(Ri)compilare il kernel

Paola Villa

Università degli studi dell'Insubria

cos'è il kernel

- il kernel è la parte più importante del sistema operativo;
  - si occupa della gestione delle risorse essenizali (la CPU, la memoria, le periferiche) presenti nel nostro computer
  - le rende direttamente accessibili agli altri programmi
- è un mediatore che serve a far interagire il software con l'hardware

cos'è il kernel

- il kernel è la parte più importante del sistema operativo;
  - si occupa della gestione delle risorse essenizali (la CPU, la memoria, le periferiche) presenti nel nostro computer
  - le rende direttamente accessibili agli altri programmi
- è un mediatore che serve a far interagire il software con l'hardware

cos'è il kernel

- il kernel è la parte più importante del sistema operativo;
  - si occupa della gestione delle risorse essenizali (la CPU, la memoria, le periferiche) presenti nel nostro computer
  - le rende direttamente accessibili agli altri programmi
- è un mediatore che serve a far interagire il software con l'hardware

perchè ricompilare il kernel?

- per elimiminare codice inutile
- per ottimizzare le prestazioni
- per aggiornare il codice
- per motivi di “sicurezza”

perchè ricompilare il kernel?

- per elimiminare codice inutile
- per ottimizzare le prestazioni
- per aggiornare il codice
- per motivi di “sicurezza”

perchè ricompilare il kernel?

- per elimiminare codice inutile
- per ottimizzare le prestazioni
- per aggiornare il codice
- per motivi di "sicurezza"

perchè ricompilare il kernel?

- per elimiminare codice inutile
- per ottimizzare le prestazioni
- per aggiornare il codice
- per motivi di “sicurezza”

che cosa bisogna sapere

per compilare bene il kernel è essenziale conoscere l'hardware della macchina che usiamo

- lshw
- comando lspci
- http://kmuto.jp/debian/hcl/

che cosa bisogna sapere

per compilare bene il kernel è essenziale conoscere l'hardware della macchina che usiamo

- lshw
- comando lspci
- http://kmuto.jp/debian/hcl/

che cosa bisogna sapere

per compilare bene il kernel è essenziale conoscere l'hardware della macchina che usiamo

- lshw
- comando lspci
- http://kmuto.jp/debian/hcl/

che cosa bisogna sapere

per compilare bene il kernel è essenziale conoscere l'hardware della macchina che usiamo

- lshw
- comando lspci
- http://kmuto.jp/debian/hcl/

http://kmuto.jp/debian/hcl/

Debian GNU/Linux device driver check page

Please paste your result of 'lspci -n' taken from GNU/Linux OS (such as Debian, Knoppix, RedHat, and so on) to below box. Then let's push 'Check' button.

```
00:01.0 0604: 1106:b198
00:0d.0 0703: 163c:3052 (rev 04)
00:0f.0 0101: 1106:3149 (rev 80)
00:0f.1 0101: 1106:0571 (rev 06)
00:10.0 0c03: 1106:3038 (rev 81)
00:10.1 0c03: 1106:3038 (rev 81)
00:10.2 0c03: 1106:3038 (rev 81)
00:10.3 0c03: 1106:3038 (rev 81)
00:10.4 0c03: 1106:3104 (rev 86)
00:11.0 0601: 1106:3227
00:11.5 0401: 1106:3059 (rev 60)
00:13.0 0200: 10ec:8139 (rev 10)
00:14.0 0c00: 1106:3044 (rev 80)
01:00.0 0300: 1002:5964 (rev 01)
01:00.1 0380: 1002:5d44 (rev 01)
```

Check Clear

[View registered hardware database]

Notice:

- **This database uses the PCI map of Debian kernel 2.6.22-1-686 .**
- **The result does NOT mean your hardware works perfectly. (vice versa)**
- **This database only verifies the PCI devices at this time. X drivers, ISA, USB, IEEE1394 or any other devices are out of its focus.**

http://kmuto.jp/debian/hcl/

Debian GNU/Linux device driver check page

PCI ID	Works?	Vendor	Device	Driver	Comment
11063189	Yes	VIA Technologies, Inc.	VT8377 [KT400/KT600 AGP] Host Bridge	via-agp	v2.6.21
1106b198	Yes	VIA Technologies, Inc.	VT8237 PCI Bridge	via-agp	v2.6.21
163c3052	-	Smart Link Ltd.	SmartLink SmartPCI562 56K Modem		
11063149	Yes	VIA Technologies, Inc.	VIA VT6420 SATA RAID Controller	sata_via	v2.6.21
11060571	Yes	VIA Technologies, Inc.	VT82C586/B/VT82C686/A/B/VT8233/A/C/VT8235 PIPC Bus Master IDE	via82cxxx	v2.6.21
11063038	Yes	VIA Technologies, Inc.	VT82xxxxx UHCI USB 1.1 Controller	usb-uhci,uhci-hcd	
11063038	Yes	VIA Technologies, Inc.	VT82xxxxx UHCI USB 1.1 Controller	usb-uhci,uhci-hcd	
11063038	Yes	VIA Technologies, Inc.	VT82xxxxx UHCI USB 1.1 Controller	usb-uhci,uhci-hcd	
11063038	Yes	VIA Technologies, Inc.	VT82xxxxx UHCI USB 1.1 Controller	usb-uhci,uhci-hcd	
11063104	Yes	VIA Technologies, Inc.	USB 2.0	ehci-hcd	
11063227	Yes	VIA Technologies, Inc.	DFI KT600-AL Motherboard	i2c-viapro	v2.6.21
11063059	Yes	VIA Technologies, Inc.	L7VMM2 Motherboard	snd-via82xx,via82cxxx_audio	v2.6.21
10ec8139	Yes	Realtek Semiconductor Co., Ltd.	RT8139	8139too,8139cp	v2.6.21
11063044	Yes	VIA Technologies, Inc.	IEEE 1394 Host Controller	ohci1394	
10025964	Yes	ATI Technologies Inc	RV280 [Radeon 9200 SE]	radeonfb,ati	v2.6.21
10025d44	Yes	ATI Technologies Inc	RV280 [Radeon 9200 SE] (Secondary)	ati	

You can help us!

Could you give me more information about your machine? Your information (PCI list and text you write below) is used only for making our HCL database more better.

Machine (or motherboard) vendor: -unknown Other:

Product (or motherboard) name:

(For example "Let's Note Y2")

che cosa bisogna sapere

un modo molto semplice per capire che cosa ci serve è dare un'occhiata all'output del comando lsmod durante l'esecuzione di un kernel generico e annotarsi il risultato

lsmod

- lsmod elenca tutti i moduli che sono caricati dal kernel in un dato momento
- un modulo é una porzione del kernel che si occupa di gestire uno solo degli aspetti del sistema operativo, e viene caricato in memoria solo quando necessario, cioé al momento del suo utilizzo
- la gestione dei moduli (cioè il loro caricamento e il loro scaricamento) è affidata al kernel stesso

se un dato modulo viene caricato ad ogni avvio dal kernel generico, vuol dire che serve sempre...

lsmod

- lsmod elenca tutti i moduli che sono caricati dal kernel in un dato momento
- un modulo é una porzione del kernel che si occupa di gestire uno solo degli aspetti del sistema operativo, e viene caricato in memoria solo quando necessario, cioé al momento del suo utilizzo
- la gestione dei moduli (cioè il loro caricamento e il loro scaricamento) è affidata al kernel stesso

se un dato modulo viene caricato ad ogni avvio dal kernel generico, vuol dire che serve sempre...

lsmod

- lsmod elenca tutti i moduli che sono caricati dal kernel in un dato momento
- un modulo é una porzione del kernel che si occupa di gestire uno solo degli aspetti del sistema operativo, e viene caricato in memoria solo quando necessario, cioé al momento del suo utilizzo
- la gestione dei moduli (cioè il loro caricamento e il loro scaricamento) è affidata al kernel stesso

se un dato modulo viene caricato ad ogni avvio dal kernel generico, vuol dire che serve sempre...

lsmod

- lsmod elenca tutti i moduli che sono caricati dal kernel in un dato momento
- un modulo é una porzione del kernel che si occupa di gestire uno solo degli aspetti del sistema operativo, e viene caricato in memoria solo quando necessario, cioé al momento del suo utilizzo
- la gestione dei moduli (cioè il loro caricamento e il loro scaricamento) è affidata al kernel stesso

se un dato modulo viene caricato ad ogni avvio dal kernel generico, vuol dire che serve sempre...

cosa vi serve

per compilare vi servono i tool per la compilazione, cioè i pacchetti necessari per compilare i sorgenti C (compilatore e librerie) ed il programma make... cose che si trovano in qualunque distro linux;

dove reperire nuove versioni del kernel

- visto che è la prima volta che compilate un kernel vi consiglio caldamente di effettuare i vostri esperimenti su un kernel diverso da quello con cui lavorate di solito
  - per motivi di sicurezza
  - perchè conviene partire dalla configurazione del kernel generico
- per reperire una nuova versione del kernel linux basta collegarsi al sito www.kernel.org e scaricare la versione full
- per compilare il kernel non serve nè posizionare subito il contenuto dell'archivio in /usr/src e nemmeno diventare root!

dove reperire nuove versioni del kernel

- visto che è la prima volta che compilate un kernel vi consiglio caldamente di effettuare i vostri esperimenti su un kernel diverso da quello con cui lavorate di solito
  - per motivi di sicurezza
  - perchè conviene partire dalla configurazione del kernel generico
- per reperire una nuova versione del kernel linux basta collegarsi al sito www.kernel.org e scaricare la versione full
- per compilare il kernel non serve nè posizionare subito il contenuto dell'archivio in /usr/src e nemmeno diventare root!

dove reperire nuove versioni del kernel

- visto che è la prima volta che compilate un kernel vi consiglio caldamente di effettuare i vostri esperimenti su un kernel diverso da quello con cui lavorate di solito
    - per motivi di sicurezza
    - perchè conviene partire dalla configurazione del kernel generico
- per reperire una nuova versione del kernel linux basta collegarsi al sito www.kernel.org e scaricare la versione full
- per compilare il kernel non serve nè posizionare subito il contenuto dell'archivio in /usr/src e nemmeno diventare root!

dove reperire nuove versioni del kernel

- visto che è la prima volta che compilate un kernel vi consiglio caldamente di effettuare i vostri esperimenti su un kernel diverso da quello con cui lavorate di solito
    - per motivi di sicurezza
    - perchè conviene partire dalla configurazione del kernel generico
- per reperire una nuova versione del kernel linux basta collegarsi al sito www.kernel.org e scaricare la versione full
- per compilare il kernel non serve nè posizionare subito il contenuto dell'archivio in /usr/src e nemmeno diventare root!

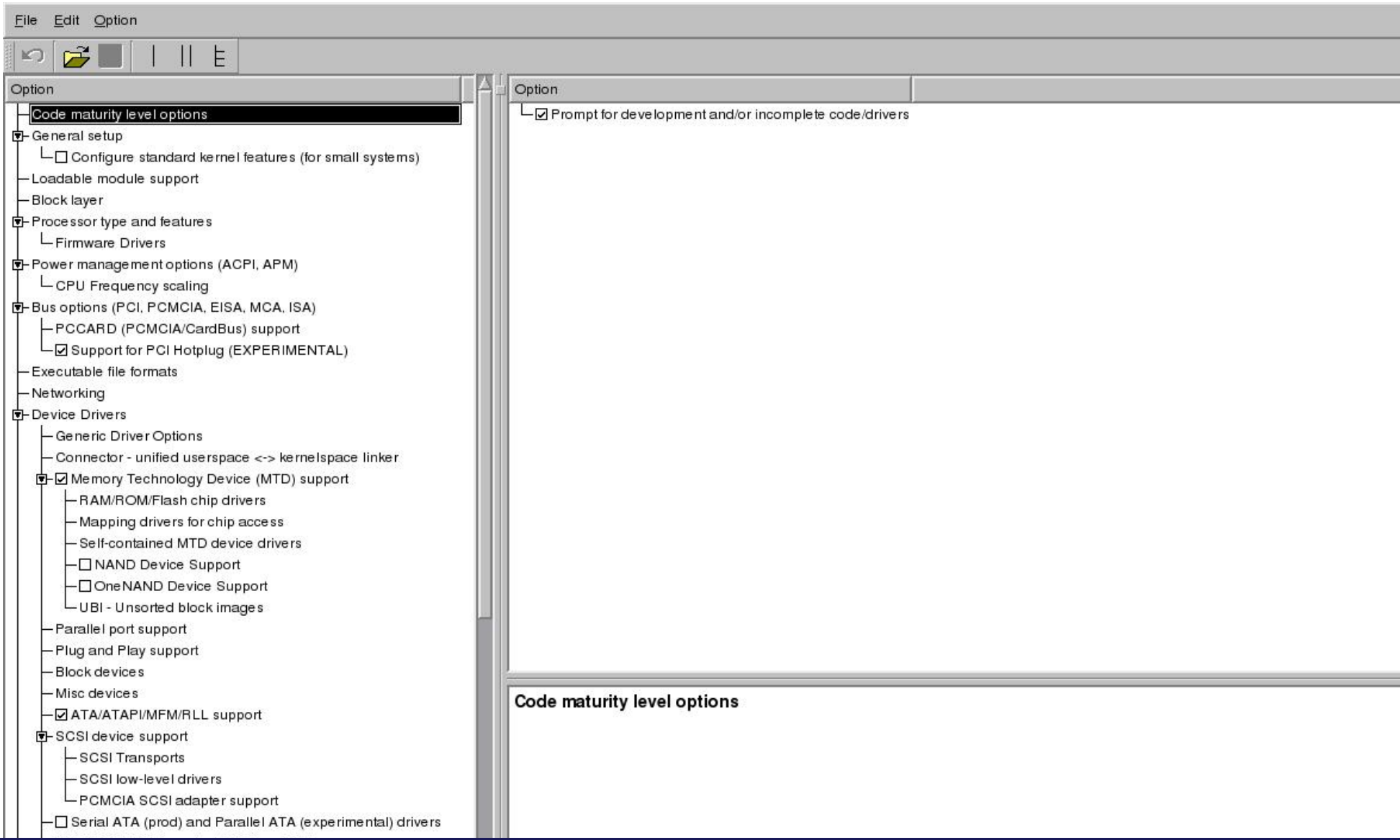
fase di configurazione

- dopo avere scompattato l'archivio con il comando `tar xvjf linux-[versione].tar.bz2` si entra nelle directory appena scompattata e si passa alla fase di configurazione vera e propria
- a questo punto è possibile scegliere tra:
  - make xconfig
  - make menuconfig
  - make config

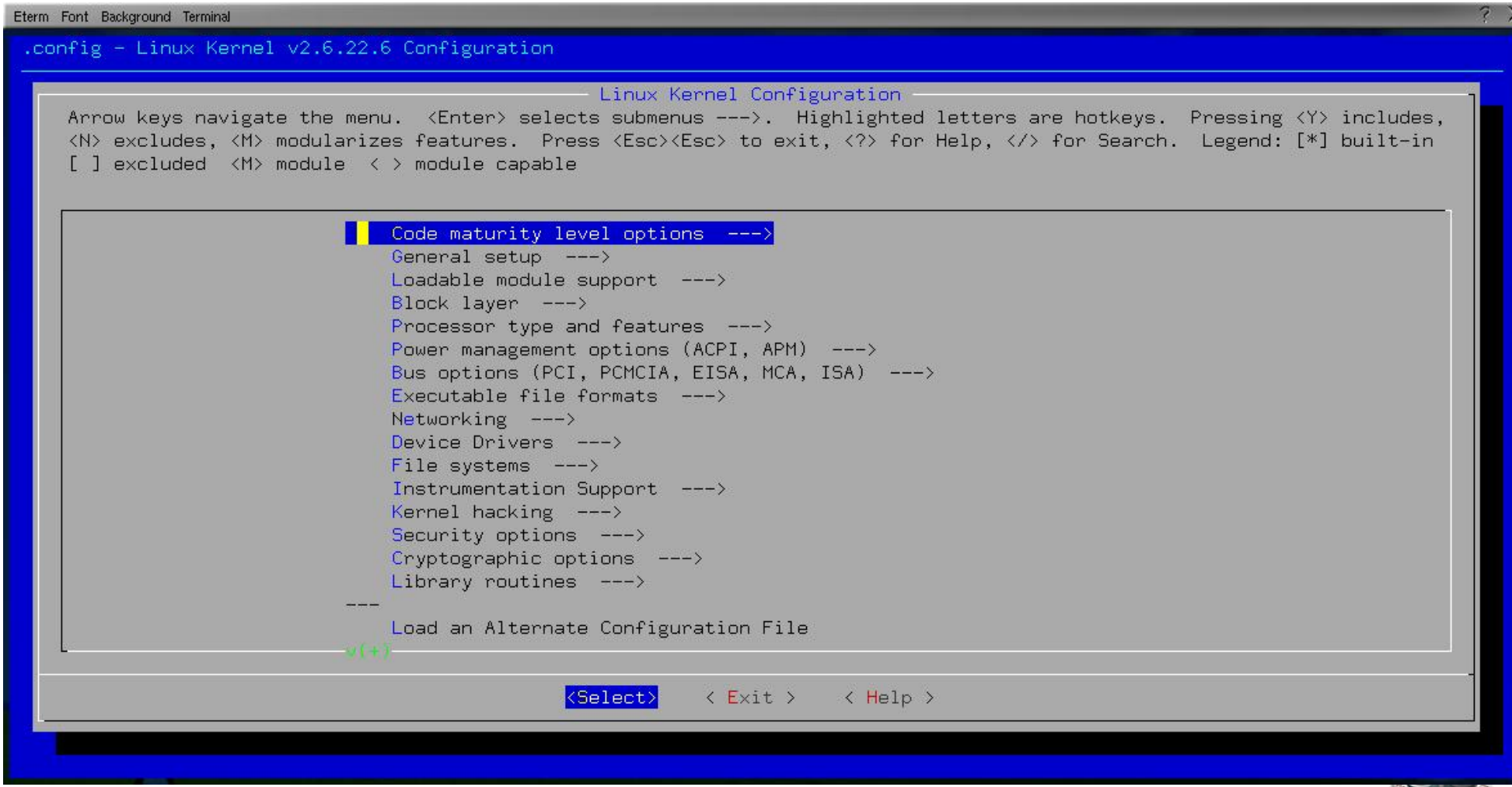
fase di configurazione

- dopo avere scompattato l'archivio con il comando `tar xvjf linux-[versione].tar.bz2` si entra nelle directory appena scompattata e si passa alla fase di configurazione vera e propria
- a questo punto è possibile scegliere tra:
  - make xconfig
  - make menuconfig
  - make config

make xconfig



make menuconfig



make config

```
scripts/kconfig/conf arch/i386/Kconfig
*
* Linux Kernel Configuration
*
*
* Code maturity level options
*
Prompt for development and/or incomplete code/drivers (EXPERIMENTAL) [Y/n/?]
```

fase di configurazione

- la cosa migliore da fare è non usare make config e caricare in memoria il file di configurazione del kernel attualmente in uso
- basta scegliere la voce Load an Alternate Configuration File e scrivere il percorso esatto, ad esempio:/boot/config
- fatto questo viene il momento per scremare tutto ciò che non serve

fase di configurazione

- la cosa migliore da fare è non usare make config e caricare in memoria il file di configurazione del kernel attualmente in uso
- basta scegliere la voce Load an Alternate Configuration File e scrivere il percorso esatto, ad esempio:/boot/config
- fatto questo viene il momento per scremare tutto ciò che non serve

fase di configurazione

- la cosa migliore da fare è non usare make config e caricare in memoria il file di configurazione del kernel attualmente in uso
- basta scegliere la voce Load an Alternate Configuration File e scrivere il percorso esatto, ad esempio:/boot/config
- fatto questo viene il momento per scremare tutto ciò che non serve

fase di configurazione

- quel che bisogna fare è scandire ogni sezione andando ad abilitare solo ciò che è necessario ed escludendo tutto il resto
- per scegliere cosa inserire basta dare un'occhiata all'output del sito che vi ho detto prima, ai moduli utilizzati dal kernel generico,all'output di lspci e di lshw

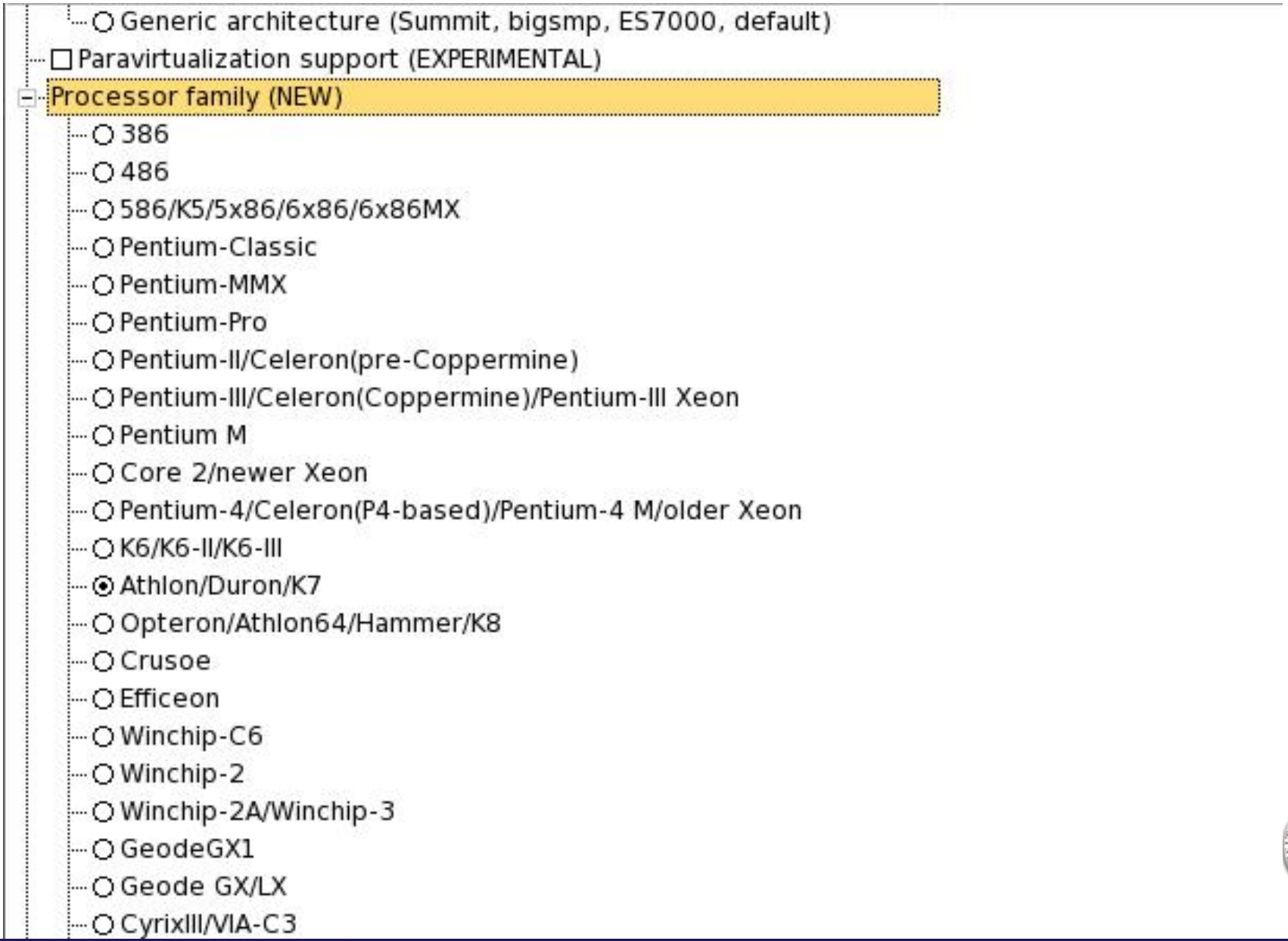
fase di configurazione

- quel che bisogna fare è scandire ogni sezione andando ad abilitare solo ciò che è necessario ed escludendo tutto il resto
- per scegliere cosa inserire basta dare un'occhiata all'output del sito che vi ho detto prima, ai moduli utilizzati dal kernel generico,all'output di lspci e di lshw

esempio

- dall'output di lshw vengo a sapere che il mio processore è un AMD Athlon(tm) XP 3000+
- a questo punto nella sezione "Processor type and features" sotto "Processor family" so che cosa abilitare

make xconfig



fase di configurazione

ci sono due modi per abilitare alcune componenti del kernel

- built-in:se io abilito una determinata componente built-in essa andrà a fare parte del file che costituisce il kernel vero e proprio;
- come modulo:se io abilito una componente come modulo il suo codice verrà tenuto staccato dal kernel e verrà messo in una directory particolare:/lib/modules;

fase di configurazione

- conviene abilitare come built-in ogni cosa che servirà ad ogni avvio: ad esempio, il filesystem che utilizziamo, il supporto per la scheda video e quello per il chipset della scheda madre...
- tutto ciò che invece non utilizziamo spesso va abilitato come modulo: nel mio caso, il modulo che consente di comunicare con la stampante, il modulo usbstorage, il supporto per il filesystem iso9660...
- la configurazione del kernel è molto soggettiva, perchè dipende dall'utilizzo che ciascuno fa del proprio computer

fase di configurazione

- conviene abilitare come built-in ogni cosa che servirà ad ogni avvio: ad esempio, il filesystem che utilizziamo, il supporto per la scheda video e quello per il chipset della scheda madre...
- tutto ciò che invece non utilizziamo spesso va abilitato come modulo: nel mio caso, il modulo che consente di comunicare con la stampante, il modulo usbstorage, il supporto per il filesystem iso9660...
- la configurazione del kernel è molto soggettiva, perchè dipende dall'utilizzo che ciascuno fa del proprio computer

fase di configurazione

- conviene abilitare come built-in ogni cosa che servirà ad ogni avvio: ad esempio, il filesystem che utilizziamo, il supporto per la scheda video e quello per il chipset della scheda madre...
- tutto ciò che invece non utilizziamo spesso va abilitato come modulo: nel mio caso, il modulo che consente di comunicare con la stampante, il modulo usbstorage, il supporto per il filesystem iso9660...
- la configurazione del kernel è molto soggettiva, perchè dipende dall'utilizzo che ciascuno fa del proprio computer

fase di configurazione

- dopo aver configurato il kernel (e aver salvato la configurazione) ho nella mia working directoy un nuovo file .config che contiene tutte le informazioni che ho inserito
- posso riutilizzare questo file come base per compilare le versioni successive del kernel che voglio installare (grazie a make oldconfig)
- la fase "noiosa" in cui devo specificare tutto va fatta una volta sola; la ricompilazione diventa una faccenda molto più sbrigativa

fase di configurazione

- dopo aver configurato il kernel (e aver salvato la configurazione) ho nella mia working directoy un nuovo file .config che contiene tutte le informazioni che ho inserito
- posso riutilizzare questo file come base per compilare le versioni successive del kernel che voglio installare (grazie a make oldconfig)
- la fase "noiosa" in cui devo specificare tutto va fatta una volta sola; la ricompilazione diventa una faccenda molto più sbrigativa

fase di configurazione

- dopo aver configurato il kernel (e aver salvato la configurazione) ho nella mia working directoy un nuovo file .config che contiene tutte le informazioni che ho inserito
- posso riutilizzare questo file come base per compilare le versioni successive del kernel che voglio installare (grazie a make oldconfig)
- la fase "noiosa" in cui devo specificare tutto va fatta una volta sola; la ricompilazione diventa una faccenda molto più sbrigativa

compilazione

per compilare i kernel della serie 2.6 basta usare semplicemente il comando make; per velocizzare il processo di compilazione si può usare l'opzione -j di make, che consente di indicare il numero di comandi che possono essere eseguiti simultaneamente.

installazione

se la compilazione è terminata senza problemi, non resta che installare il nuovo kernel; per questo diventate root e date questi comandi:

- *make modules_install*
- *cp .config /boot/config – [versione]*
- *cp System.map /boot/System.map – [versione]*
- *cp arch/[vostra – architettura]/boot/bzImage /boot/vmlinuz – [versione]*

avvio di un computer

L'avvio di un sistema Linux su un sistema x386 prevede le fasi seguenti:

- all'accensione il bios su rom non volatile definisce l'ordine dei device da utilizzare per effettuare il boot
- il boot sector del primo device di boot contiene il codice (o i riferimenti su dove trovarlo) del boot loader che esegue il bootstrap del sistema operativo
- il boot loader fa partire il caricamento del kernel di Linux, che si copia in memoria ed esegue i controlli e il riconoscimento dell'hardware presente e poi avvia il processo init; esso si occupa del caricamento di tutti gli altri processi

Se non modifico il bootloader non riuscirò mai ad avviare il mio nuovo kernel...

avvio di un computer

L'avvio di un sistema Linux su un sistema x386 prevede le fasi seguenti:

- all'accensione il bios su rom non volatile definisce l'ordine dei device da utilizzare per effettuare il boot
- il boot sector del primo device di boot contiene il codice (o i riferimenti su dove trovarlo) del boot loader che esegue il bootstrap del sistema operativo
- il boot loader fa partire il caricamento del kernel di Linux, che si copia in memoria ed esegue i controlli e il riconoscimento dell'hardware presente e poi avvia il processo init; esso si occupa del caricamento di tutti gli altri processi

Se non modifico il bootloader non riuscirò mai ad avviare il mio nuovo kernel...

avvio di un computer

L'avvio di un sistema Linux su un sistema x386 prevede le fasi seguenti:

- all'accensione il bios su rom non volatile definisce l'ordine dei device da utilizzare per effettuare il boot
- il boot sector del primo device di boot contiene il codice (o i riferimenti su dove trovarlo) del boot loader che esegue il bootstrap del sistema operativo
- il boot loader fa partire il caricamento del kernel di Linux, che si copia in memoria ed esegue i controlli e il riconoscimento dell'hardware presente e poi avvia il processo init; esso si occupa del caricamento di tutti gli altri processi

Se non modifico il bootloader non riuscirò mai ad avviare il mio nuovo kernel...

avvio di un computer

L'avvio di un sistema Linux su un sistema x386 prevede le fasi seguenti:

- all'accensione il bios su rom non volatile definisce l'ordine dei device da utilizzare per effettuare il boot
- il boot sector del primo device di boot contiene il codice (o i riferimenti su dove trovarlo) del boot loader che esegue il bootstrap del sistema operativo
- il boot loader fa partire il caricamento del kernel di Linux, che si copia in memoria ed esegue i controlli e il riconoscimento dell'hardware presente e poi avvia il processo init; esso si occupa del caricamento di tutti gli altri processi

Se non modifico il bootloader non riuscirò mai ad avviare il mio nuovo kernel...

Grub

se usate GRUB basta aggiungere la seguente riga nel file menu.lst (generalmente /boot/grub/menu.lst):
root [*partizione*]
title linux – [*versione*]
kernel /*boot*/*vmlinuz* – [*versione*]*ro*

Lilo

se usate lilo aggiornate il file /etc/lilo.conf aggiungendo le seguenti righe:
$image = /boot/vmlinuz - [versione]$
$root = /dev/[partizione]$
$label = linux - new$
$read - only$
poi date il comando: $lilo -v$

riavviate e godetevi il nuovo kernel!