

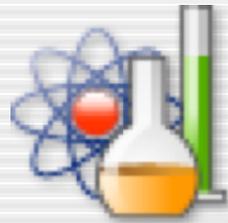
Principali caratteristiche di un sistema Linux con kernel aggiornato alla versione 2.6.X

Ferrara Francesco Saverio - 566/811 – 17 Novembre 2006

Seminario di Sistemi Operativi mod. B

Università Federico II - Facoltà di Scienze MM.FF.NN. - Informatica

Argomenti Trattati



- Supporto Multi Piattaforma
- Linux Internals
- File System
- Dispositivi di I/O
- Miglioramenti
- Sicurezza
- Allegato: Passare al Kernel 2.6

Supporto Multi Piattaforma



- Sistemi Radicati
 - Motorola, Hitachi, NEC, Axis Communication
 - Caratteristica principale: non hanno MMU
- Processori 64-Bit
 - AMD Opteron, Athlon 64, Hammer, K8
- Server
 - Principalmente SMP e NUMA

Option

Subarchitecture Type

- PC-compatible
- AMD Elan
- Voyager (NCR)
- NUMAQ (IBM/Sequent) ←
- Summit/EXA (IBM x440)
- Support for other sub-arch SMP systems with more than 8 CPUs ←
- SGI 320/540 (Visual Workstation)
- Generic architecture (Summit, bigsmg, ES7000, default)
- Support for Unisys ES7000 IA32 series

Processor family

- 386
- 486
- 586/K5/5x86/6x86/6x86MX
- Pentium-Classic
- Pentium-MMX
- Pentium-Pro
- Pentium-II/Celeron(pre-Coppermine)
- Pentium-III/Celeron(Coppermine)/Pentium-III Xeon
- Pentium M
- Pentium-4/Celeron(P4-based)/Pentium-4 M/Xeon
- K6/K6-II/K6-III
- Athlon/Duron/K7
- Opteron/Athlon64/Hammer/K8 ←
- Crusoe
- Winchip-C6
- Winchip-2
- Winchip-2A/Winchip-3
- CyrixIII/VIA-C3
- VIA C3-2 (Nehemiah)

Generic x86 support

HPET Timer Support

Symmetric multi-processing support ←

Linux Internals



- Ipercarico
- Scalabilità
- Directory Sincrone
- Thread
- Seqlocks
- Interattività
 - ◆ Futexes
 - ◆ Kernel prelazionabile
 - ◆ I/O Asincrono

Linux Internals

- Ipercarico (hyperthreading)



Linux è stato il primo OS a supportare l'ipercarico, o hyperthreading.

Hyperthreading: abilità per un singolo processore di mascherarsi in due o più processori.

Lo scheduling di Linux 2.6 è stato ottimizzato per l'hyperthreading

Linux Internals

- Scalabilità



- Numero di utenti:
 - ◆ Da 65.000 a più di *4.000.000.000*
- Numero di pid (process id):
 - ◆ Da 32.000 a *1.000.000.000*
- Dimensione file system:
 - ◆ Fino a *2 TB*



- Directory Sincrone

- Ogni cambiamento fatto in una directory sincrona viene confermato prima di ritornare il controllo alla processo:
 - ◆ Utile per software come i DBMS
 - ◆ Minor possibilità di perdere i dati
 - ◆ Peggiori prestazioni in termini di tempo

Per settare una directory sincrona:

chattr +S /path/della/directory

Per verificare se un directory è sincrona:

lsattr -d /path/della/directory

Linux Internals

- Thread



- Pthread: Posix Thread
 - ◆ Thread a livello utente
- NTPL: Native Thread Posix Library
 - ◆ Thread a livello del nucleo
 - ◆ I kernel 2.4 non supportano NTPL

Linux 2.6 gestisce i thread con un modello 1:1

Linux Internals

- Seqlocks



Proteggono variabili
condivise
dall'accesso
contemporaneo e
frequente da parte
di più thread.

```
#include <linux/seqlock.h>

seqlock_t lock;
int variabile_condivisa;

seqlock_init(&lock);

write_seqlock(&lock);
    variabile_condivisa++;
write_sequnlock(&lock);
```

Linux Internals

- Interattività



- Futexes
- Kernel prelazionabile
- I/O Asincrono

Con maggior velocità di interattività, linux 2.6 si presta a essere un buon SO anche per computer desktop.



Futexes: Fast User-Space Mutexes

In programmi multithread, possiamo avere più thread che attendono per lo stesso mutex: non c'è modo di dare la priorità ai compiti più importanti...

Con i futexes possiamo dare una priorità ai thread in attesa!



- Interattività: Kernel prelazionabile

C'è la possibilità di prelazionare il kernel:

- ◆ Soft real-time più veloce

Sono stati inseriti all'interno del kernel dei "punti sicuri" durante i quali può avvenire la prelazione:

- ◆ Ci sono ancora parti del kernel non prelazionabili



- Interattività: I/O Asincrono

- Miglior interattività con l'utente
- Separa l'operazione di I/O dalla funzione che la sta eseguendo
 - Non bloccante

Funzioni per l'I/O asincrono:
io_setup()
io_submit()
io_getevents()
io_destroy()

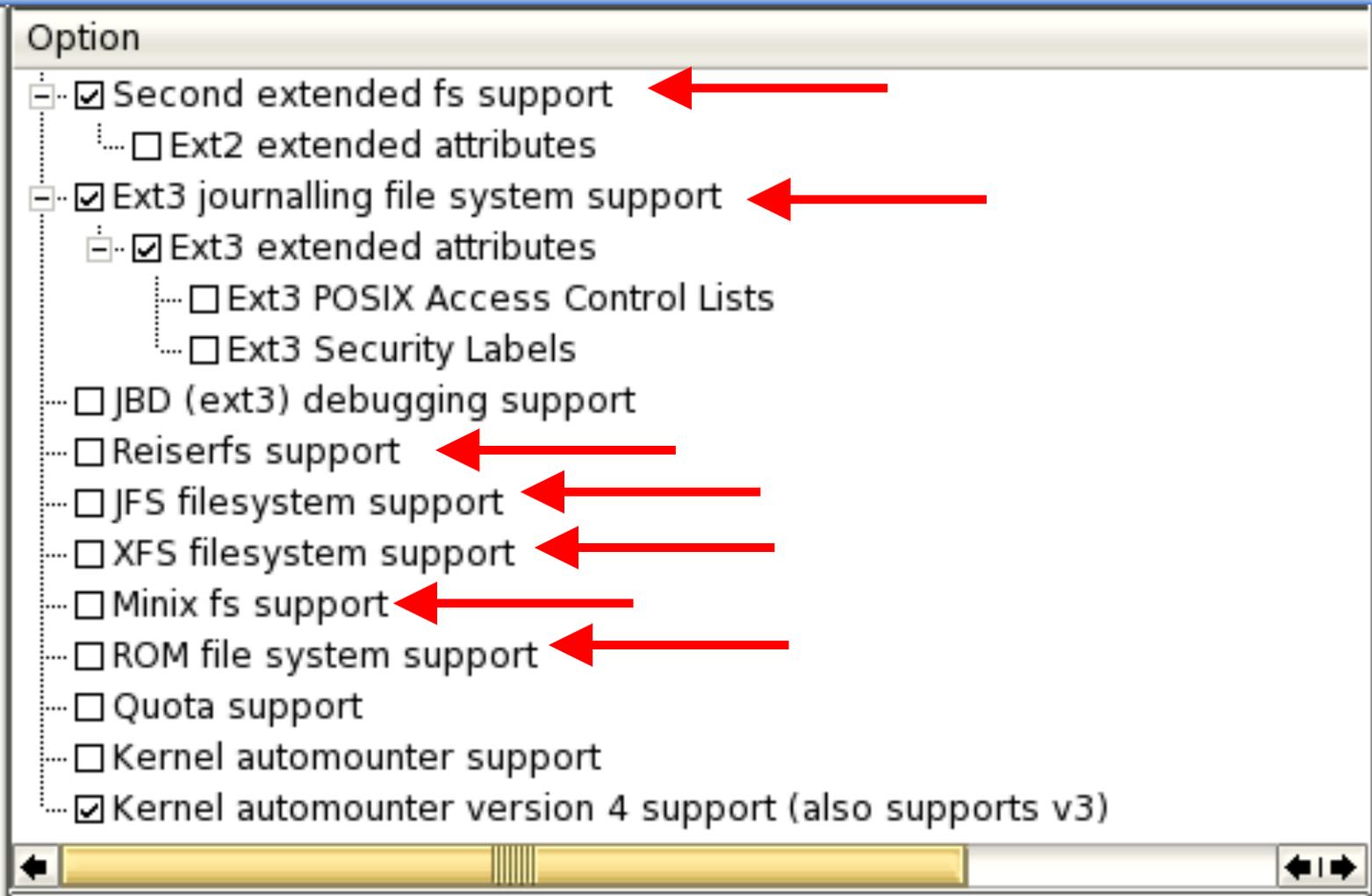
File System



- FS Utilizzati
- File Sytem /proc & /sys
- Attributi Estesi

File System

- FS Utilizzati



File System

- File System /proc & /sys



Proc e Sys non sono memorizzati fisicamente sul disco, ma si trovano in memoria

- Mantenuti dal kernel
- Contengono informazioni e statistiche

Proc: Contiene informazioni sui processi in esecuzione

Sys: Contiene una rappresentazione dell'albero dei dispositivi così come lo vede il kernel

File System

- File System /proc & /sys



saverix@localhost: /proc

```
[saverix@localhost proc]$ ls
```

```
1/      1403/  244/  9/      driver/  kmsg      self@
1025/   1404/  3/     apm     execdomains  loadavg   slabinfo
11/     1405/  391/   buddyinfo  fb         locks     splash
1100/   1406/  4/     bus/     filesystems  mdstat    stat
1125/   15/    5/     cmdline  fs/        meminfo   swaps
113/    1537/  6/     config.gz  ide/       misc      sys/
1156/   1611/  7/     cpufreq  interrupts  modules   sysrq-trigger
1165/   1613/  8/     cpuinfo  iomem      mounts@   sysvipc/
1183/   1663/  804/   crypto   ioports    net/      tty/
1277/   1666/  828/   devices  irq/       partitions  uptime
1401/   1701/  836/   diskstats  kallsyms  pci       version
1402/   2/    896/   dma      kcore     scsi/     vmstat
```

```
[saverix@localhost proc]$
```

File System

- File System /proc & /sys



E' cambiato il formato
del file
/proc/meminfo

```
xterm
saverio@virtuallinux:/proc$ uname -a
Linux virtuallinux 2.4.26 #6 Mon Jun 14 19:07:27 PDT 2004 i686 unknown unkno
saverio@virtuallinux:/proc$ pwd
/proc
saverio@virtuallinux:/proc$ cat meminfo
      total:      used:      free:  shared: buffers:  cached:
Mem: 63184896 61583360 1601536      0 2080768 26664960
Swap: 904769536 200704 904568832
MemTotal:      61704 kB
MemFree:      1564 kB
MemShared:      0 kB
Buffers:      2032 kB
Cached:      25944 kB
SwapCached:      96 kB
Active:      13308 kB
Inactive:      35148 kB
HighTotal:      0 kB
HighFree:      0 kB
LowTotal:      61704 kB
LowFree:      1564 kB
SwapTotal:      883564 kB
SwapFree:      883368 kB
saverio@virtuallinux:/proc$
```

```
saverix@localhost: /proc ((
[saverix@localhost saverix]$ cd /proc
[saverix@localhost proc]$ cat meminfo
MemTotal:      61488 kB
MemFree:      3244 kB
Buffers:      2936 kB
Cached:      29800 kB
SwapCached:      0 kB
Active:      33656 kB
Inactive:      17344 kB
HighTotal:      0 kB
HighFree:      0 kB
LowTotal:      61488 kB
LowFree:      3244 kB
SwapTotal:      511520 kB
SwapFree:      511520 kB
Dirty:      244 kB
Writeback:      0 kB
Mapped:      25188 kB
Slab:      5360 kB
Committed_AS: 23188 kB
PageTables:      332 kB
VmallocTotal: 974772 kB
VmallocUsed:   5436 kB
VmallocChunk: 968608 kB
[saverix@localhost proc]$
```

```
[saverix@localhost sys]$ pwd
/sys
[saverix@localhost sys]$ ls *
block:
fd0/  md0/  ram10/  ram13/  ram2/  ram5/  ram8/
hda/  ram0/  ram11/  ram14/  ram3/  ram6/  ram9/
hdc/  ram1/  ram12/  ram15/  ram4/  ram7/

bus:
eisa/  ide/  pci/  platform/  pnp/  scsi/  usb/

cdev:
dev.console/  dev.ptmx/  dev.tty/  dev.vc0/  major/  tty/

class:
input/  misc/  pci_bus/  scsi_generic/  tty/  usb_host/
mem/  net/  scsi_device/  scsi_host/  usb/

devices:
pci0000:00/  platform/  pnp0/  system/

firmware:
edd/

power:
disk  state
[saverix@localhost sys]$
```

File System

- Attributi Estesi



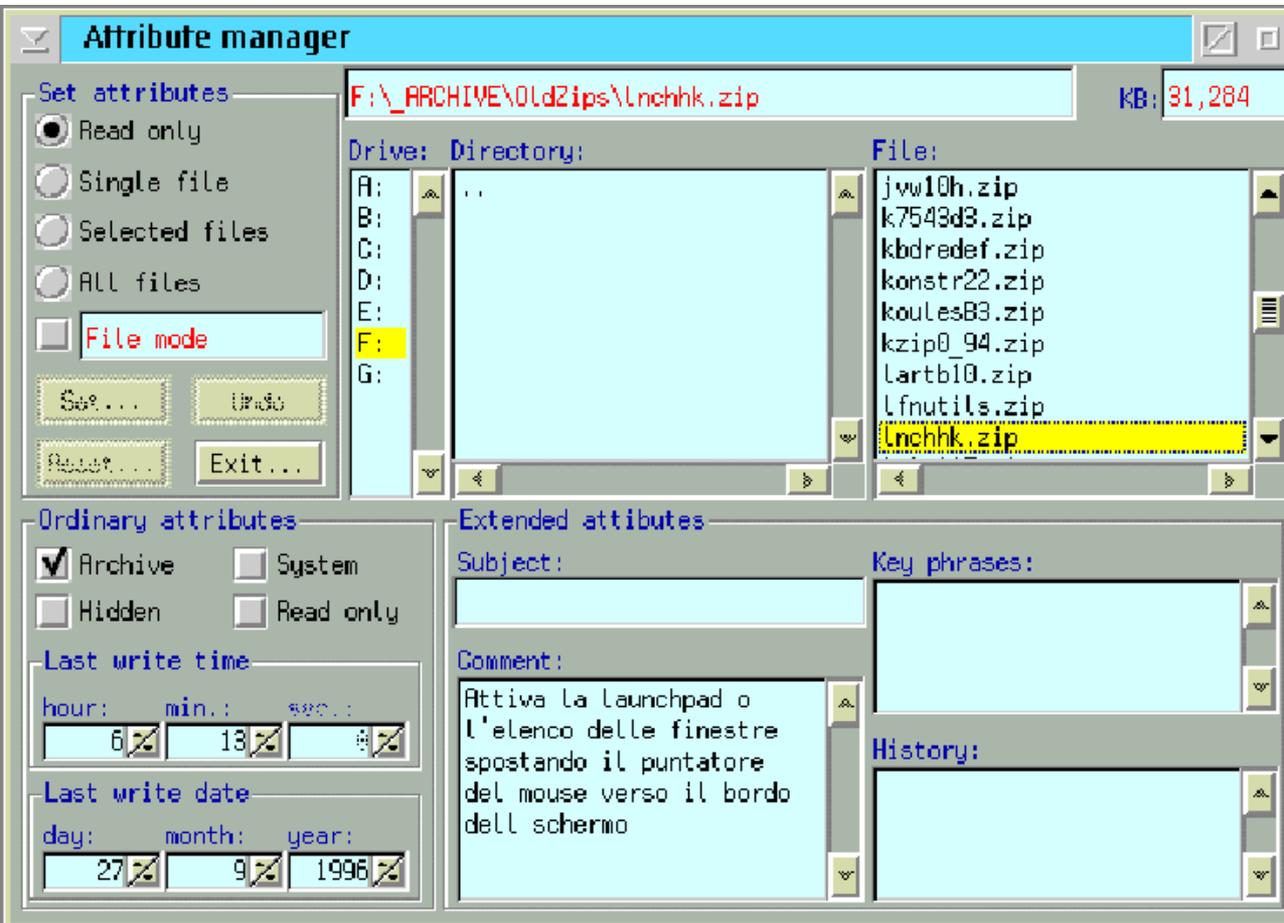
Gli attributi estesi sono supportati dai file system ext2 e ext3

Un attributi esteso è una coppia (nome, valore) che contiene informazioni su di un file

Ogni file può avere più di un attributo esteso

File System

- Attributi Estesi



Questo è un gestore degli attributi estesi per il sistema operativo OS/2

File System

- Attributi Estesi



```
#include <sys/types.h>
#include <attr/xattr.h>

char file = "seminario.ppt"
char nome = "contenuto";
char valore = "introduzione a linux 2.6";

setxattr(file, nome, valore, strlen(valore), 0);
```

Programma che imposta un attributo esteso al file "seminario.ppt". La coppia rappresentante l'attributo è: (contenuto, introduzione a linux 2.6)

Dispositivi di I/O



- Human Interface Devices
- Media

Dispositivi di I/O

- Human Interface Devices



- Tastiere con pulsanti speciali
- Mouse con doppio scroll
- Touch screen
- Joystick
- Dispositivi per non vedenti

Il kernel 2.6 è il primo che permette di essere compilato senza il sottosistema di I/O per utenti.

Option

- ..
- ... Userland interfaces
- Mouse interface (NEW)
- ... Joystick interface
- ... Touchscreen interface
- ... Event interface
- ... Event debugging
- ... Input I/O drivers
- ... Gameport support
- ... Serial port line discipline
- ... ct82c710 Aux port controller
- ... Parallel port keyboard adapter
- ... PCI PS/2 keyboard and PS/2 mouse controller
- ... Input Device Drivers
- Keyboards (NEW)
- ... Sun Type 4 and Type 5 keyboard support
- ... DECstation/VAXstation LK201/LK401 keyboard support
- ... XT Keyboard support
- ... Newton keyboard
- Mice
- ... PS/2 mouse
- ... Serial mouse
- ... InPort/MS/ATIXL busmouse
- ... Logitech busmouse
- ... IBM PC110 touchpad
- ... DEC VSXXX-AA/GA mouse and VSXXX-AB tablet
- ... Joysticks
- ... Touchscreens
- ... Misc



Dispositivi di I/O

- Media



- ALSA (Advanced Linux Sound Architecture)
 - Hardware USB e MIDI
 - Playback e registrazione Full-Duplex
 - Mixing hardware
- Supporto hardware
 - Webcams
 - Schede radio e TV
 - Videoregistratori digitali
- Per amanti dei video
 - Supporto per sintonizzatori TV e videocamere
 - V4L (Video4Linux)

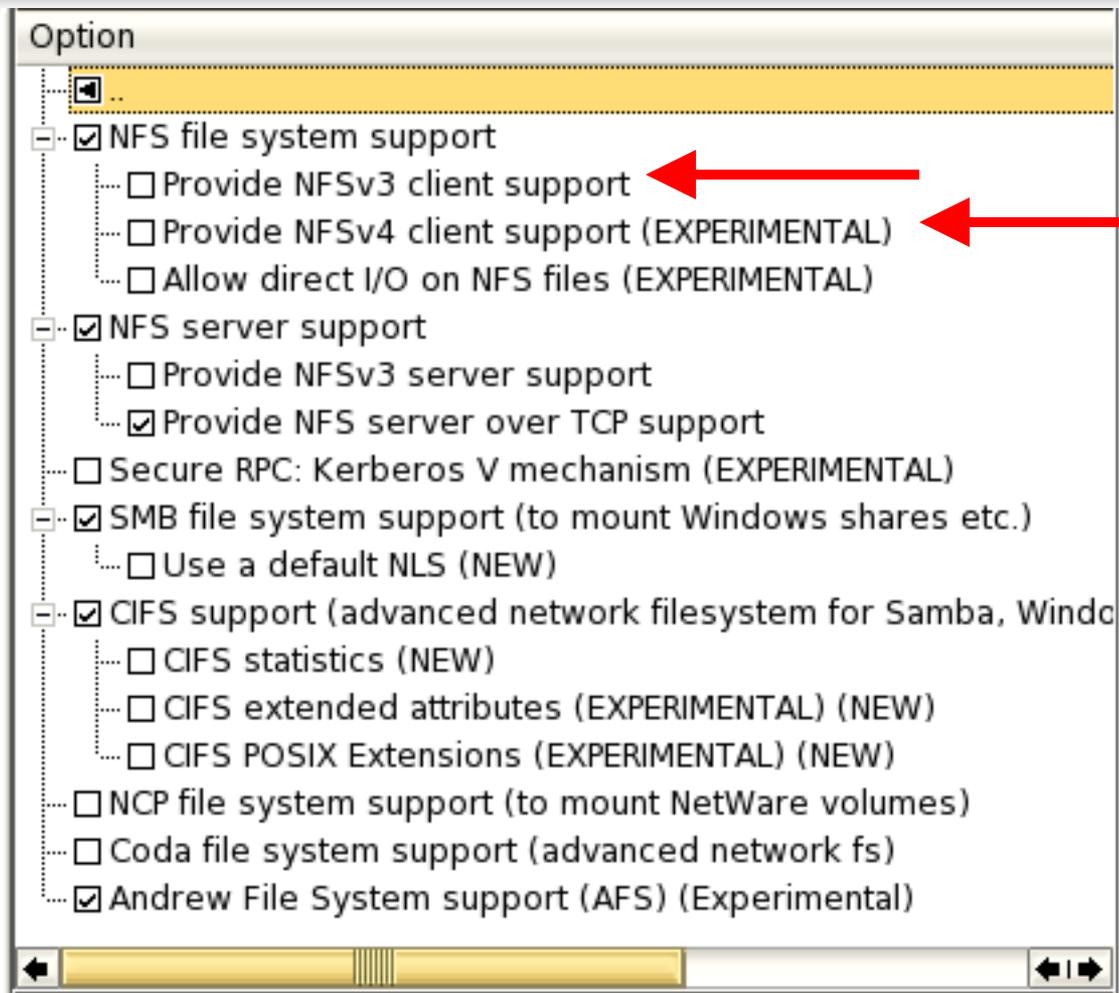


- NFS & IP
- Compatibilità con Windows
- Gestione dell'Energia
- Bus

Miglioramenti - NFS & IP



- NFSv3
- NFSv4
 - ◆ Nuovo!!!
- **Ipsec** (ip in versione crittografata)



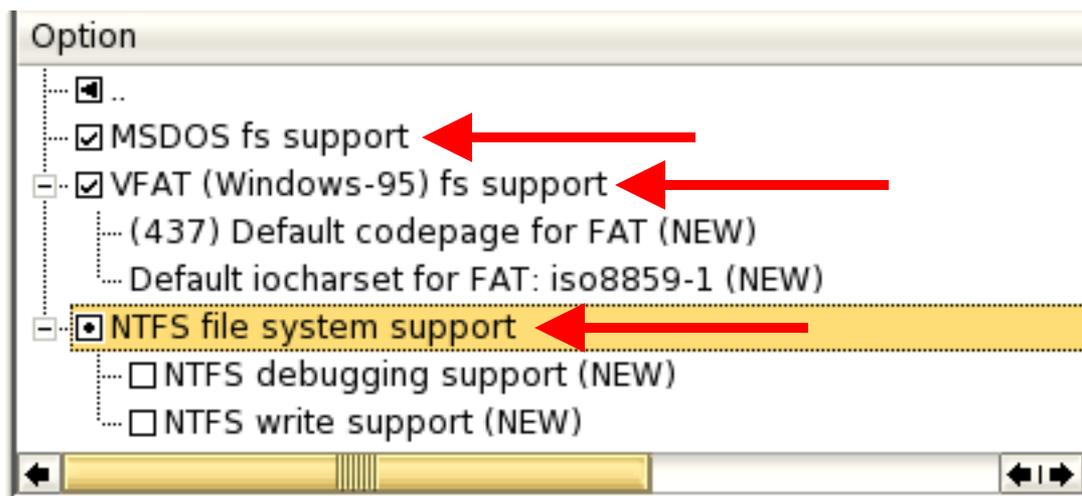
Miglioramenti

- Compatibilità con Windows



- Supporto per file system microsoft

- ◆ MSDOS
 - FAT16
- ◆ Windows 9X
 - FAT32
- ◆ Windows 2k/NT/XP
 - NTFS



Adesso Linux può scrivere anche su partizioni NTFS!

Miglioramenti

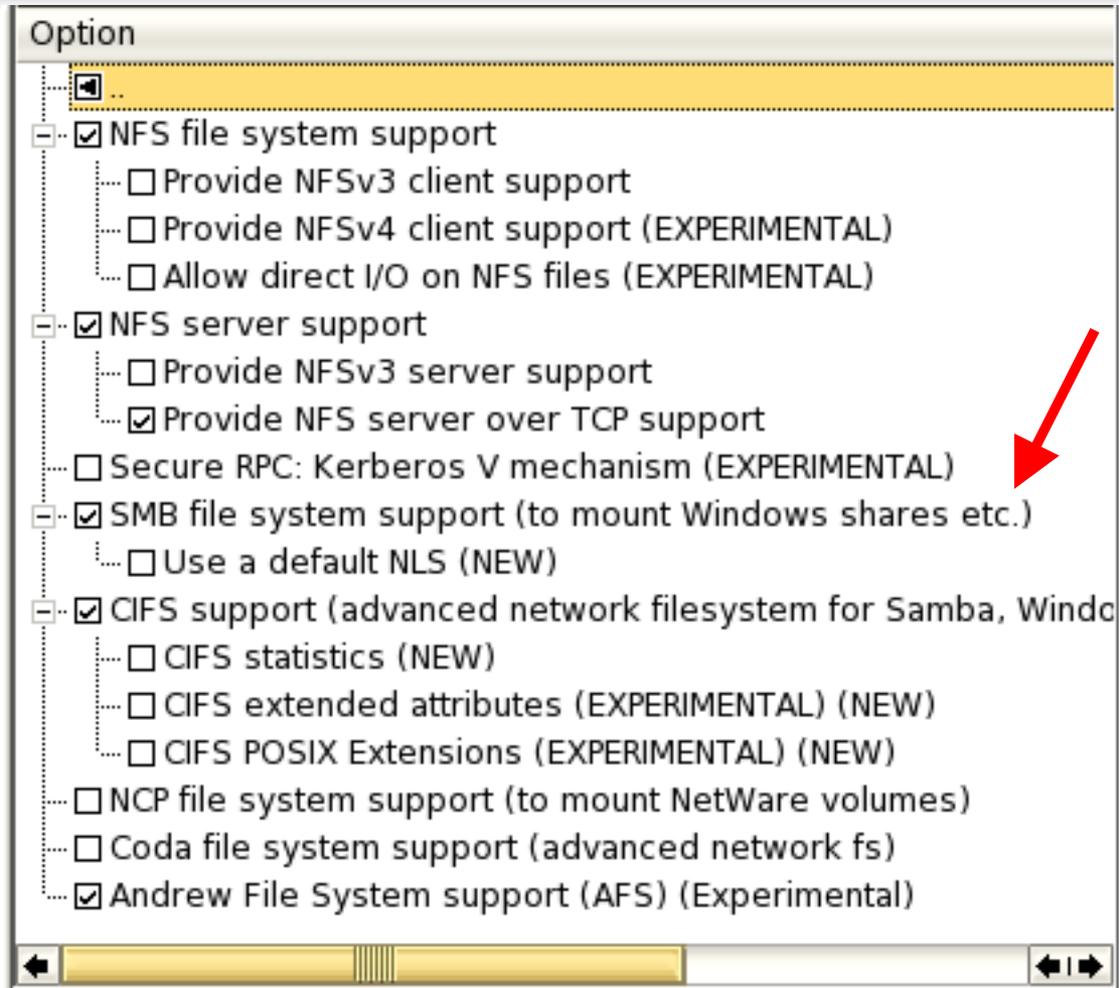
- Compatibilità con Windows



Linux 2.6 permette di montare partizioni CIFS che usano il protocollo SMB.

Questo protocollo è usato da Windows

L'applicativo che sfrutta questa caratteristica del kernel si chiama Samba

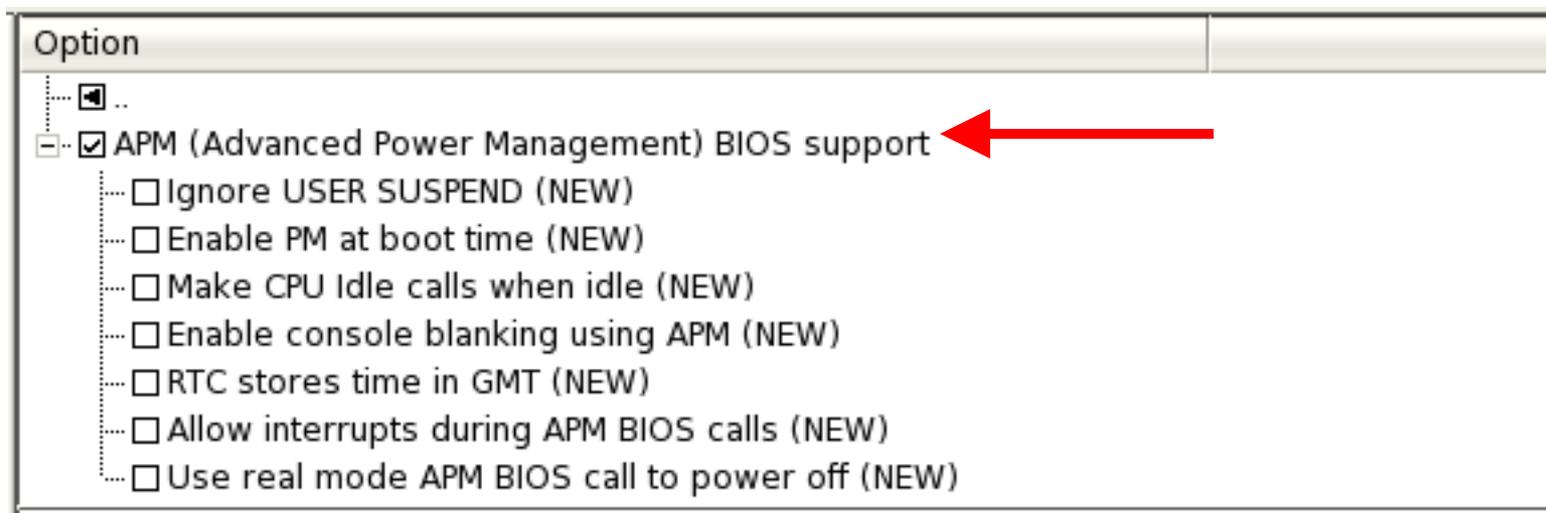


Miglioramenti

- Gestione dell'Energia



- APM (Advanced Power Management)
 - Per vecchi computer

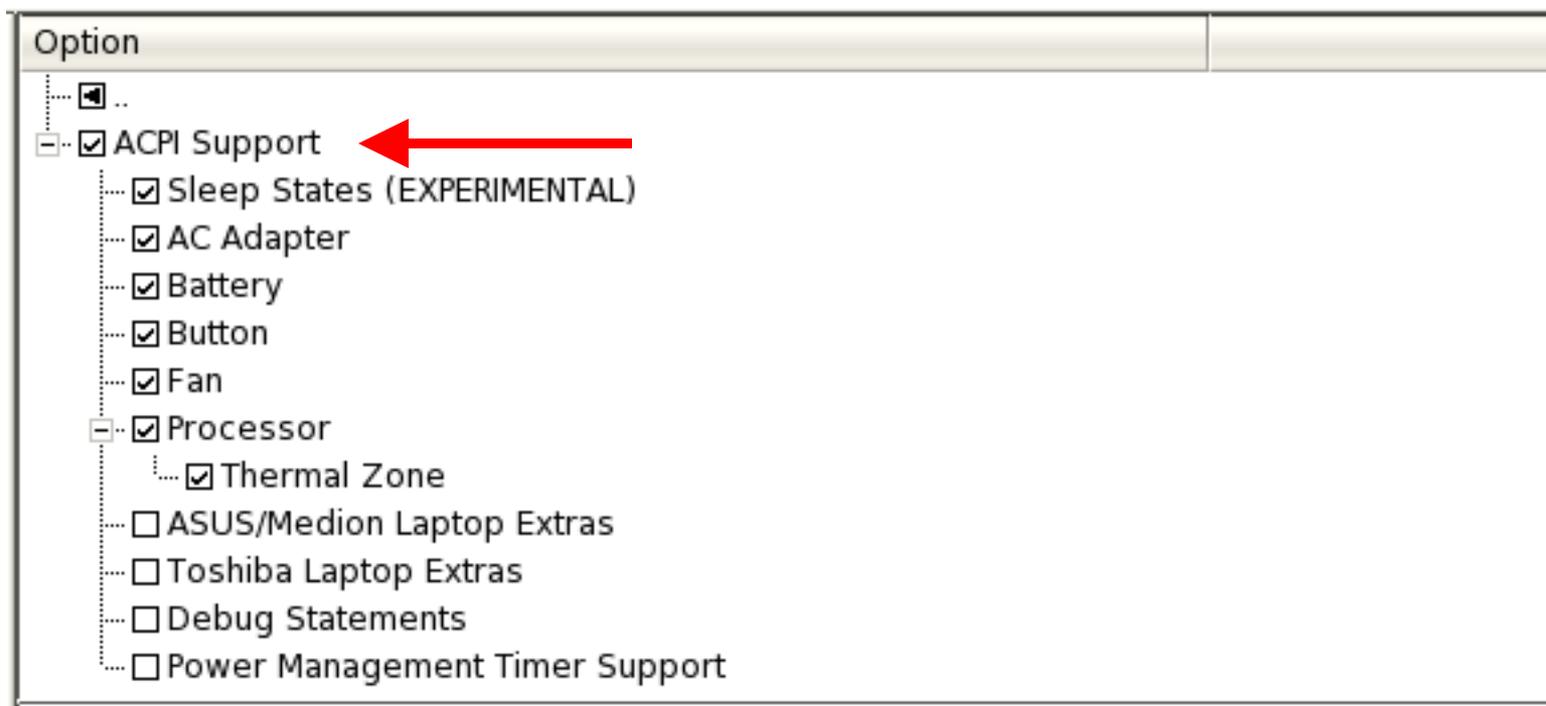


Miglioramenti

- Gestione dell'Energia



- ACPI (Advanced Control and Power Interface)
 - Presente in tutti i nuovi portatili



Miglioramenti

- Gestione dell'Energia



- Cambio di frequenza dinamico del processore
 - ◆ AMD Mobile
 - ◆ AMD Opteron
 - ◆ Intel
 - ◆ Altri mostrati nella prossima slide...

Option

- CPU Frequency scaling
- /proc/cpufreq interface (deprecated) (NEW)
- Default CPUFreq governor (NEW)
 - performance (NEW)
 - userspace (NEW)
- 'performance' governor (NEW)
- 'powersave' governor (NEW)
- 'userspace' governor for userspace frequency scaling (NEW)
- CPU frequency table helpers (NEW)
- CPUFreq processor drivers
- ACPI Processor P-States driver (NEW)
- AMD Mobile K6-2/K6-3 PowerNow! (NEW)
- AMD Mobile Athlon/Duron PowerNow! (NEW)
- AMD Opteron/Athlon64 PowerNow! (NEW)
- Cyrix MediaGX/NatSemi Geode Suspend Modulation (NEW)
- Intel Enhanced SpeedStep (NEW)
- Intel Speedstep on ICH-M chipsets (ioport interface) (NEW)
- Intel SpeedStep on 440BX/ZX/MX chipsets (SMI interface) (NEW)
- Intel Pentium 4 clock modulation (NEW)
- Transmeta LongRun (NEW)
- VIA Cyrix III Longhaul (NEW)



Miglioramenti

- Bus



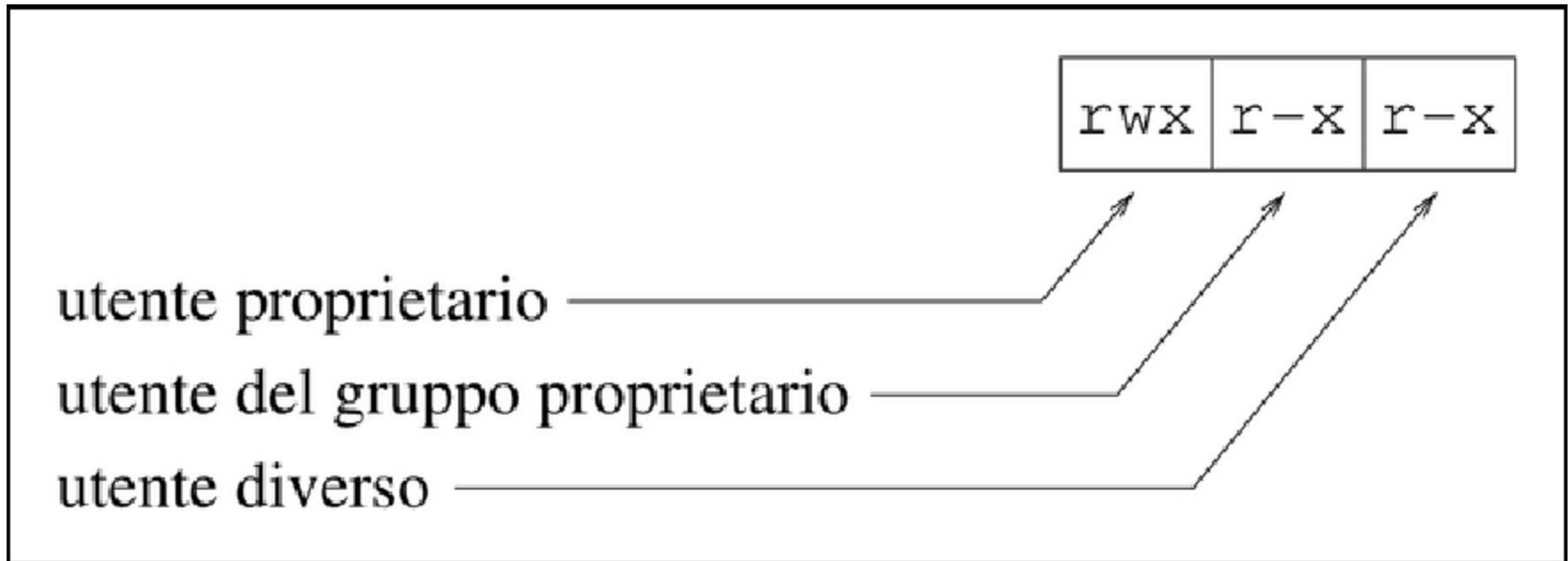
- Bus Interni
 - ◆ ISA, PCI, AGP, Legacy Bus
- Bus Esterni
 - ◆ USB (1 e 2), USB-OTG
- Wireless
 - ◆ AX.25, 802.11, Bluetooth, IrDA



- Access Control List stile Posix
- API di Crittografia

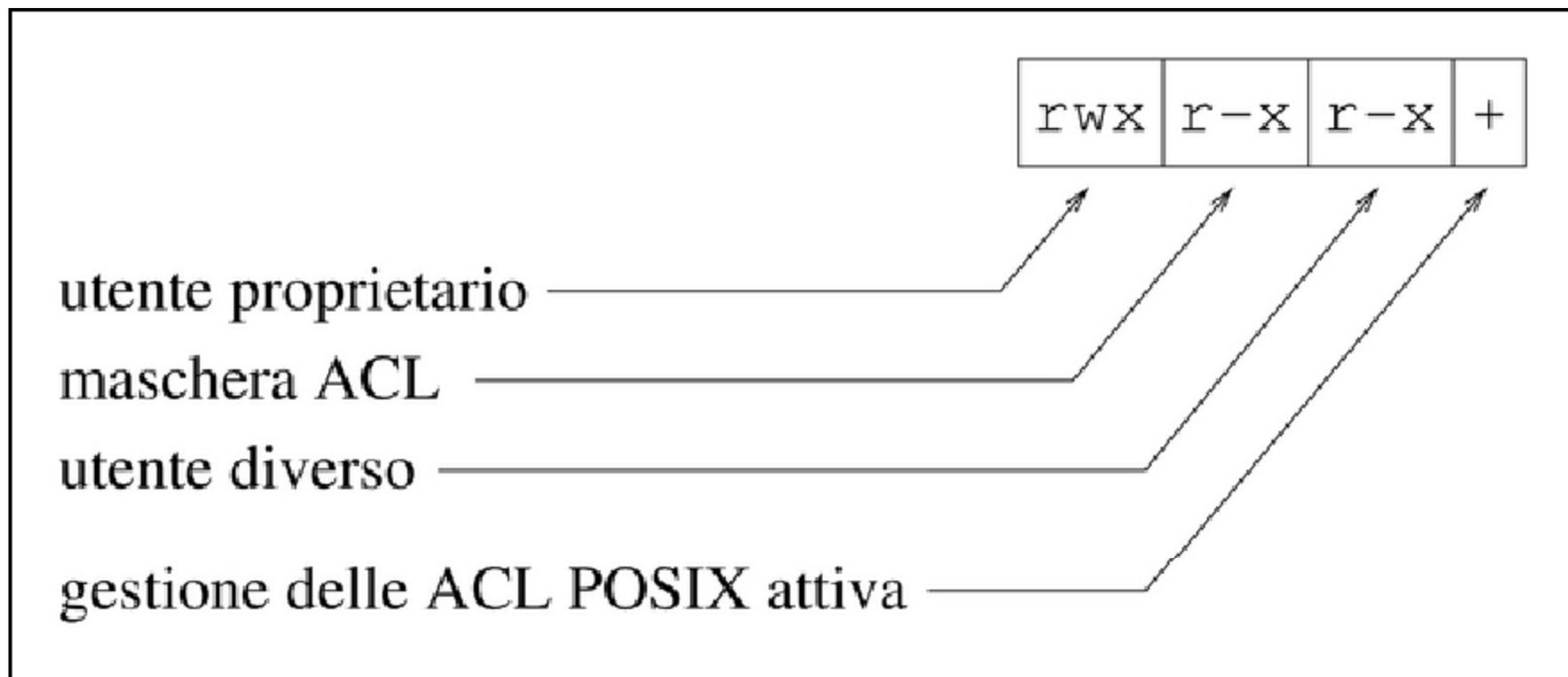


I permessi Unix hanno delle limitazioni





Le ACL permettono di stabilire permessi per determinati utenti e gruppi





- Maschera ACL
 - Definisce i permessi massimi degli utenti o gruppi contenuti nella lista
- Utente diverso
 - Per tutti coloro che non si trovano nella lista
- Gestione ACL attiva
 - [+] se la lista contiene almeno un elemento
 - [-] se la lista è vuota

Sicurezza

- Access Control List stile Posix



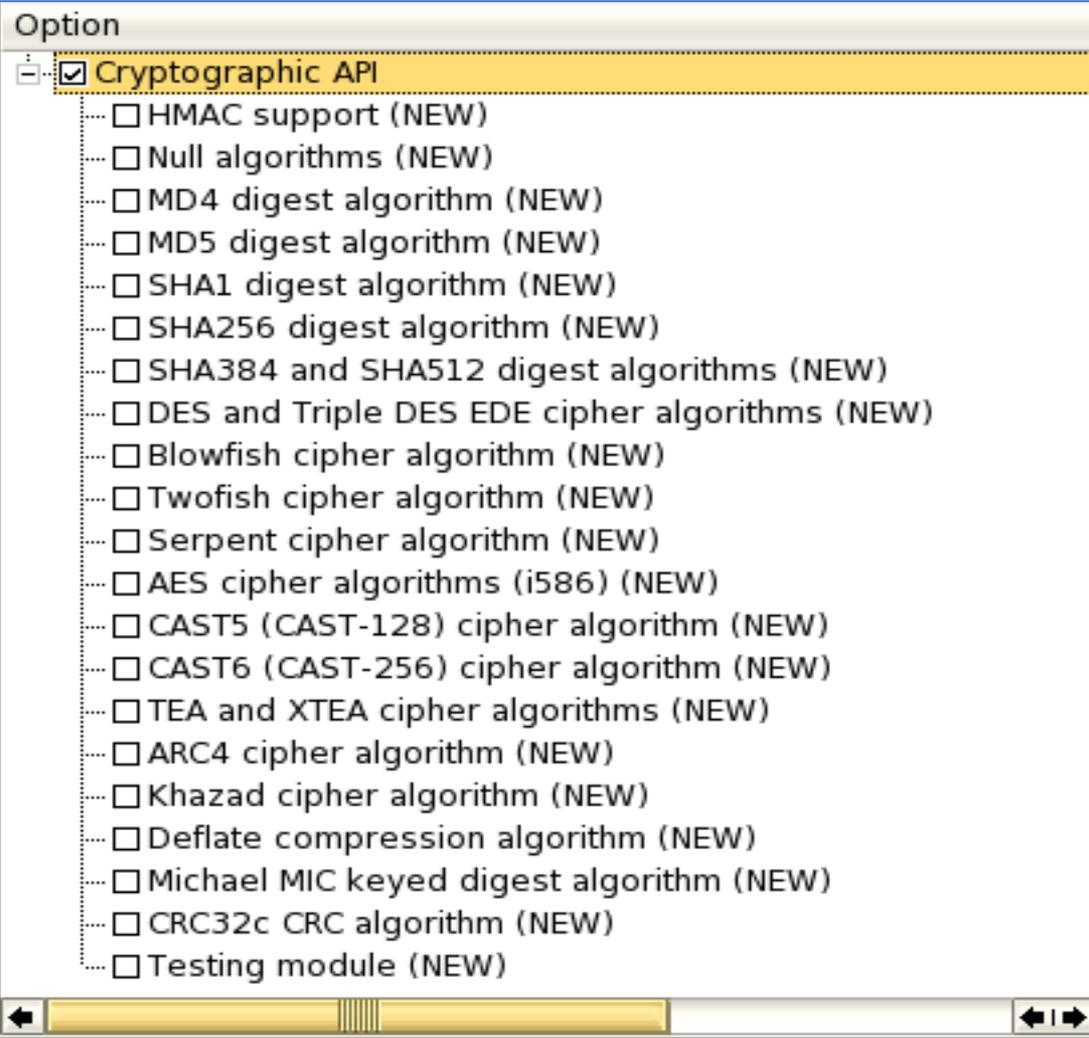
Voce	Descrizione
<code>user::permessi</code>	Dichiarazione dei permessi associati all'utente proprietario (questi permessi non sono filtrati dalla maschera ACL).
<code>group::permessi</code>	Dichiarazione dei permessi associati agli utenti appartenenti al gruppo proprietario.
<code>user:utente:permessi</code>	Dichiarazione dei permessi associati a un utente particolare.
<code>user:gruppo:permessi</code>	Dichiarazione dei permessi associati agli utenti di un gruppo particolare.
<code>mask::permessi</code>	Dichiarazione della maschera dei permessi concessi a tutte le classi di utenti, escluso il proprietario e gli utenti che non ricadono in alcuna categoria specificata.
<code>other::permessi</code>	Dichiarazione dei permessi associati agli utenti che non vengono individuati in alcuna categoria particolare (questi permessi non sono filtrati dalla maschera ACL).

Sicurezza

- API di Crittografia

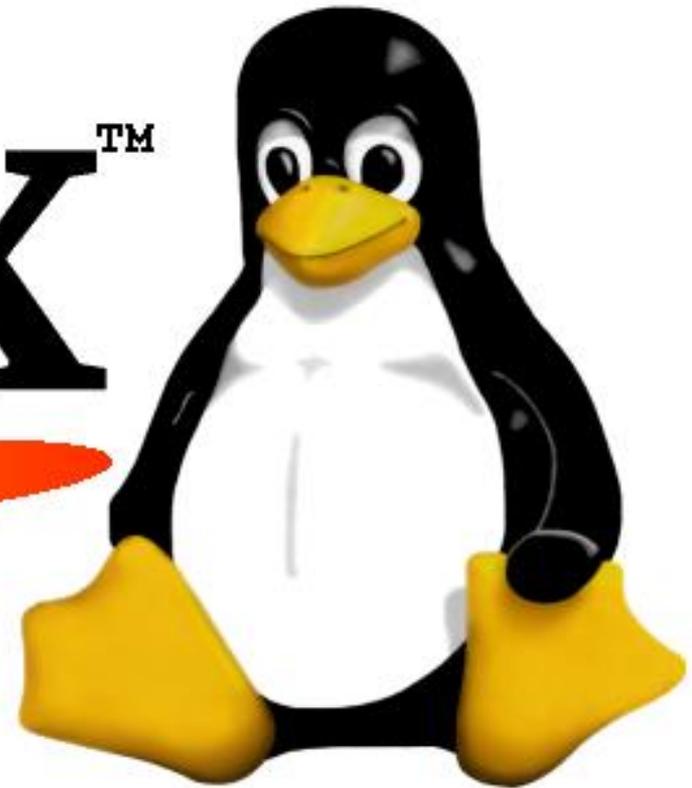


- HMAC
- MD4
- MD5
- SHA
- DES
- AES
- TEA
- ARC4
- Ed altri...



Fine

Linux™





Link Utili



- <http://www.kernel.org>
 - ◆ Dove è possibile scaricare un kernel linux
- <http://www.pluto.linux.it>
 - ◆ Documentazione in italiano per i sistemi linux
- <http://www.linuxdesktop.it>
 - ◆ Problemi incontrati installando linux
- <http://www.linuxiso.org>
 - ◆ Dove scaricare una distribuzione linux
- <http://studenti.unina.it/~fsterrar>
 - ◆ Per scaricare queste slide

Passare al Kernel 2.6



- Nuova installazione
- Compilazione
 - ◆ Scaricare il kernel
 - ◆ Configurazione
 - ◆ Compilazione
 - ◆ Installazione
 - ◆ Reboot

Passare al Kernel 2.6

- Nuova installazione



Possiamo avere un sistema linux 2.6 installando una qualsiasi delle seguenti versioni (distribuzioni) di linux:

- Fedora Core
- Mandrake
- SuSe
- Debian
- Gentoo
- Linspire
- Ubuntu
- ...tante altre!

Questo è il metodo più semplice...

Passare al Kernel 2.6

- Compilazione



Si può passare al kernel 2.6 senza rimuovere la precedente installazione di linux.

Operazioni necessarie:

- Scaricare il kernel
- Configurazione
- Compilazione
- Installazione
- Reboot

Faremo un esempio completo di installazione di un nuovo kernel, ma per sfruttare al massimo le potenzialità di un kernel 2.6 bisogna installare anche programmi come: acpid, setfacl, getfacl, alsamixer, ed altri...

Passare al Kernel 2.6



- Compilazione: Scaricare il kernel

Dal sito www.kernel.org possiamo scaricare:

- La versione completa del nuovo kernel
- La path di aggiornamento dalla versione precedente

Nel nostro esempio useremo la versione completa 2.6.8.1

La prima cosa da fare è decomprimere il kernel appena scaricato...

Passare al Kernel 2.6

- Compilazione: Scaricare il kernel



Dopo aver scaricato il kernel lo decomprimiamo con il comando:

```
tar xjf nome_kernel.tar.bz2
```

In alternativa possiamo utilizzare programmi grafici come "Ark" di KDE oppure "File Roller" di Gnome.

Passare al Kernel 2.6

- Compilazione: Scaricare il kernel



```
xterm
saverio@virtuallinux:~$ ls -l
total 34844
drwxr-xr-x  2 saverio users      4096 2004-10-08 15:34 Desktop/
drwx----- 2 saverio users      4096 2004-12-10 22:09 Mail/
-r--r--r--  1 saverio users 35628066 2005-01-16 22:33 linux-2.6.8.1.tar.bz2
saverio@virtuallinux:~$ tar xjf linux-2.6.8.1.tar.bz2
saverio@virtuallinux:~$ ls -l
total 34848
drwxr-xr-x  2 saverio users      4096 2004-10-08 15:34 Desktop/
drwx----- 2 saverio users      4096 2004-12-10 22:09 Mail/
drwxr-xr-x 18 saverio users      4096 2004-08-14 12:56 linux-2.6.8.1/
-r--r--r--  1 saverio users 35628066 2005-01-16 22:33 linux-2.6.8.1.tar.bz2
saverio@virtuallinux:~$ cd linux-2.6.8.1
saverio@virtuallinux:~/linux-2.6.8.1$ ls
COPYING          Makefile         crypto/          init/            mm/              sound/
CREDITS           README           drivers/         ipc/             net/             usr/
Documentation/   REPORTING-BUGS  fs/             kernel/          scripts/
MAINTAINERS      arch/           include/         lib/             security/
saverio@virtuallinux:~/linux-2.6.8.1$
```

Passare al Kernel 2.6

- Compilazione: Configurazione



Dopo aver decompresso il kernel dobbiamo configurarlo con il comando:

make xconfig

In alternativa possiamo usare

make menuconfig

Oppure

make config

Passare al Kernel 2.6

- Compilazione: Configurazione



```
xterm
saverio@virtuallinux:~/linux-2.6.8.1$ make xconfig ←
HOSTCC  scripts/basic/fixdep
HOSTCC  scripts/basic/split-include
HOSTCC  scripts/basic/docproc
SHIPPED scripts/kconfig/zconf.tab.h
HOSTCC  scripts/kconfig/conf.o
sed < scripts/kconfig/lkc_proto.h > scripts/kconfig/lkc_defs.h 's/P(\([^,]*\))
,./#define \1 (\*\1_p)/'
HOSTCC  scripts/kconfig/kconfig_load.o
HOSTCC  scripts/kconfig/mconf.o
/usr/lib/qt/bin/moc -i scripts/kconfig/qconf.h -o scripts/kconfig/qconf.moc
HOSTCXX scripts/kconfig/qconf.o
SHIPPED scripts/kconfig/zconf.tab.c
SHIPPED scripts/kconfig/lex.zconf.c
HOSTCC  -fPIC scripts/kconfig/zconf.tab.o
HOSTLLD -shared scripts/kconfig/libkconfig.so
HOSTLD  scripts/kconfig/qconf
scripts/kconfig/qconf arch/i386/Kconfig
#
# using defaults found in arch/i386/defconfig
#
arch/i386/defconfig:175: trying to assign nonexistent symbol PCI_USE_VECTOR
arch/i386/defconfig:251: trying to assign nonexistent symbol BLK_DEV_CARMEL
arch/i386/defconfig:569: trying to assign nonexistent symbol NET_FASTROUTE
arch/i386/defconfig:1247: trying to assign nonexistent symbol X86_STD_RESOURCE
ES
□
```



Option

- Code maturity level options
- General setup
 - Configure standard kernel features (for small systems)
- Loadable module support
- Processor type and features
 - Firmware Drivers
- Power management options (ACPI, APM)
 - ACPI (Advanced Configuration and Power Interface) Support
 - APM (Advanced Power Management) BIOS Support
 - CPU Frequency scaling
- Bus options (PCI, PCMCIA, EISA, MCA, ISA)
 - PCMCIA/CardBus support
 - PCI Hotplug Support
- Executable file formats
- Device Drivers
 - Generic Driver Options
 - Memory Technology Devices (MTD)
 - Parallel port support
 - Plug and Play support
 - Block devices
 - ATA/ATAPI/MFM/RLL support
 - SCSI device support
 - Old CD-ROM drivers (not SCSI, not IDE)
 - Multi-device support (RAID and LVM)
 - Fusion MPT device support
 - IEEE 1394 (FireWire) support
 - I2O device support
 - Networking support
 - Amateur Radio support
 - IrDA (infrared) subsystem support
 - Bluetooth subsystem support
 - ISDN subsystem
 - Telephony Support
 - Input device support
 - Character devices
 - Serial drivers
 - IPMI

Option

- Prompt for development
- Select only drivers

Code maturity level options

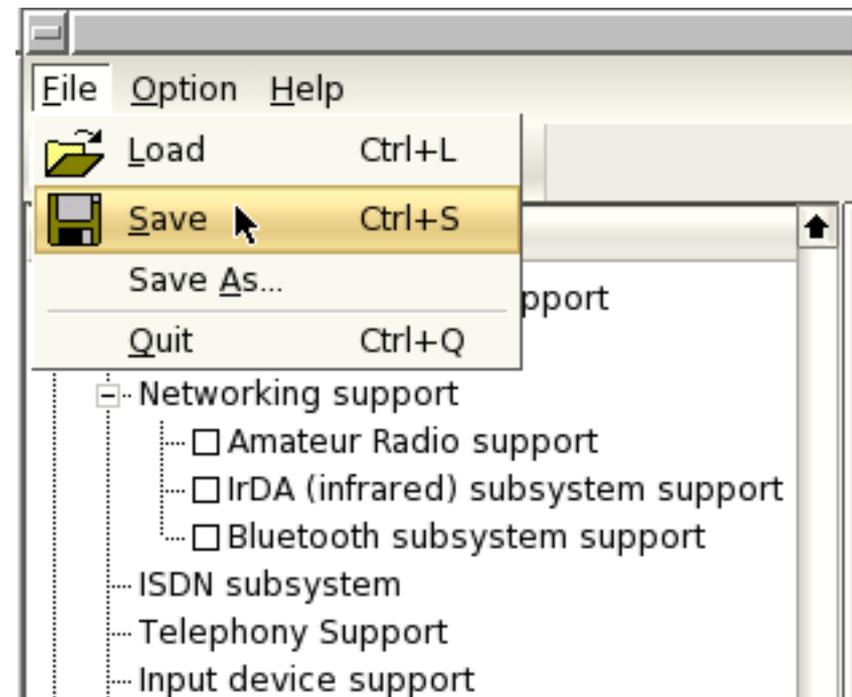
Passare al Kernel 2.6

- Compilazione: Configurazione



Dopo aver configurato il kernel a nostro piacimento, dobbiamo salvare la configurazione.

Il prossimo passo è la compilazione!



Passare al Kernel 2.6

- Compilazione: Compilazione



Il comando per compilare il kernel è:
make bzImage

Per compilare i moduli:
make modules

Alla fine installeremo i moduli con:
make modules_install

```
saverio@virtuallinux:~/linux-2.6.8.1$ make bzImage
```



```
CHK      include/linux/version.h
UPD      include/linux/version.h
SYMLINK  include/asm -> include/asm-i386
SPLIT    include/linux/autoconf.h -> include/config/*
CC       scripts/mod/empty.o
HOSTCC   scripts/mod/mk_elfconfig
MKELF    scripts/mod/elfconfig.h
HOSTCC   scripts/mod/file2alias.o
HOSTCC   scripts/mod/modpost.o
HOSTCC   scripts/mod/sumversion.o
HOSTLD   scripts/mod/modpost
HOSTCC   scripts/conmakehash
HOSTCC   scripts/kallsyms
HOSTCC   scripts/pnmtologo
HOSTCC   scripts/bin2c
CC       arch/i386/kernel/asm-offsets.s
CHK      include/asm-i386/asm_offsets.h
UPD      include/asm-i386/asm_offsets.h
CC       init/main.o
CHK      include/linux/compile.h
```

```
saverio@virtuallinux:~/linux-2.6.8.1$ make modules ←  
make[1]: `arch/i386/kernel/asm-offsets.s' is up to date.  
CC [M] fs/ntfs/aops.o  
CC [M] fs/ntfs/attrib.o  
CC [M] fs/ntfs/collate.o  
CC [M] fs/ntfs/compress.o  
CC [M] fs/ntfs/debug.o  
CC [M] fs/ntfs/dir.o  
CC [M] fs/ntfs/file.o  
CC [M] fs/ntfs/index.o  
CC [M] fs/ntfs/inode.o
```

Passare al Kernel 2.6

- Compilazione: Compilazione



```
xterm
saverio@virtuallinux:~/linux-2.6.8.1$ su ←
Password:
root@virtuallinux:/home/saverio/linux-2.6.8.1# make modules_install ←
□
```

Uso il comando "**su**" per cambiare il dominio di protezione da quello dell'utente "saverio" a quello del superutente "root".

Passare al Kernel 2.6

- Compilazione: Compilazione



Finora abbiamo compilato e installato i moduli del kernel.

L'immagine del kernel (bzImage) è stata compilata, ma non ancora abbiamo settato il computer per partire con il nuovo kernel.

Passare al Kernel 2.6



- Compilazione: Installazione

L'immagine del kernel (cioè l'eseguibile) è situato nel nostro caso nella directory:

`./arch/i386/boot/bzImage`

Se avevamo una diversa architettura, bzImage veniva creato in un'altra directory.

Passare al Kernel 2.6

- Compilazione: Installazione



```
xterm
saverio@virtuallinux:~/linux-2.6.8.1$ pwd
/home/saverio/linux-2.6.8.1
saverio@virtuallinux:~/linux-2.6.8.1$ cd arch/i386/boot/
saverio@virtuallinux:~/linux-2.6.8.1/arch/i386/boot$ ls
Makefile      bzImage*      edd.S          mtools.conf.in  tools/
bootsect.S    compressed/   install.sh     setup.S          video.S
saverio@virtuallinux:~/linux-2.6.8.1/arch/i386/boot$ su
Password:
root@virtuallinux:/home/saverio/linux-2.6.8.1/arch/i386/boot# mv bzImage /boot
root@virtuallinux:/home/saverio/linux-2.6.8.1/arch/i386/boot# cd /boot
root@virtuallinux:/boot# ls
README.initrd      boot.0302      config-ide-2.4.26  vmlinuz
System.map         boot_message.txt  diag1.img          vmlinuz-ide-2.4.26
System.map-ide-2.4.26  bzImage        lost+found
boot.0300          config          map
root@virtuallinux:/boot#
```



Passare al Kernel 2.6

- Compilazione: Installazione



Dobbiamo settare il bootloader in modo che al momento del bootstrap ci permetta di far avviare il computer con il nuovo kernel.

Faremo l'esempio con il bootloader LiLo (Linux Loader) presente nella maggior parte delle distribuzioni linux.

Passare al Kernel 2.6

- Compilazione: Installazione



```
xterm
root@virtuallinux:/boot# emacs /etc/lilo.conf
root@virtuallinux:/boot# lilo
Added Linux *
Added Mio_Kernel
root@virtuallinux:/boot#
```

```
emacs@virtuallinux.emac
File Edit Options Buffers Tools Help
[Icons]
# End LILO global section
# Linux bootable partition config begins

image = /boot/vmlinuz
root = /dev/hda2
label = Linux
read-only

image = /boot/bzImage
root = /dev/hda2
label = Mio_Kernel
read-only

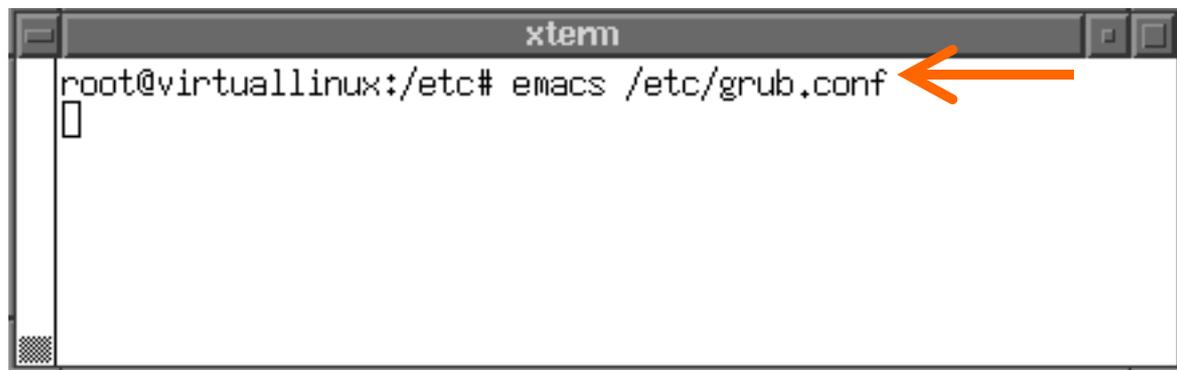
# Linux bootable partition config ends
[ ]
--** lilo.conf (Fundamental)--L48--Bot-----
[ ] Auto-saving... done
```

Passare al Kernel 2.6



- Compilazione: Installazione

Se al posto di lilo vogliamo usare grub, basterà modificare opportunamente il file **/etc/grub.conf**

A terminal window titled 'xterm' showing a root prompt at 'root@virtuallinux:~/etc#'. The command 'emacs /etc/grub.conf' has been entered. An orange arrow points to the end of the command line.

```
xterm
root@virtuallinux:~/etc# emacs /etc/grub.conf
```

Passare al Kernel 2.6

- Compilazione: Reboot



Non ci resta che riavviare il sistema.

Il bootloader ci chiederà se vogliamo avviare linux con il kernel vecchio o con il nuovo.

In caso di cattivo settaggio del kernel il computer potrebbe non avviarsi: non disperarsi!!! Possiamo utilizzare sempre il vecchio kernel.