters
/usr/share/doc/linux-doc-2.6.28/kernel-parameters.txt.gz
$ sudo gunzip /usr/share/doc/linux-doc/kernel-
parameters.txt.gz
$ vi /usr/share/doc/linux-doc-2.6.28/kernel-parameters.txt
```

- ♦ Infinitat de paràmetres: Els més importants
 - ro, quiet, splash, single, vga, debug i verbose, init, panic, maxcpus, selinux, raid, mem, initrd, apm, acpi, nodma
- ♦ Quins paràmetres estem utilitzant al nucli actual?:

```
$ cat /proc/cmdline
root=UUID=7f5a86b3-89b8-434c-b2cb-c66b971b247e ro
quiet splash
```




Paràmetres del nucli del sistema

- ♦ **ro:** read only. L'arrel es muntarà en mode només lectura. Només pel nucli per tal de poder fer un fsck abans de muntar amb rw segons /etc/fstab
- ♦ **quiet:** no mostra els missatges del buffer circular del nucli. Altres opcions són debug i verbose.
- ♦ **splash:** En comptes de mostrar els missatges d'arrancada del nucli, mostra una pantalla splash
- ♦ **single:** són les opcions recovery mode. Entre a single user mode del procés init
- ♦ **vga:** permet indicar la resolució de la pantalla.



Paràmetres del nucli del sistema

➤ VGA

```
$ qemu -kernel /vmlinuz -append "vga=ask"
```

```

eoposta... 200 OK
QEMU

Please visit :
. http://bochs.sourceforge.net
. http://www.nongnu.org/vgabios

cirrus-compatible VGA is detected

QEMU BIOS - build: 11/02/09
$Revision: 1.182 $ $Date: 2007/08/01 17:09:51 $
Options: apmbios pcibios eltorito rombios32

ata0 master: QEMU HARDDISK ATA-7 Hard-Disk ( 420 MBytes)
ata1 master: QEMU DVD-ROM ATAPI-4 CD-Rom/DVD-Rom

Press <ENTER> to see video modes available, <SPACE> to continue, or wait 30 sec
Mode: Resolution:  Type: Mode: Resolution:  Type: Mode: Resolution:  Type:
0 F00  80x25      UGA  1 F01  80x50      UGA  2 F02  80x43      UGA
3 F03  80x28      UGA  4 F05  80x30      UGA  5 F06  80x34      UGA
6 F07  80x60      UGA  7 301  640x480x8   VESA  8 310  640x480x15  VESA
9 311  640x480x16  VESA  a 312  640x480x24  VESA  b 303  800x600x8   VESA
c 313  800x600x15  VESA  d 314  800x600x16  VESA  e 315  800x600x24  VESA
f 305  1024x768x8  VESA  g 316  1024x768x15 VESA  h 317  1024x768x16 VESA
i 318  1024x768x24 VESA  j 307  1280x1024x8 VESA  k 319  1280x1024x15 VESA
l 31A  1280x1024x16 VESA

Enter a video mode or "scan" to scan for additional modes: _
  
```



Paràmetres del nucli del sistema

- ♦ **init:** Indica quin és el primer procés a executar. Per defecte /sbin/init

```
$ qemu -kernel /boot/vmlinuz-2.6.31-15-generic -initrd  
/boot/initrd.img-2.6.31-15-generic -hda image.img -append  
"init=/bin/bash"
```

- ♦ **initrd:** indica quin és el fitxer ram disk d'inici
- ♦ **panic:** que fer en cas de tenir un kernel panic
- ♦ **maxcpus:** nombre màxim de cpus
- ♦ **selinux:** activa o desactiva selinux
- ♦ **raid:** paràmetres del raid



Paràmetres del nucli del sistema

- ♦ **mem**: Permet indicar la memòria a utilitzar.
- ♦ **apm=off**: No utilitzis **Advanced Power Managment**
- ♦ **acpi=off** i **noacpi**: No utilitzar **Advanced Configuration and Power Interface**
- ♦ **nodma**: No utilitzis **Direct Memory Access**
- ♦ **hdx=noprobe**: permet “desconnectar” un disc dur.
- ♦ **ether**: permet configurar la targeta de xarxa
- ♦ ...



Compilació/instal·lació d'un nou nucli

◆ Exemple pas a pas Ubuntu/Debian

◆ Creació dels paquets Debian:

```
$ sudo apt-get install kernel-package libncurses5-dev fakeroot wget bzip2
$ cd /usr/src
$ wget http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.31.6.tar.bz2
$ sudo tar xjvf linux-2.6.31.6.tar.bz2
$ sudo ln -s linux-2.6.31.6 linux
$ cd /usr/src/linux
$ sudo cp /boot/config-`uname -r` ./config
$ sudo make menuconfig
$ sudo make-kpkg clean
$ fakeroot make-kpkg --initrd --append-to-version=-custom kernel_image
kernel_headers
$ cd /usr/src
$ dpkg -i linux-headers-*.deb
$ dpkg -i linux-image-*.deb
```



REISUB

♦ Aturada gentil del nucli. Magic SysRq key

ALT+ "Impr Pant" + R + E + I + S + U + B

- ♦ S'ha de mantenir apretat Alt+Impr Pant mentre s'escriu lentament REISUB

```
ALT + "Impr Pant" + R: Torna el control al teclat (Raw)
ALT + "Impr Pant" + E: Envia a tots els processos la senyal TERM
ALT + "Impr Pant" + I: Envia a tots els processos la senyal kill.
ALT + "Impr Pant" + S: Sincronitza el disc dur (Sync)
ALT + "Impr Pant" + U: Desmuntat tots els sistemes de fitxers (Unmount)
ALT + "Impr Pant" + B: Torna a iniciar l'ordinador (reBoot)
```

Raising Elephants Is So Utterly Boring



sysrq

♦ Combinació de tecles **Alt+Impr Pant**

- ♦ També accessible des de **/proc**. Forçar un reboot

```
$ echo b > /proc/sys/kernel/sysrq //Equivalent a Ctrl+Impr Pant+ B
```

- ♦ Pot estar desactivat

```
$ cat /etc/sysctl.conf | grep sysrq  
kernel.sysrq = 0
```

```
// Modifiqueu a 1 i l'activeu amb:  
$ sudo sysctl -p
```

- ♦ Es poden consultar els esdeveniments sysrq a dmesg
- ♦ Sysrq key a la wikipedia

sysrq

| Action | QWERTY | Dvorak | AZERTY |
|---|---------------------------|---------------------------|--|
| Set the console log level, which controls the types of kernel messages that are output to the console | 0 through 9 | 0 through 9 | 0 through 9 (without using shift) |
| Immediately reboot the system, without unmounting partitions or syncing | b | x | b |
| Reboot kexec and output a crashdump | c | j | c |
| Display all currently held Locks | d | e | d |
| Send the SIGTERM signal to all processes except init (PID 1) | e | . | e |
| Call oom_kill, which kills a process to alleviate an OOM condition | f | u | f |
| When using Kernel Mode Setting, provides emergency support for switching back to the kernel's framebuffer console ^[3] | g | i | g |
| Output a terse help document to the console Any key which is not bound to a command should also do the trick | h | d | h |
| Send the SIGKILL signal to all processes except init | i | c | i |
| Kill all processes on the current virtual console (Can be used to kill X and svgalib programs, see below) This was originally designed to imitate a Secure Access Key | k | t | k |
| Show a stack backtrace of all currently executing tasks | l | n | l |
| Output current memory information to the console | m | m | , |
| Reset the nice level of all high-priority and real-time tasks | n | b | n |
| Shut off the system | o | r | o |
| Output the current registers and flags to the console | p | l | p |
| Display all active high-resolution timers and clock sources. | q | ' | a |
| Switch the keyboard from raw mode, the mode used by programs such as X11 and svgalib , to XLATE mode | r | p | r |
| Sync all mounted filesystems | s | o | s |
| Output a list of current tasks and their information to the console | t | y | t |
| Remount all mounted filesystems in read-only mode | u | g | u |
| Output Voyager SMP processor information | v | k | v |
| Display list of blocked (D state) tasks | w | , | z |



Arrencada. Fitxers de log

- ♦ **Missatge durant l'arrencada del sistema**
 - ♦ 1a Fase: Execució del nucli. dmesg
 - **1a:** Execució del nucli
 - **1b:** Execució del initrd
 - ♦ 2s Fase: Resta de l'arrencada
 - **Fitxers de registre: syslog**
 - **/var/log/messages**
 - **Boot log: /var/log/boot**
 - Altres fitxers de syslog
 - ♦ Cal tenir en compte les opcions quiet i splash del nucli del sistema operatiu



dmesg

♦ Consulta del kernel ring buffer

```
$ man dmesg  
...  
dmesg - print or control the kernel ring buffer
```

- ♦ El kernel, guarda tots els missatges que genera en una memòria intermèdia (buffer o cache) circular o en anell.
- ♦ Aquest missatges són els que es mostren per les consoles virtuals durant l'arrancada del sistema.
- ♦ L'arrancada del sistema pot ser un procés molt ràpid, si voleu consultar a posteriori aquests missatges, podeu utilitzar **dmesg**.
- ♦ Ordre proporcionada pel paquet **util-linux**



dmesg

♦ Exemple

```
$ dmesg | more
[    0.000000] BIOS EBDA/lowmem at: 0009f800/0009f800
[    0.000000] Initializing cgroup subsys cpuset
[    0.000000] Initializing cgroup subsys cpu
[    0.000000] Linux version 2.6.28-15-generic
(buildd@rothera) (gcc version 4.3.3 (Ubuntu 4.3.3-5ubuntu4) )
#52-Ubuntu SMP Wed Sep 9 10:49:34 UTC
2009 (Ubun
tu 2.6.28-15.52-generic)
[    0.000000] KERNEL supported cpus:
[    0.000000]   Intel GenuineIntel
[    0.000000]   AMD AuthenticAMD
[    0.000000]   NSC Geode by NSC
[    0.000000]   Cyrix CyrixInstead
[    0.000000]   Centaur CentaurHauls
[    0.000000]   Transmeta GenuineTMx86
[    0.000000]   Transmeta TransmetaCPU
[    0.000000]   UMC UMC UMC UMC
```



dmesg

◆ Opcions:

- ◆ **-c**: Esborra el ring buffer després de mostrar-lo.
- ◆ **-r**: Mostra el buffer tal com és (en cru, raw).
- ◆ **-sbufsize**: La mida per defecte del buffer és 16392. Si s'ha modificat la mida del buffer, amb aquesta opció podeu veure el buffer complet.
- ◆ **-nlevel**: Estableix el nivell dels missatges a mostrar.
 - Per exemple -n 1 mostra tots els missatges excepte el missatges de panic.

NOTA: Algunes distribucions com Ubuntu, no mostren aquest missatge durant l'arrancada ja que el nucli s'executa amb l'opció `quiet`

dmesg

♦ Més exemples:

- ♦ Es pot guardar en un fitxer

```
$ dmesg > dmesg.txt
```

- ♦ Es pot buscar amb grep

```
$ dmesg | grep eth
```

- ♦ Buffer circular --> Es reutilitza. Algunes distros guarden el resultat de dmesg just després d'arrencar:

```
/var/log/dmesg
```

- ♦ El buffer està al fitxer

```
/proc/kmsg
```

```
$ ps aux | grep syslog  
root      647  0.0  0.0   1848   504 ?  
          Ss   10:11   0:00 dd bs=1  
if=/proc/kmsg of=/var/run/rsyslog/kmsg
```

NOTA: Cal tenir en compte que el nucli és el primer en executar-se durant l'arrencada del sistema i per tant tots els processos posteriors a l'execució del nucli s'han de consultar en altres fitxers de registre, com per exemple `/var/log/messages`



/var/log/messages

♦ Registres del sistema (syslog)

- ♦ Un dels principals fitxers de registre (log) és /var/log/messages.
- ♦ També s'escriuen els missatges del ring buffer del nucli. Podeu discriminar tipus de missatges amb grep:

```
$ cat /var/log/messages | grep "kernel:"
```

- ♦ **Bootlog:** Missatges de les consoles virtuals durant l'arrencada:

Ctrl+Alt+F8

```
$ sudo vi /etc/default/bootlog  
BOOTLOGD_ENABLE=Yes  
/var/log/boot
```

NOTA: Recordeu que podeu trobar missatges més antics a fitxers com messages.1, messages.2.gz, messages.3.gzm etc... depenent de la configuració de les rotacions del fitxers de log. Consulteu `logrotate`



Gestor d'arrencada

♦ **bootloader (gestor d'arrencada)**

- ♦ Petita aplicació que s'encarrega de l'engegada del sistema.
- ♦ Escollit el dispositiu d'arrencada (segons la configuració de la BIOS), si el dispositiu és un dispositiu de blocs, **s'executa un programa que es troba al primer sector del dispositiu (MBR).**

♦ **Chain Loading (canvi de context)**

- ♦ Es commuta la memòria i la CPU del firmware pel gestor d'arrencada.



Gestor d'arrencada

- ♦ **Els bootloaders més coneguts són:**
 - ♦ **Gnu/Linux:**
 - **GNU Grub** (Grand Unified Bootloader)
 - **LILO** (Linux LOader)
 - ♦ **Windows Xp:** NTLDR (NT Loader). Les versions de vista i Windows 7 utilitzen altres gestors.
 - Tots ells permeten en major o menor mida, configurar la càrrega del SO. El més habitual és poder escollir entre múltiples sistemes operatius en sistemes duals, o poder configurar el pas de paràmetres al nucli del sistema operatiu.



IPL (Initial Program Loader)

♦ PL (Initial Program Loader)

- ♦ 446 bytes que es col·loquen al MBR d'un disc dur i que contenen el codi (petit programa en hexadecimal) que fa que el MBR apunti (executi) cap a la resta de codi del gestor d'arrancada.
- ♦ Els 446 bytes són massa petits per incloure realment cap gestor d'arrancada mínimament decent. Per tant només conté el codi just i necessari per executar (en un altre zona de disc) el veritable gestor d'arrancada
- ♦ Un altre nom que s'utilitza és el de primer estadi o **first stage** o **stage1**.



IPL i altres fases

- ❖ Els IPL llegeixen la taula de particions (que es troba al mateix MBR) i dir-li al sistema on del disc dur s'ha d'anar per executar la resta del procés d'arrancada.
- ❖ La resta del procés s'anomena stage2
- ❖ **NTLDR de Windows:** apunta al sector d'arrancada de la partició activa on està instal·lat Windows i s'acaba d'executar NTLDR.
- ❖ **GRUB:** A la resta de la primera pista (sector de compatibilitat de DOS) hi ha l'stage1.5. Només utilitza els primers 21 sectors.



IPL i altres fases

- ♦ És opcional però ajuda a GRUB a entendre múltiples sistemes de fitxers i poder mostrar segons quins tipus d'errors que no es podrien mostrar en aquesta fase de l'arrancada sense aquest codi.
- ♦ La primera pista mai es formatada pels sistemes de fitxers i per tant no pertany a cap sistema operatiu.
- ♦ El stage 2 és el que interpreta el fitxer de configuració **/boot/grub/menu.lst** i mostra el menú.



GRUB

♦ GNU Grub (Grand Unified Bootloader)

- ♦ GNU GRUB és en l'actualitat (2009) el gestor d'arrancada o bootloader més utilitzat en distribucions Linux, to i que encara, també és molt utilitzat LILO.
- ♦ Gestor per a sistemes x86
- ♦ Grub llegeix els següents sistemes de fitxers:
 - ext2/ext3/ext4 (Grub2), ReiserFS, XFS de SGI, UFS, VFAT, NTFS, JFS de IBM, HFS de Apple Inc...
 - Altres gestors accedeixen directament als blocs de disc
- ♦ Ubuntu 9.10 utilitza GRUB 2. Encara no hi ha versió estable (2009)



GRUB. Stages

♦ Fases

- ♦ **Stage 1:** Conté el IPL (Initial Program Loader) de GRUB i es troba al MBR.
- ♦ **Stage 1.5:** Conté codi addicional i opcional de l'stage 1. Normalment es col·loca als primers sectors de després del MBR, a la secció de compatibilitat de DOS situada a la primera pista del disc dur.
- ♦ **Stage 2:** La resta del codi. Mostra el menú de càrrega interpretant el fitxer de configuració `/boot/grub/menu.lst`. Normalment s'instal·la al sector d'arrancada de la partició on està instal·lat Linux.
- ♦ **NOTA:** La versió 2 de GRUB introdueix més flexibilitat en aquestes fases.



Stages

♦ Stage 1

```
$ ls -la /boot/grub/stage1  
-rw-r--r-- 1 root root 512 ... /boot/grub/stage1
```

♦ Fitxer:

- És una còpia del MBR sencer

♦ Stage 1.5

♦ Suport per a llegir sistemes de fitxers

```
$ ls -l | grep stage1_5  
-rw-r--r-- 1 root root 8288 ... e2fs_stage1_5  
-rw-r--r-- 1 root root 7856 ... fat_stage1_5  
-rw-r--r-- 1 root root 8712 ... jfs_stage1_5  
-rw-r--r-- 1 root root 7352 ... minix_stage1_5  
-rw-r--r-- 1 root root 9756 ... reiserfs_stage1_5  
-rw-r--r-- 1 root root 9556 ... xfs_stage1_5
```



Stages

♦ Stage 2

```
$ ls -lah stage2  
-rw-r--r-- 1 root root 119K 2009-04-26 10:51 stage2
```

- ♦ Realitza tota la feina que realitza tota la feina de carregar el kernel a la memòria RAM de l'ordinador i arrancar-lo.
- ♦ També és el que s'encarrega de mostrar el menú.

Notació GRUB

♦ Sintaxi:

```
(device[,part-num][,bsd-subpart-letter])
```

- ♦ Device: pot ser hd (disc dur) o fd (disquet)
- ♦ [] indica opcional

```
floppy disk drive = (fd0)
First hard drive = (hd0)
second hard drive = (hd1)
...
First partition = 0
second partition = 1
...
first partition on first hard drive =(hd0,0)= /dev/hda1 in Linux
2nd partition on first hard drive   =(hd0,1)= /dev/hda2 in Linux
3rd partition on first hard drive   =(hd0,2)= /dev/hda3 in Linux
...
first partition on second hard drive=(hd1,0)= /dev/hdb1 in Linux
2nd partition on second hard drive=(hd1,1)= /dev/hdb2 in Linux
3rd partition on second hard drive=  (hd1,2)= /dev/hdb3 in Linux
```

♦ Mapa:

```
$ cat /boot/grub/device.map
(hd0)  /dev/sda
(hd1)  /dev/sdb
```



Instal·lació de GRUB

♦ Característiques

- ♦ A diferencia de LILO, no es necessari reinstal·lar GRUB al MBR cada cop que es fa un canvi a la seva configuració.
- ♦ Grub s'executa en dos fases:
 - **Primera fase:** executa grub des del MBR
 - **Segona fase:** busca els fitxers de configuració en una partició normal. Si no troba o no té accés als fitxers de configuració grub retorna la línia de comandes de GRUB.
- ♦ GRUB es pot executar des d'un CD, disquet o USB per tal d'arrancar un sistema que no funciona correctament.



Instal·lació de GRUB

◆ grub-install

Sintaxi: `grub-install [OPTION] install_device`

◆ Instal·lació al MBR d'un disc:

```
$ grub-install /dev/sda
```

- **-h, --help**: mostra l'ajuda
- **-v, --version**: Mostra la versió
- **--root-directory=DIR**: instal·la GRUB al directori especificat en comptes de instal·lar-lo a l'arrel del sistema (/boot)
- **--grub-shell=FILE**: Utilitza el fitxer FILE com a grub shell
- **--no-floppy**: no utilitza el disquet
- **--force-lba**: força a GRUB a utilitzar LBA fins i tot en BIOS amb errors.
- **--recheck**: torna a comprovar el fitxer device map (/boot/grub/device.map). Cal utilitzar aquesta opció si hem afegit/modificat discos.



Exemple de recuperació de GRUB

♦ Es pot fer muntant un LIVE-CD (o USB)

- ♦ Consulteu els dispositius muntats `$ mount`
- ♦ Cal tenir muntat el disc al que volem instal·lar grub.
- ♦ Sinó esta instal·lat el localitzem amb: `$ sudo fdisk -l`
- ♦ Suposem que la partició és /dev/sda2:

```
$ cd  
$ cd mkdir p  
$ sudo mount /dev/sda2 p
```

- ♦ Instal·leu GRUB:

```
$ sudo grub-install --root-directory=p /dev/sda
```




Instal·lació de GRUB

♦ Es pot instal·lar en una partició

```
$ grub-install /dev/sda
```

- ♦ Per exemple per combinar múltiples gestors d'arrancada (dos GRUBs, LILO+GRUB)
- ♦ Els gestors d'arrancada es poden executar entre ells amb Chain Loading



Instal·lació en una imatge de disc

♦ Utilitzant **debootstrap**:

- ♦ Opció include

```
# debootstrap --arch=amd64 --include=linux-image-server, grub  
--components=main,universe,multiverse jaunty /mnt/xen  
http://us.archive.ubuntu.com/ubuntu/
```

♦ Utilitzant **chroot**

- ♦ Pot faltar fins i tot el nucli de Linux
- ♦ Un cop dins la imatge, configurar la xarxa i executar:

```
# apt-get install grub  
# grub-install /dev/sda  
# update-grub
```



GRUB CLI

♦ GRUB Command Line Interface

- ♦ Si accedeix des del menú de GRUB prement la **tecla c**.

```
GNU GRUB version 0.95 (638K lower / 523200K upper memory)

[ Minimal BASH-like line editing is supported. For the first word, TAB
  lists possible command completions. Anywhere else TAB lists the possible
  completions of a device/filename. ESC at any time exits. ]

grub> _
```



GRUB Shell

- ♦ **Es pot accedir a la línia d'ordres de grub amb:**

```
$ sudo grub
```

- ♦ Amb **help** podeu consultar les possibles ordres
- ♦ Suporta el tabulador
- ♦ **Executar Linux des de la línia d'ordres de GRUB:**
 - No va amb la SHELL, però funciona bé amb CLI

```
grub> root (hdx,y)
grub> root (hd0,1) i tabuleu
grub> root (hd0,1) Filesystem type is ext2fs, partition type 0x83
grub> kernel /boot/vmlinuz-2.6.28-15-generic root=/dev/sdb5
grub > initrd /boot/initrd.img-2.6.28-15-generic
grub > boot
```



GRUB Shell/GRUB CLI

♦ Buscador

- ♦ Buscar particions amb kernel Linux:
- ♦ Buscar particions root:
- ♦ Es pot buscar qualsevol fitxer
- ♦ Saber la distribució que hi ha instal·lada:

```
> find /vmlinuz  
(hd0,0)  
(hd1,0)
```

```
grub> find /sbin/init  
(hd0,0)  
(hd1,0)
```

```
grub> cat (hd0,0)/etc/lsb-release
```

- ♦ Consultar els principals fitxers de GRUB:

```
grub> cat (hd0,0)/boot/grub/menu.lst  
grub> cat (hd0,0)/etc/fstab  
grub> cat (hd0,0)/boot/grub/device.map
```



GRUB Shell/GRUB CLI

♦ Executar Linux des de fitxer de configuració:

```
grub> find /boot/grub/menu.lst
(hd0,0)
(hd1,0)
grub> configfile (hd0,0)/boot/grub/menu.lst
```

♦ Chain Loading. Carregar altres gestors:

- ♦ Cal carregar la partició on està instal·lat l'altre gestor.

```
grub> root (hd0,5) Filesystem is ext2fs, partition type is 0x83
grub> chainloader +1
grub> boot
```

GRUB Shell/GRUB CLI

♦ Instalar GRUB des de la línia d'ordres:

```
grub> find /boot/grub/stage1
(hd0,1)
(hd0,5)
grub> root (hd0,5) Filesystem is ext2fs, partition type is
0x83
grub> setup (hd0,5)
Checking if "/boot/grub/stage1" exists...yes
Checking if "/boot/grub/stage2" exists...yes
Checking if "/boot/grub/e2fs_stage1_5" exists...yes
Running "embed /boot/grub/embed/e2fs_stage1_5
(hd0,5)"...failed (this is not fatal)
Running "embed /boot/grub/embed/e2fs_stage1_5
(hd0,5)"...failed (this is not fatal) Running "install
/boot/grub/stage1 d (hd0,5) /boot/grub/stage2 p
/boot/grub/menu.lst"...
succeeded Done.
grub> chainloader +1
grub> boot
```




GRUB Shell/GRUB CLI

♦ Carregar Windows

```
grub> find /boot.ini
(hd0,0)
grub> rootnoverify (hd0,0) //o només root si estas carregant Windows amb
FAT32 (Windows 95, Windows 98, Windows ME, o Windows XP Home Edition. Si no
està segur --> rootnoverify
grub> makeactive // Marca la partició com activa.
grub> chainloader +1
grub> boot
```

♦ Amb l'ordre map:

```
grub> rootnoverify (hd1,0)
grub> makeactive
grub> map (hd0) (hd1)
grub> map (hd1) (hd0)
grub> chainloader +1
grub> boot
```



GRUB. Paràmetres del nucli

♦ Sintaxi

```
kernel /camí/al/nucli_delinux root={PARTICIÓ DE ROOT} [opcions del kernel]
```

♦ Exemple en una Ubuntu actual:

```
kernel    /boot/vmlinuz-2.6.28-13-generic  
root=UUID=8b669217-d286-4a45-be7e-69c5f0f29090 ro quiet  
splash
```



Recuperar paraula de pas des de GRUB

♦ Utilitzar el paràmetre init per executar una shell

- ♦ Útil encara que no es pugui iniciar un LIVE-CD (BIOS protegida).
- ♦ Reinicieu l'ordinador. Entreu amb ESC al menú de GRUB i editeu amb la tecla e l'opció de menú amb la que arranqueu.
- ♦ Afegiu a les opcions del nucli: `init=/bin/bash`
- ♦ Premeu b (boot). Obtindreu una shell

```
# mount -o remount,rw /      // O canvieu ro per rw al boot
# mount /proc
# passwd usuari
# sync
```

| Tecla del teclat català | Tecla del teclat americà |
|-------------------------|--------------------------|
| - | , |
| = | i |
| / | - |

- ♦ Es pot protegir GRUB amb paraula de pas contra les modificacions



Configuració de GRUB.

♦ **/boot/grub/menu.lst**

♦ /boot/grub/grub.conf a Fedora

♦ Dos parts:

- Comentaris que no són comentaris! Són paràmetres per defecte

```
$ cat /boot/grub/menu.lst
...
## default num
# Set the default entry to the entry number NUM. Numbering starts from
0, and
# the entry number 0 is the default if the command is not used.
#
...
## can be true or false
# updatedefaultentry=false
## ## End Default Options ##
```

♦ Són utilitzats per update-grub, cada vegada que cal tornar a crear el menú



Configuració de GRUB

◆ Opcions de menú

- ◆ Part generada automàticament, està entre:
 - **### BEGIN DEBIAN AUTOMAGIC KERNELS LIST**
 - **### END DEBIAN AUTOMAGIC KERNELS LIST**
- ◆ Si modifiquem està part els canvis es perdran al instal·lar un nou nucli durant una actualització
- ◆ Modificacions permanent han d'anar abans o després del bloc automàtic. Altres sistemes operatius estan després:

```
title                Other operating systems:
root
# This entry automatically added by the Debian installer for a non-linux OS
# on /dev/sda1
title                Microsoft Windows XP Professional
root                 (hd0,0)
savedefault
makeactive
chainloader           +1
```



Configuració de GRUB

♦ Kernels Linux:

```
title      Ubuntu, kernel 2.6.15-27-386
root       (hd0,1)
kernel     /boot/vmlinuz-2.6.15-27-386 root=/dev/sda2 ro quiet splash
initrd     /boot/initrd.img-2.6.15-27-386
savedefault
boot

title      Ubuntu, kernel 2.6.15-27-386 (recovery mode)
root       (hd0,1)
kernel     /boot/vmlinuz-2.6.15-27-386 root=/dev/sda2 ro single
initrd     /boot/initrd.img-2.6.15-27-386
Boot

...
Altres nuclis...
```

♦ Més opcions a la [wiki del curs](#)



Configuració de GRUB

♦ Opcions per defecte:

- ♦ **default:** indica quina és l'opció de menú a iniciar per defecte
- ♦ **timeout:** temps que es mostra el menú abans d'escollir l'opció per defecte
- ♦ **hiddenmenu:** No mostra el menú, excepte si premem ESC (tenim el temps del timeout per prémer ESC)
- ♦ **color cyan/blue white/blue:** Afegir colors
- ♦ **howmany:** quants nuclis mostrar al menú
- ♦ **memtest86:** Afegir una opció per al test de RAM



Configuració de GRUB

♦ Canviar el sistema operatiu per defecte

♦ 3 opcions:

- Col·locar l'entrada de menú que ens interessa a l'inici, abans de la llista automàtica. **El paràmetre default a de valer 0**
- Editar el número de l'entrada de menú a iniciar amb el paràmetre default.
 - Té el problema de si apareixen més entrades al menu
- Utilitzar la comanda savedefault. El paràmetre default ha d'estar posat a saved:



Configuració de GRUB

♦ Posar paraula de pas

```
$ sudo grub
grub> md5crypt
Password: ****
Encrypted: $1$I/KTB/$NtsUA1ejfg0t57djGAKUN.
grub > quit

$ gksudo gedit /boot/grub/menu.lst

password $1$I/KTB/$NtsUA1ejfg0t57djGAKUN.

title                Ubuntu 9.04, kernel 2.6.28-15-generic
uuid                 8b669217-d286-4a45-be7e-69c5f0f29090
kernel               /boot/vmlinuz-2.6.28-15-generic root=UUID=...
initrd               /boot/initrd.img-2.6.28-15-generic
lock
quiet
```



GRUB 2

♦ Instal·lació

```
$ sudo apt-get install grub2
```

- ♦ Amb chainLoading s'executarà GRUB 2 des de GRUB 1
- ♦ Executeu:

```
$ upgrade-from-grub-legacy
```
- ♦ Si hi ha un error canvieu editant el menú amb, root per uuid

♦ Configuració:

- ♦ Fitxers /etc/default/grub i /etc/grub.d
- ♦ Cal aplicar els canvis amb:

```
$ sudo update-grub
```



LILO

♦ LILO (Linux LOader)

- ♦ Gestor d'arrancada utilitzat durant fa molts anys en sistemes Linux.
- ♦ LILO normalment s'instal·la al MBR del disc dur.
- ♦ LILO no depèn de cap sistema de fitxers específic.
- ♦ LILO va ser durant un anys (després de l'ús de loadlin) el gestor d'arrancada més utilitzat. Actualment (2009), la majoria de distribucions utilitzen GRUB
- ♦ També es pot instal·lar en el primer sector d'una partició en comptes del MBR



LILO

♦ Instal·lació:

```
$ sudo apt-get install lilo
```

♦ Ens avisa que caldrà executar:

- \$ sudo liloconfig
- /sbin/lilo

♦ LILO no suporta UUID

- Cal canviar /etc/fstab de UUID a notació de dispositius Linux

```
$ sudo cp /etc/fstab /etc/fstab.old
```

- No instal·leu LILO al MBR! Farem una instal·lació conjunta de GRUB i LILO (tots dos alhora)



LILO

♦ Instal·lar conjuntament amb GRUB:

- ♦ Afegiu a GRUB l'opció:

```
title          LILO
root           (hd0,0)
makeactive
chainloader    +1
```

- ♦ Afegir una opció a LILO per tornar a GRUB:

```
other=/dev/sda
label="Tornar a GRUB"
```

- ♦ Per aplicar els canvis a la configuració de LILO cal executar:

```
$ sudo lilo
...
Added Memory_Test+
Added Tornar_a_GRUB
2 warnings were issued.
```



superblock

♦ Superblock del sistema de fitxers:

- ♦ Conté dades vitals per al correcte funcionament del sistema de fitxers. Si es perd el superblock aleshores perdrem tot el sistema de fitxers. Dades:
 - El tipus de sistema de fitxers
 - La mida del sistema de fitxers
 - L'estatus (clean, dirty)
 - Informació sobre la resta de les estructures de metadades del sistema de fitxers.
- ♦ Els sistemes ext mantenen còpies de seguretat repartides pel disc dur. Les podeu consultar amb:



superblock

```
# dumpe2fs /dev/sda1 | grep -i superblock
Primary superblock at 0, Group descriptors at 1-1
Backup superblock at 32768, Group descriptors at 32769-
32769
Backup superblock at 98304, Group descriptors at 98305-
98305
Backup superblock at 163840, Group descriptors at
163841-163841
...
```

- ♦ També es pot utilitzar: `$ mke2fs -n /dev/sda3`
- ♦ Es pot recuperar un sistema amb errors al superbloc:

```
$ sudo e2fsck -f -b 8193 /dev/sda3
```



Reconeixement 3.0 Unported

Sou lliure de:



copiar, distribuir i comunicar públicament l'obra



fer-ne obres derivades

Amb les condicions següents:



Reconeixement. Heu de reconèixer els crèdits de l'obra de la manera especificada per l'autor o el llicenciador (però no d'una manera que suggereixi que us donen suport o rebeu suport per l'ús que feu l'obra).

- Quan reutilitzeu o distribuïu l'obra, heu de deixar ben clar els termes de la llicència de l'obra.
- Alguna d'aquestes condicions pot no aplicar-se si obteniu el permís del titular dels drets d'autor.
- No hi ha res en aquesta llicència que menyscabi o restringeixi els drets morals de l'autor.

Advertiment

Els drets derivats d'usos legítims o altres limitacions reconegudes per llei no queden afectats per l'anterior
Això és un resum fàcilment llegible del **text legal** (la llicència completa).

<http://creativecommons.org/licenses/by/3.0/deed.ca>

LPIC-1. Examen 101. Objectiu 101.2 i 102.2

ICE-UPC



Autor: Sergi Tur Badenas