



by Philip de Groot

<philip/at/authors.linuxfocus.org>

About the author:

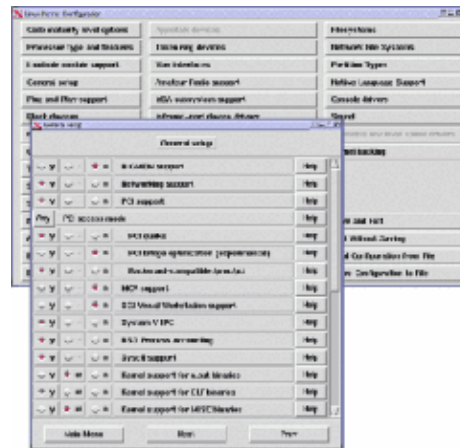
Quest'anno mi aspetto di conseguire il Dottorato di Ricerca all'Università van Nijmegen. La mia tesi riguarda ciò che chiamiamo chemometrics. Attualmente mi occupo di tecnologia informatica applicata alla biologia, presso lo "Academic Medical Center" di Amsterdam, in Olanda. Inoltre, mantengo la mia [pagina \(in olandese\) per i neofiti di Linux](#): È una delle varie iniziative da e per la comunità Linux. Mi piace lavorare con Linux e frequentemente riporto le mie esperienze.

Translated to English by:

Nino R. Pereira

<pereira/at/speakeasy.org>

Compile il vostro kernel



Abstract:

Vedrete immediatamente come avere, configurare, compilare e installare il vostro proprio kernel.

Introduzione

Perchè potreste voler compilare e installare un nuovo kernel tutto vostro? Le possibili ragioni sono:

- Il nuovo kernel ha un migliore supporto hardware.
- Il nuovo kernel offre certi vantaggi, come un miglior supporto per macchine multi-processore, o un supporto per USB. Questo avviene sui kernel 2.4.x.
- Il nuovo kernel elimina vecchi bug.
- Il vostro kernel elimina gli elementi superflui ed è così più veloce e più stabile.

Il problema è che compilare ("rolling") il vostro kernel richiede un buon ammontare di risorse del vostro computer. Perciò un nuovo utente di Linux non provi a compilarlo con leggerezza. Questo articolo mostra come compilare il kernel tramite il comando 'make xconfig'. Con questo comando l'utente modifica il kernel tramite una GUI, un'Interfaccia Utente Grafica, e un mouse. Ci sono circa 40 dump dello schermo che spiegano perchè scegliere o meno alcune opzioni in situazioni particolari. Questi 40 dump possono sembrare eccessivi, ma è il miglior modo di spiegare il funzionamento interno del kernel e il come e il perchè di alcune opzioni del kernel. I dump dello schermo sono basati sul kernel 2.4.6. Il più recente è adesso il 2.4.19, ma a parte alcune nuove opzioni nei menu (es. supporto per nuovo hardware) le schermate sono le stesse e anche il procedimento per compilare il kernel. Un avviso: stampate queste pagine prima di cominciare, così avrete sempre accesso a tutte le informazioni necessarie!

L'articolo è strutturato come segue. Prima discute dove puoi trovare il codice sorgente su Internet, e come installarlo, di seguito le schermate di configurazione del kernel. Una volta configurato il kernel deve essere compilato, ma anche un nuovo kernel compilato non è ancora pronto per l'utilizzo. Primo, il nuovo kernel deve essere installato con il boot manager 'lilo', e prima di usare 'lilo' bisogna creare il file '/etc/lilo.conf'. Potete anche copiare il kernel compilato in una partizione da cui lanciare Linux con un programma DOS/Windows chiamato 'loadlin'. Inoltre ci sono una serie di punti specifici che devono essere considerati, come il supporto PCMCIA necessario ai laptop. PCMCIA, piccole schede che spesso gestiscono connessioni di rete e assomigliano ad una grossa carta di credito, sono supportate dal kernel stesso solo dalla versione 2.4.x. I kernel più vecchi possono supportare le PCMCIA attraverso una compilazione e installazione separata. Suse Linux ha un'altro problema, cioè supporta il sonoro tramite i driver ALSA. Questi driver non sono parte del kernel e devono essere compilati ed installati separatamente, perchè i driver originali non funzionano più. A complicare le cose, il passaggio da una versione ad un'altra del kernel, ad esempio dalla 2.2 alla 2.4, può essere accompagnata da problemi con alcune utility del kernel, le cosiddette 'modutils'. Queste contengono il codice necessario a caricare un modulo kernel: La Figura 3 spiega cos'è un modulo. A volte il nuovo kernel non sa cosa fare con le vecchie 'modutils', così bisogna compilare ed installare una versione più recente di modutils. Problemi come questo sono rari ma possono accadere, ed è meglio conoscerli in anticipo.

Ma, seguendo fedelmente le procedure di questo articolo non c'è pressoché nulla che possa andare male. Il nuovo kernel è aggiunto al 'lilo', o copiato sulla partizione di 'loadlin'. Comunque, in caso di emergenza potete ancora ripartire con il kernel originale. Poi, lavorando sul proprio kernel potete provare risolvere i problemi con il nuovo kernel. Anche quando avete delle difficoltà con le 'modutils' del nuovo kernel è ancora possibile ripartire con il vecchio kernel e risolvere il problema ad esempio, compilandole ed installandole separatamente: tutte le nuove versioni delle 'modutils' sono retrocompatibili con i vecchi kernel, così le nuove 'modutils' funzionano bene con i vecchi kernel.

Installare il codice sorgente del kernel

Tutto ciò che farete richiede i privilegi di root, così dovete iniziare loggandovi come root. Per prima cosa dovete installare il codice sorgente del kernel, ad esempio, dal CD di installazione. Nella Suse il sorgente è localizzato nella parte 'd' (per 'sviluppo') come pacchetto 'lx_kernel'. È consigliabile installare il codice sorgente del kernel allegato alla vostra distribuzione, perchè le varie GUI sono così installate automaticamente. Una volta fatto questo il tarball del nuovo kernel può essere scaricato da <http://www.kernel.org/pub/linux/kernel/v2.4/> e installato. Le corrispondenti modutils si trovano in <http://www.kernel.org/pub/linux/utils/kernel/modutils/v2.4/>. Notate che il numero di versione delle 'modutils' non corrisponde a quello del kernel. La compilazione e l'installazione delle modutils è discusso più avanti, sotto il paragrafo 'Installazione modutils'. Prima faremo il kernel stesso.

Il codice sorgente del kernel è adesso nella directory '/usr/src/linux/'. È saggio salvare questo particolare codice sorgente, ad esempio rinominando la directory linux come segue:

```
cd /usr/src
mv linux linux-2.2.19 (se il codice sorgente originale è 2.2.19).
```

Solamente quando avrete salvato il kernel originale potrete scompattare il nuovo kernel: vedrete che il file `linux-2.4.6.tar.bz2` scompatta tutto nella directory 'linux' di default. Viene sovrascritto se questa directory esiste già, e poi voi avete dei problemi: Non potete più ricompilare il kernel originale, perdetevi la configurazione originale, ecc. In questo esempio io rinomino la directory 'linux' subito dopo aver scompattato il codice sorgente del kernel 'linux-2.4.6', e io creo un link simbolico con il nome 'linux' alla directory 'linux-2.4.6'. Il vantaggio di questa procedura è che potete vedere immediatamente le versioni dei kernel presenti nel sistema. Inoltre è facile installare un aggiornamento del kernel. I comandi sono (ricordate come root):

```
cd /usr/src
cp ~/linux-2.4.6.tar.bz2 ( assumendo che il tarball sia stato scaricato )
                        ( nella vostra directory ('~') )
bzip2 -d linux-2.4.6.tar.bz2 (può volerci un po' )
tar -xvf linux-2.4.6.tar
mv linux linux-2.4.6
ln -s /usr/src/linux-2.4.6 /usr/src/linux
```

Una volta fatto questo andate nella directory del kernel, ed eseguite:

```
cd /usr/src/linux
make xconfig (vedi Figura 1)
```

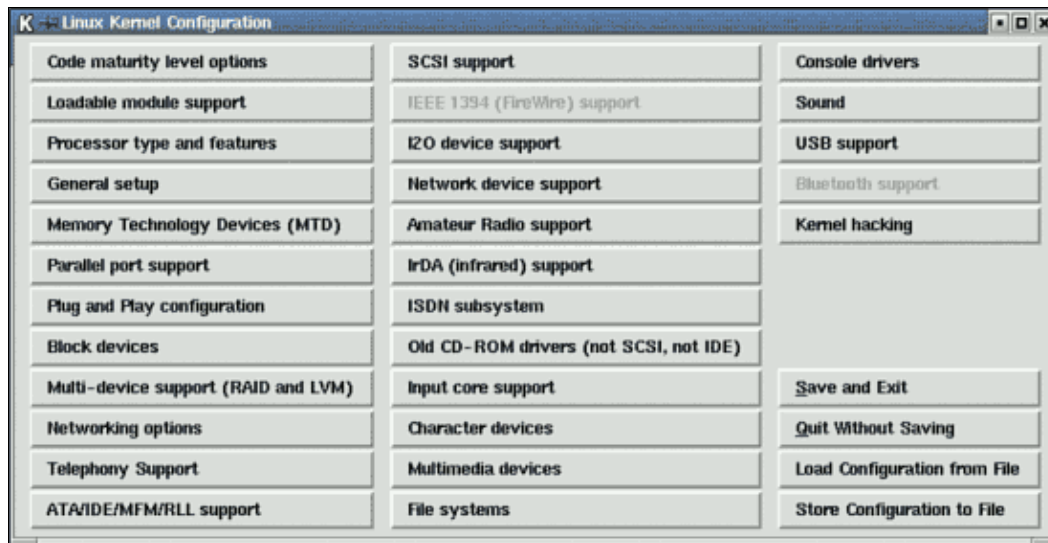


Figura 1: L'interfaccia grafica per configurare il vostro kernel Linux dopo il comando 'make xconfig'.

Questo è il menu principale per configurare il kernel. Dovete cliccare sulle varie opzioni. Cliccando 'Save and Exit' salvate le vostre impostazioni sul disco, e una volta fatto questo potete finalmente compilare ed installare il kernel (come in Figura 40). Ma non siamo ancora arrivati qui.

Configurazione del kernel

Sotto ho riprodotto i vari dump dello schermo con le mie selezioni spuntate. Ogni immagine è accompagnata da una spiegazione del perchè ho scelto quella specifica opzione. Leggendo attentamente questi esempi capirete le ragioni delle mie scelte, e capirete anche meglio quali opzioni selezionare nella vostra specifica situazione. Il pulsante 'help' fornisce informazioni simili. Potete vedere lo 'help' lanciando il vostro proprio 'make xconfig' della vostra distribuzione Linux. Poi, cliccando 'Help'. Il testo che compare normalmente raccomanda quale opzione selezionare.

Gli esempi possono naturalmente non discutere tutto l'hardware che potete avere. Comunque, possono chiarire come configurare il vostro specifico hardware, e come guardare nel kernel per vedere se il vostro hardware è supportato.

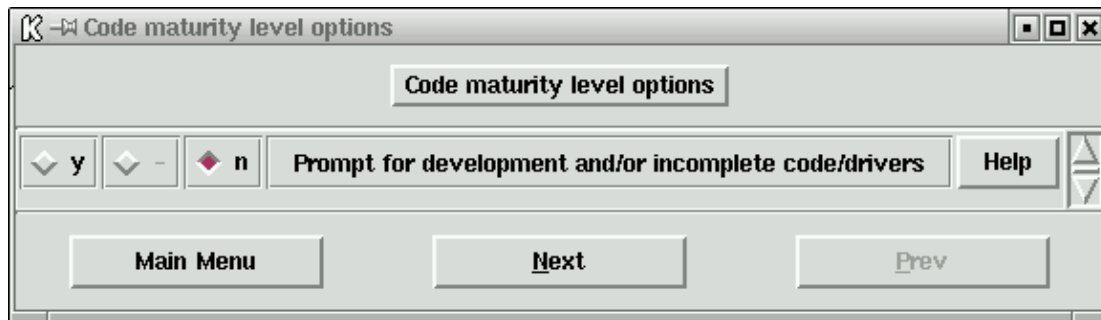


Figura 2: Selezione del 'code maturity level options'.

In questa sezione potete abilitare l'utilizzo delle opzioni sperimentali del kernel. A volte sono necessarie, ad esempio per avere un supporto ai nuovi tipi di schede. Comunque, in molti casi deselectionate questa opzione in quanto il codice sperimentale produce un kernel meno stabile. In Figura 1 vedete in grigetto le opzioni 'IEEE 1394 (FireWire) support' e 'Bluetooth support'. Con il corrente setup non potete selezionarle perchè il codice corrispondente è ancora sperimentale.

Figura 3: loadable modules support.

(da adesso i dump dello schermo saranno link, potete aprirli in una nuova finestra per vederli)

I moduli sono pezzi di codice del kernel, ad esempio, driver, che sono compilati separatamente ma idealmente nello stesso tempo della compilazione del kernel. Quindi, questo codice non è parte del kernel, ma può essere caricato e quindi essere disponibile quando ne avete bisogno. La raccomandazione generale è di compilare il kernel come moduli quando è possibile, perchè questo produce un kernel più piccolo e stabile. Attenzione: mai compilare il file system come un modulo, vedere Figura 32. Se fate questo errore e compilate il file system come un modulo, il kernel prodotto non potrà leggere il proprio file system. Poi, il kernel non potrà caricare i file di configurazione, cosa che ovviamente è un prerequisito per un corretto avvio di Linux. Vedrete che io i moduli con parsimonia: Voglio che il mio kernel sia in grado di comunicare direttamente con tutto l'hardware senza dover caricare moduli, ma questa è solo una mia preferenza.

Figure 4: Selezione di processor type and features.

Qui selezionate il tipo di processore che avete, e indicate come le varie opzioni devono essere applicate. In generale l'opzione '/dev/cpu' è piuttosto avanzata e non deve essere selezionata da molti utenti. 'High Memory Support' diventa necessaria solo se il vostro computer ha più di 1 gigabyte of RAM (non spazio disco). Molti computer hanno da 64 a 512 megabyte di ram (e da 8 a 60 Gb di hard disk), così 'High Memory Support' è usualmente deselectionata. Dovete abilitare l'opzione 'Math Emulation' se usate linux su un sistema 386 o 486SX. Questi vecchi sistemi mancano del coprocessore matematico su cui conta linux, così in questo caso dovete selezionare 'Math Emulation'. Virtualmente tutti i moderni processori hanno un coprocessore integrato, così voi potete lasciare questa opzione deselectionata. L'opzione 'MTRR' permette una più veloce comunicazione attraverso il bus PCI o AGP. Siccome tutti i moderni sistemi hanno la scheda video sul bus

PCI o AGP dovete normalmente selezionare 'MTRR': in ogni caso, è sempre meglio abilitare questa opzione anche se il vostro sistema non usa il bus PCI o AGP per la scheda video. Il supporto per multi-processore simmetrico (SMP) è applicabile solo alle piastre madri con più di un processore, ad esempio, una piastra madre con due processori Pentium II. SMP assicura che il kernel carichi ottimamente entrambi i processori. L'ultima opzione (APIC) si applica ai sistemi multiprocessore ed è normalmente disabilitata.

Figura 5: General Kernel Options.

Qui specificate alcune opzioni generali del kernel. Tutti selezionano 'Networking support' perchè ne avete sempre bisogno, ad esempio, per Internet. Linux è pesantemente Internet-oriented e non può funzionare correttamente senza la rete. Inoltre, il supporto di rete è necessario da tutti quei tipi di azioni che non sembrano aver molto a che fare con la rete. È anche possibile che il kernel non si compili senza il supporto di rete. In breve: includete il supporto di rete. Tutti i moderni sistemi usano il bus PCI così scegliete anche questa opzione. Il testo in grigio 'PCMCIA/CardBus support' mostra che questa opzione non è disponibile, perchè prima avete selezionato che non volete utilizzare il codice sperimentale (vedere Figura 2). Se usate un laptop avete bisogno del supporto PCMCIA/CardBus nel kernel per permettere l'utilizzo della rete o di un modem (vedere anche sotto il paragrafo 'pcmcia support (laptops)' più avanti). 'System V IPC' permette ai programmi di comunicare e sincronizzare, 'BSD process accounting' mantiene ad esempio, i codice di errore quando termina un processo, e 'Sysctl support' permette ai programmi di modificare alcune opzioni del kernel senza ricompilarlo o riavviare il sistema. Queste opzioni sono solitamente abilitate. Le moderne distribuzioni linux hanno il loro 'kernel core (/proc/kcore/) format' selezionato come 'ELF': questo è il formato standard delle varie librerie di sistema, parti di codice a disposizione del sistema e utilizzate dai programmi. 'ELF' è il successore del formato obsoleto 'a.out', ed è simile ai file .dll di windows. Tutti i moderni programmi linux usano le librerie ELF, ma sfortunatamente qualche vecchio programma richiede anche il supporto per il formato 'a.out'. Un esempio è 'Word Perfect 8 for XWindows': questa applicazione nativa XWindows/Linux è disponibile solo nel formato 'a.out', così che 'xwp' non funziona senza il supporto al formato 'a.out'. Includete 'a.out' come un modulo se volete utilizzare 'xwp'. Io includo anche 'MISC' come modulo. In principio non lo utilizzavo, ma è diventato comodo avere il codice disponibile se siete un frequente utilizzatore di java, python, o l'emulatore dos DOSEMU. Io ho selezionato 'Power Management support' e 'Advanced Power Management BIOS support' (non mostrato in Figura 5). Queste due opzioni sono il minimo richiesto con le moderne piastre madri ATX per permettere al kernel di spegnere automaticamente il computer quando chiudete linux. Le altre funzioni di gestione energetica sono disabilitate perchè normalmente non funzionano sotto XWindows (che è il mio standard quando uso linux). KDE e Gnome hanno le loro funzioni standard di gestione energetica che potete selezionare.

Figura 6: Configurazione Memory Technology Devices.

Avete bisogno di questa opzione per far leggere a linux, ad esempio, le schede flash. Le schede flash sono spesso utilizzate dalle macchine fotografiche digitali. Con questa opzione linux può leggere le schede flash (dall'apposito hardware) e copiare le foto sul disco come file .jpg. Non sapendo se vi serviva ho disabilitato questa opzione: Se vi serve potete sempre aggiungerla dopo.

Figura 7: Configurazione porta parallela.

Prima che esistesse la tecnologia USB la porta parallela era comunemente utilizzata per connettere stampanti e scanner. La mia stampante ha una porta parallela, quindi io voglio che sia disponibile su linux. Notate che configurare la porta parallela è diverso di configurare la stampa: questo viene fatto più avanti in Figura 28.

Figura 8: Configurazione Plug & Play.

Quasi tutti hanno un sistema 'Plug & Play' e quindi vogliono supportare questa opzione. Abilitare questa opzione permette al kernel di configurare le periferiche 'Plug & Play' e renderle disponibili al sistema. A volte è necessario abilitare l'opzione 'Plug & Play OS' nel BIOS, perchè altrimenti linux (e anche windows) non può configurare le periferiche 'Plug & Play'. L'opzione 'ISA Plug & Play support' si riferisce alle schede ISA che sono 'Plug & Play' ma che usano il bus ISA. Un esempio è la sound blaster AWE64. Il bus ISA non ha mai avuto uno standard 'Plug & Play', ciò rende difficile configurare queste schede. Prima del kernel 2.4.x, gli

utenti linux dovevano chiamare il programma 'isapnp' (pacchetto isapnptools, rpm -qil isapnptools per vedere tutti i file) durante il processo di avvio. 'isapnp' legge il file '/etc/isapnp.conf'. Questo file contiene tutte le porte, indirizzi e interrupt utilizzati dalle varie schede. Se le informazioni del file '/etc/isapnp.conf' erano incorrette, o se 'isapnp' non veniva chiamato, le schede non erano accessibili da linux e il modem, la scheda di rete o la scheda sonora non funzionavano. Selezionando l'opzione 'ISA Plug & Play support' si sostituisce la vecchia procedura: il file '/etc/isapnp.conf' non è più utilizzato. Invece, le configurazioni sono rilevate automaticamente. Sotto Suse 7.1 ho dovuto rinominare il file '/etc/isapnp.conf' in ad esempio, '/etc/isapnp.conf.old' dopo aver compilato 2.4.x, perchè sia il kernel che 'isapnp' richiedevano le stesse risorse, con conseguenze disastrose. Il problema è che Suse 7.1 (e le versioni precedenti) attivano automaticamente 'isapnp' durante il boot, anche se il kernel già contiene il supporto necessario. Questo è comunque rilevante solo per i vecchi sistemi Linux, quelli nuovi non utilizzano di default isapnp.

Figura 9: Configurazione block devices.

Virtualmente tutti utilizzano un disco floppy, così la prima opzione è abilitata (o, nel mio caso, selezionata come modulo). Alla richiesta di accedere ad un floppy il kernel carica automaticamente il modulo necessario, ammesso che il file '/etc/modules.conf' o '/etc/conf.modules' sia correttamente configurato nella vostra distribuzione come normalmente avviene. Come utente non avrete problemi con questo se avete selezionato le opzioni corrette in Figura 3. Per accedere al floppy il kernel deve naturalmente essere in grado di leggere il suo file system. Di conseguenza dovete anche copiare correttamente la Figura 32. Le altre opzioni sono importanti se utilizzate periferiche di salvataggio IDE tramite porta parallela, ma normalmente sono disabilitate. Una possibile eccezione è 'loopback device support'. Prima di masterizzare CD su linux voi normalmente fate un'immagine del CD, e il 'loopback device' è necessario per vedere il contenuto dell'immagine. Ho selezionato questa opzione (5 linee sotto) come un modulo (non mostrato in Figura 9).

Figura 10: Configurazione multiple devices.

Normalmente gli utenti di linux non hanno bisogno del supporto raid o LVM. 'Raid' implica che il sistema usi due o più dischi fissi per salvare in parallelo le informazioni. Se un disco si rompe gli altri continuano a funzionare, e il sistema continua a lavorare. LVM rende possibile aggiungere un disco fisso in modo che una partizione esistente sembri più grande. In pratica questo significa che non dovete ripartizionare o copiare una partizione più piccola in una più grande. Anche i percorsi rimangono gli stessi. Questa possibilità è molto pratica ma normalmente i piccoli utenti non ne hanno bisogno.

Figura 11: Configurazione networking options.

Avete bisogno dell'opzione 'Packet Socket' per comunicare con gli elementi di rete senza implementare nel kernel un protocollo di rete. Posso riassumere: selezionatela sempre. Molte altre opzioni sono disabilitate, a meno che non abbiate bisogno di supportarle specificatamente. Per esempio, ho selezionato 'Network packet filtering (replaces ipchains)' perchè uso il firewall standard della Suse. Un firewall protegge il vostro computer contro attacchi esterni, ad esempio, attraverso Internet, almeno quando avrete configurato propriamente il firewall. La protezione firewall a livello del kernel è ovviamente molto vantaggiosa. Ulteriori opzioni nel configurare 'network packet filtering' sono spiegate in Figura 12. Avete bisogno dei 'Unix domain sockets' per le connessioni di rete, ma anche altrove: XWindows usa automaticamente i socket Unix, così non potete usare XWindows senza questa opzione. Abilitate sempre questa. 'TCP/IP networking' contiene i protocolli necessari per Internet, e anche per le reti interne. Normalmente vorrete attivare il supporto al TCP/IP. Se dubitate nel selezionare un'opzione, provate il testo di aiuto. Se ancora non lo sapete potete sempre includere il supporto, e poi rimuoverlo più tardi durante il test. Compilare certe opzioni come moduli è anche una buona possibilità.

Figura 12: Configurazione del IP netfilter (firewall).

Per operare correttamente il firewall della Suse Linux richiede il supporto per ipchains. Quindi, per Suse ho selezionato questa opzione. Se utilizzate un firewall su altre distribuzioni o installazioni, consultate il loro manuale.

Figura 13: Configurazione telephony support.

Questa è necessaria solo se avete una scheda telefonica nel vostro computer, ad esempio, per fare una chiamata telefonica tramite Internet. Molti piccoli utenti non hanno questa scheda e non hanno bisogno di questa opzione.

Figura 14: Configurazione supporto ATA, IDE, MFM e RLL (protocolli di comunicazione per hard disk).

Quasi tutti hanno bisogno di questi protocolli, con le sole eccezioni dei sistemi che hanno solo dischi SCSI e nessun altro tipo di disco. Comunque, molti utenti selezionano questa opzione. Cliccando la linea immediatamente sotto si ottiene un sottomenu con altre opzioni. Queste sono discusse sotto. A causa della loro importanza non sono una ma tre le schermate stampate. Riempitele molto attentamente: sono estremamente importanti.

Figura 15: Configurazione supporto ATA, IDE, MFM e RLL: schermata 1.

La prima opzione è necessaria a tutti coloro che utilizzano hardware attraverso l'interfaccia IDE/ATAPI. Questo include i dischi fissi, ma anche registratori a nastro, dischi ZIP, e i lettori CD e i masterizzatori. Tutti i computer moderni usano un'interfaccia IDE/ATAPI quindi questa opzione è abilitata. L'opzione 'include IDE/ATA-2 DISK support' è necessaria per supportare i dischi fissi di sistema. Quindi, anche questa opzione deve essere abilitata, eccetto se avete solamente un sistema SCSI.

Figura 16: Configurazione supporto ATA, IDE, MFM e RLL: schermata 2.

L'opzione 'include IDE/ATAPI CDROM support' è normalmente selezionata se avete un CDROM ATAPI. Comunque, i masterizzatori CD ATAPI devono essere utilizzati tramite l'emulazione SCSI. L'emulazione SCSI può essere utilizzata per accedere sia ai CD che ai masterizzatori. Comunque, potete avere dei problemi se montate i vostri CD tramite l'emulazione SCSI, come dei messaggi d'errore quando montate i CD, o quando lanciate un CD player per ascoltare dei CD musicali. La soluzione migliore è quella di abilitare sia 'include IDE/ATAPI CDROM support' che 'SCSI emulation support' come mostrato in Figura 16. La periferica che necessita dell'emulazione SCSI, normalmente il masterizzatore, può essere definita in '/etc/lilo.conf' aggiungendo la riga: 'append="hdd=ide=scsi"': questo è discusso più avanti sotto il paragrafo 'Configurazione lilo'. Da quando ho un drive ZIP interno che comunica con la piastra madre tramite l'interfaccia ATAPI, ho selezionato l'opzione 'include IDE/ATAPI FLOPPY support'. Avete bisogno della stessa opzione per accedere ad altre periferiche simil-floppy, come i drive LS120. Molte piastre madri usano 'PCI IDE' per accedere ai dischi fissi, CDROM e floppy, quindi questa opzione è normalmente abilitata. Inoltre ci sono due possibilità di abilitare il DMA. Il DMA fornisce al vostro hardware un accesso diretto alla memoria del sistema, senza l'intervento del processore. Come risultato i dischi IDE sono accessibili più velocemente. L'opzione 'sharing PCI IDE interrupts support' è disabilitata perchè normalmente non ne avete bisogno. Veramente, alcuni controller IDE permettono di condividere gli interrupt con altre periferiche, per esempio una scheda di rete esotica. Sfortunatamente, condividere gli interrupt IDE decrementa le prestazioni dei dischi condivisi, così normalmente si condividono gli interrupt solo quando è l'unico modo di risolvere alcuni gravi problemi hardware.

Figura 17: Configurazione supporto ATA, IDE, MFM e RLL: schermata 3.

La mia piastra madre ha un Pentium II e un chipset Intel, così naturalmente voglio uno specifico supporto per questo particolare chipset. Quando riempierete le vostre opzioni del kernel vedrete gli altri chipset che non sono mostrati in Figura 17.

Figura 18: Configurazione SCSI support.

Se avete una scheda SCSI dovete naturalmente selezionare le opzioni che vi servono. La schermata mostra solamente le opzioni necessarie al vostro masterizzatore ATAPI se avete selezionato 'SCSI emulation support' (Figura 16).

Figura 19: Configurazione I2O device support.

Dovete selezionare questa opzione se avete un'interfaccia I2O nel vostro computer. Molte persone non ce l'hanno, e in questo caso dovete semplicemente disabilitarla.

Figura 20: Configurazione network device support.

Non sono mai stato in grado di compilare un kernel senza il supporto per le periferiche di rete. Dovete comunque sempre selezionare questa opzione. Potete selezionare anche il driver dummy, anche se attualmente, non c'è una rete fisica, come nel caso di molti utenti domestici. Anche se c'è una rete linux usa frequentemente il driver dummy. In questo menu potete selezionare la vostra rete e la vostra scheda di rete, come mostrato dall'esempio in Figura 21. Notate che avete bisogno di altro se volete collegarvi a Internet tramite un modem: dovete abilitare il supporto al ppp selezionando 'PPP support for async serial ports' (per le porte COM) o 'PPP support for sync tty ports' (per connessioni veloci attraverso, ad esempio, un adattatore SyncLink). Se dimenticate di farlo il kernel vi dirà che il modulo ppp non esiste, anche se l'avete fatto, un errore che non aiuta ad trovare il problema reale. Potete scegliere entrambi i metodi di compressione senza problemi: se il kernel ne ha bisogno li usa, altrimenti no.

Figura 21: Configurazione delle periferiche ethernet.

La mia scheda di rete è una 3COM /100 Mbit che usa il chipset 3c509/3c529. Sino a quando non avrò una connessione fisica con la rete (ho una scheda di rete, ma sono connesso alla rete tramite un modem) compilo questo driver come un modulo, potrei aver bisogno della scheda in futuro. Naturalmente voi sceglierete la rete e la scheda di rete della vostra macchina. Inoltre, dovete configurare la connessione di rete con un programma di configurazione, come 'yast2' sotto Suse.

Figura 22: Configurazione amateur radio support.

Selezionate questa opzione se volete supportare amateur radio, e abilitate le opzioni necessarie. Molte persone non usano questa opzione.

Figura 23: Configurazione comunicazioni infrarosso (wireless).

Abilitate la comunicazione a infrarosso se avete una periferica senza filo, ad esempio, un mouse o una tastiera senza filo. Molti sistemi desktop non ce l'hanno e non hanno bisogno di questa opzione.

Figura 24: Configurazione ISDN support.

Qui selezionate il supporto per una scheda ISDN che potete avere nel vostro sistema. È importante sapere quale scheda avete, incluso il chipset: avete bisogno di questa informazione per scegliere il driver giusto.

Figura 25: Configurazione old CDROM drivers.

Nei vecchi sistemi 486 e 386 il CDROM non è connesso attraverso il controller IDE (ATAPI) del disco fisso, ma tramite una scheda audio o una scheda speciale. Usare questi vecchi CD richiede la selezione dei driver corrispondenti. Questa opzione è sostituita nei sistemi moderni, e quindi superflua.

Figura 26: Configurazione input core support.

Questo si riferisce ad una delle principali aggiunte al kernel 2.4.x: il supporto USB. Input core support è uno strato tra il kernel e alcune periferiche USB. La Figura 38 mostra le varie periferiche USB che potete selezionare, e il testo di aiuto per alcune di queste indica quali necessitano del 'input core support': vedi Figura 38. Dovete abilitare 'input core support' se una delle vostre periferiche USB lo richiedono. Tutte le moderne piastre madri hanno una connessione USB, così come regola abilitatela. Ma, per essere onesti, non ho bisogno del supporto USB nel mio sistema così l'ho disabilitato.

Figura 27: Configurazione character devices: schermata 1.

La prima opzione ('virtual terminal') abilita la possibilità di aprire un xterm (usando XWindows) o di usare la modalità testo per il login. Normalmente questa opzione è abilitata. La seconda opzione ('support for console on virtual terminal') dice al kernel dove deve inviare i messaggi, come i warning a proposito di moduli mancanti o che funzionano male, problemi con il kernel stesso, e i messaggi di avvio. Sotto XWindows spesso

c'è una finestra a parte per messaggi del kernel, ma in modalità testo tipicamente vanno sul primo terminale virtuale ('CTRL+ALT+F1'). Lasciate abilitata questa opzione. Potete anche scegliere di spedire questi messaggi alla porta seriale, ad esempio, ad una stampante o ad un'altro terminale (la quarta opzione). Per spedire i messaggi alla stampante dovete anche attivare la porta tramite la terza opzione. Inoltre, dovete attivare questa porta se volete usare un mouse seriale. Ancora, normalmente la terza opzione ('standard/generic (8250/16550 and compatible UARTs) serial support') è abilitata. Nel mio sistema ho scelto di compilarla come un modulo. La ragione è che durante l'avvio SuSe si lamenta della mancanza di un modulo 'serial support', e includerne il supporto come modulo è una via elegante per evitare il messaggio assicurandosi che il modulo esista. Configurare il 'character devices' è estremamente importante. Se non lo fate correttamente potreste ritrovarvi con un sistema non funzionante. Quindi le Figure 28 e 30 discutono alcune opzioni in più.

Figura 28: Configurazione 'character devices': schermata 2.

Se volete utilizzare un xterm sulla vostra macchina da un sito remoto, per esempio tramite 'telnet' o 'ssh', dovete abilitare l'opzione 'unix98 PTY support'. Sembra che i sistemi autonomi non necessitino di questa opzione, ma alcuni processi di background la utilizzano. Quindi, è una buona idea attivare questa opzione in ogni caso, anche solo per evitare messaggi di errore (almeno per la SuSe) durante l'avvio. Tutti coloro che connettono una stampante tramite la porta parallela necessitano naturalmente del 'Parallel printer support'. Ancora, non tutti hanno bisogno della porta parallela: non le moderne stampanti USB. I messaggi del kernel possono essere spedite alla stampante parallela abilitando 'Support for console on line printer': normalmente, non lo vogliamo. Avete bisogno dell'opzione 'support for user-space parallel port device drivers' se avete dell'hardware che blocca la porta parallela, ma normalmente non avviene. Inoltre, usualmente non avete bisogno del 'I2C support': È necessario per alcune schede che gestiscono il video, ma se vi serve potrete sempre aggiungerlo dopo, quando saprete che il kernel lavora bene. Scegliete il supporto per il mouse e il joystick quando li usate, ma non tutti i mouse utilizzano questo driver (vedere più avanti alla Figura 29). I masterizzatori attuali hanno reso obsoleti i registratori a nastro che necessitano del 'QIC-02 Tape support', quindi questa opzione è normalmente disabilitata.

Figura 29: Configurazione 'character devices': Mouse.

Non avete bisogno di nulla da questa opzione se avete un mouse seriale, ma per tutti gli altri tipi di mouse dovete configurare alcuni parametri. Se utilizzate un'ORIGINALE bus mouse dovete selezionare la prima opzione, con sotto il corrispondente tipo di bus mouse. Molti odierni computer hanno un'altro tipo di mouse, normalmente (ed erroneamente) chiamati 'busmouse' o 'PS/2 mouse'. Questi mouse sono spesso connessi al '/dev/aux' e inseriti in un piccolo connettore simile a quelli utilizzati per le tastiere. Spesso questo tipo di mouse utilizza la tastiera per connettersi al computer. Per far funzionare propriamente questi mouse dovete selezionare le opzioni come mostrato in Figura 29, 'mouse support (not serial and bus mice)' e 'PS/2 mouse (aka "auxiliary device" support)'.

Figura 30: Configurazione 'character devices': schermata 3.

Le opzioni per configurare il kernel tra la Figura 28 e la Figura 30 non sono discusse qui. Sono normalmente disabilitate. L'opzione 'Ftape, the floppy tape device driver' si riferisce al supporto dei registratori a nastro connessi tramite il controller floppy. Anche se avete questo tipo di periferica non è essenziale compilarne il supporto, almeno non in questa prima fase. Le altre opzioni si riferiscono alle moderne schede video 3D. Se avete una scheda video connessa tramite il bus AGP, potete abilitare il supporto AGP, e anche il driver specifico per la vostra scheda video (sotto '/dev/agpgart (AGP support)'). Notate che è possibile avere un kernel perfettamente funzionante senza questa opzione, ma non necessariamente! Le persone che hanno una scheda video integrata nella piastra madre, come nel chipset intel i815, DEVONO usare il driver del kernel! Altrimenti, XWindows 4.0 o superiori (usati nelle più recenti distribuzioni) non funzionano. Il mio sistema ha una scheda AGP, una NVidia TNT2, ma questa scheda non è supportata con un modulo specifico del kernel (NVidia si rifiuta di distribuire le specifiche hardware, necessarie per sviluppare i driver). Purtroppo, nel mio caso ha poco senso abilitare il supporto AGP. Nonostante questo problema, posso usare XWindows 4.0 senza il driver del kernel. 'Direct rendering support' è un'opzione in XWindows a partire dalla versione 4.0 per

accelerare le prestazioni grafiche tramite il kernel. Per usare questa opzione la vostra specifica scheda video deve essere supportata, e dovete usare XFree86 4.0 o superiore. Inoltre dovete anche attivare 'AGP support'. Ancora, potete tranquillamente disabilitare queste opzioni e avere un kernel linux perfettamente funzionante.

Figura 31: Configurazione 'multimedia devices'.

Questa opzione è abilitata se avete una scheda che gestisca video o radio. Come prima, questa opzione non è essenziale per un kernel propriamente funzionante.

Figura 32: Configurazione 'file systems': schermata 1.

Qui potete specificare i filesystem leggibili dal kernel linux. Potete volere un kernel che possa leggere i dischi o i floppy Windows, ma dovete essere sicuri che il kernel possa leggere il proprio ext2 filesystem, o il nuovo ReiserFS. Linux può anche non avviarsi se sbagliate, perchè il kernel può non leggere il proprio disco di avvio (come detto precedentemente in Figura 3). Per leggere dischi e floppy DOS/Windows dovete abilitare l'opzione 'DOS FAT support': Comunque, per leggere dischi Windows NT/Windows 2000 avete bisogno di un separato driver a sola lettura che può essere selezionato nel prossimo menu. Per leggere e anche scrivere dischi e floppy DOS/Windows necessitate dell'opzione 'MSDOS fs support'. Virtualmente tutti lo vogliono, così molte persone abilitano questa opzione. 'VFAT' si riferisce al supporto dei nomi lunghi dei file sotto Windows 95 o 98. Il mio sistema è un cosiddetto sistema dual boot, nel quale posso avviare sia Windows 98 che Linux (usando il boot manager lilo, vedere sotto 'Configurazione lilo'). Quindi ho attivato 'VFAT'. Dovete includere il supporto per ISO 9660 per leggere i CD nel formato standard. Sotto c'è l'opzione 'Joliet extensions', che permette nomi di file più lunghi del MSDOS 8.3 che è il limite dello standard ISO 9660. Quasi tutti vogliono leggere i moderni CD, così queste opzioni sono normalmente attivate. La Figura 33 chiarisce alcune opzioni aggiuntive fra le quali c'è Linux ext2 file system.

Figura 33: Configurazione file systems: schermata 2.

I file nella directory '/proc' contengono informazioni riguardanti lo stato del sistema, ad esempio, quali interrupt sono in uso. Normalmente abilitate sempre questa opzione. L'opzione 'Second extended fs support' si riferisce al file system (ancora) standard di linux. Dovete assolutamente compilarlo (NON come modulo)! La Figura 32 e 33 non mostrano l'opzione RaiserFS che può essere selezionata qui: Successore dell'ext2, RaiserFS gestisce meglio i danni dovuti ad una mancanza di alimentazione o problemi simili. Attualmente RaiserFS è ancora in fase di sviluppo e quindi visto come codice sperimentale. Nondimeno alcune recenti distribuzioni supportano già l'uso di RaiserFS, ma anche se si suppone che RaiserFS sostituirà 'ext2' in futuro, io non lo consiglio come file system per tutte le partizioni in questo momento. Necessitate del 'UDF file system support' se usate (sotto Windows) il programma 'packetCD', che vi permette di copiare al volo file sul CD come su un lento disco fisso. È molto comodo per scambiare dati con altri PC. Leggere questi packetCD anche sotto linux è possibile caricandoli con il file system 'udf', ad esempio, con un comando quale 'mount -t udf /dev/scd0 /cdrom'. Questa parte contiene anche elementi come 'Network file systems', 'partition types' e 'Native language support'. Non dovete avere a che fare con 'Network File Systems' a meno che il vostro computer non faccia parte di una grande rete, nel qual caso dovete abilitare 'NFS File System Support' e forse anche 'SMB file support', ma per un computer isolato non avete bisogno di queste opzioni. L'opzione 'Partition Types' è abbastanza avanzata ma non necessaria per un effettivo utilizzo del kernel linux. È meglio disabilitarla. Le Figure 34 e 35 spiegano 'Native Language Support'.

Figura 34: Configurazione 'native language support': schermata 1.

In questo menu selezionate quale tabella di codici verrà usata da linux per gestire i nomi dei file sotto DOS e Windows. La tabella codici di Figura 34 è per gli usuali nomi file del DOS. La tabella NLS in Figura 35 è necessaria per i nomi di file lunghi. La prima opzione in Figura 34, 'Default NLS option', determina quali simboli saranno standard in linux. La Figura 35 descrive e spiega l'opzione 'iso8859-15'.

Figura 35: Configurazione 'native language support': schermata 2.

Avete bisogno dell'opzione 'NLS ISO 8859-15' per riprodurre correttamente la FAT Windows e le estensioni Joliet del file system dei CD, che è sempre una buona idea. Selezionare 'NLS ISO 8859-15' è appropriato per

i linguaggi Occidentali, e include il simbolo dell'Euro. Quindi questa tabella di codici è quasi sempre compilata. La tabella 'NLS ISO 8859-1' è la precedente tabella per i linguaggi occidentali ma senza il simbolo dell'Euro.

Figura 36: Configurazione console drivers.

'VGA text console' è per abilitare la modalità testo in risoluzione VGA. Quasi tutti vogliono questo, così questa opzione quasi sempre attivata. Solo pochi vecchi computer 386 non hanno una scheda VGA compatibile, mentre molti moderni computer non hanno il minimo problema con questa scelta. La seconda opzione, 'video mode selection support', rende possibile scegliere la risoluzione della modalità testo durante il processo di avvio. Questa è a volte comoda per avere più lettere su una linea, ma normalmente la lascerete disabilitata. Le ultime due opzioni sono sperimentali, e vi consiglio di deselezionarle.

Figura 37: Configurazione audio.

In questa sezione configurate l'audio. Se la vostra distribuzione usa i driver audio ALSA (come la SuSe 6.3 e superiori), è sufficiente selezionare 'sound card support' come MODULO. I driver ALSA sono compilati e collegati più avanti (vedere avanti sotto il paragrafo 'SuSE e i driver audio ALSA'). Se la vostra distribuzione usa i driver audio standard del kernel dovete selezionare quello corretto per la vostra scheda audio. Virtualmente sono indicate tutti i marchi di schede audio, così in linea di principio la selezione del driver corretto non è un problema. Se la vostra scheda audio funziona bene con il kernel standard della vostra distribuzione potete anche usare dei programmi di configurazione (come 'yast2' per la SuSE) per trovare quale driver specifico sia necessario per la vostra scheda audio. Sappiate che l'audio non è critico: mancherà l'audio se qui qualcosa va male, ma il kernel funzionerà ugualmente bene.

Figura 38: Configurazione 'USB support'.

La mia piastra madre ha una porta USB, ma non la uso. Comunque, se il supporto USB è disabilitato SuSE mi da un errore durante l'avvio. Naturalmente SuSE supporta l'USB e quindi tenta di caricare i moduli necessari, da qui la mia selezione del 'Support for USB' come modulo. Anche se questo messaggio d'errore non è importante, l'ho risolto in un modo elegante compilando il driver necessario per le porte USB della mia piastra madre. Per fare questo il minimo è impostare il 'Preliminary USB device filesystem' a 'y' e caricare uno specifico driver USB. Siccome la mia piastra madre Pentium II è un po' vecchia, ho scelto il driver 'UHCI (Intel PIIX4, VIA, ...)' come modulo. Ma, dovete scegliere il modulo 'UHCI Alternate Driver (JE) support' se avete una piastra madre recente con il chipset Intel, mentre per esempio, i computer Compaq dovranno scegliere come modulo 'OHCI support'. In linea di principio avete bisogno solamente di uno di questi tre moduli, ma in caso di dubbio potete selezionarli tutti e tre. La vostra distribuzione linux ha già trovato quale vi serve, e ha caricato automaticamente quello corretto.

Abilitare semplicemente le poste USB della vostra piastra madre non è sufficiente, dovete anche specificare i driver (moduli) delle periferiche USB connesse al vostro computer.

Figura 39: Configurazione 'kernel hacking'.

Questo non è facile: NON selezionatelo! È un'opzione utile per i programmatori che vogliono vedere il motivo per un crash del kernel o leggere la cache del disco fisso: l'opzione è completamente inutile per un utente medio.



Figure 40: Save and Exit.

Pffft, ce l'abbiamo fatta. Tutto ciò che rimane è compilare e installare il kernel come discusso sotto.

Compilazione del kernel

Cliccando 'Save and Exit' salvate la vostra configurazione nel file './.config' (o '/usr/src/linux/.config' se compilate in /usr/src/linux). D'altra parte, è comodo salvare questo file e copiarlo nella directory sorgente del nuovo kernel se fate solo un aggiornamento minore, ad esempio dal kernel 2.4.5 al 2.4.6. In questo modo potete (usualmente) mantenere tutte le vostre scelte, che possono risparmiarvi molto lavoro. Allo stesso modo potete partire utilizzando il file di configurazione del kernel 'standard' della vostra distribuzione, che in alcune distribuzioni è chiamato /boot/config (copiatelo in ./config per iniziare ad usarlo). Ma, se aggiornate il sorgente del vostro kernel e avete degli strani problemi quando compilate il kernel questo è naturalmente il primo file da rimuovere! Com'è naturale dovete, per questioni di sicurezza, copiare e salvare accuratamente la configurazione del kernel che funziona correttamente. La procedura per compilare il kernel è la seguente:

```
make dep
make clean (per i vecchi kernel)
make bzImage
make modules
make modules_install
```

La Figura 40 mostra già la necessità del comando 'make dep'. Naturalmente eseguite questo comando nella directory sorgente, normalmente '/usr/src/linux'. I kernel della serie 2.0.x o più vecchi necessitano anche del comando 'make clean', che rimuovono i precedenti file prima della compilazione del nuovo kernel. Il comando 'make clean' previene degli strani messaggi di errore che sono difficili da rintracciare, ma che presumibilmente sono causati da vecchi file oggetto (.o) che non vengono sovrascritti. Il comando 'make bzImage' compila il nuovo kernel, ma non lo installa. Potete anche compilare il kernel con altri comandi 'make', ad esempio, 'make bzlilo' o 'make zImage', ma questi comandi possono dare degli errori inaspettati. Molti kernel sono troppo grandi per permettere una corretta esecuzione di 'make zImage': Ottenete un messaggio di errore durante la compilazione, e terminate senza un kernel. Con il comando 'make bzlilo' tutto deve essere configurato correttamente nel file '/etc/lilo.conf', ma questo non è sempre vero. Quindi è preferibile evitare questi comandi. Il comando 'make modules' compila i moduli: i quali sono installati con il comando 'make modules_install'. Questo comando mette i moduli nella directory '/lib/modules/2.4.6/' se la versione attuale del kernel è la 2.4.6: cambia quando compilate una nuova versione del kernel. In questo modo i moduli corrispondenti ad un particolare kernel finiscono automaticamente in una directory separata, evitando conflitti con moduli obsoleti o problemi simili. Durante l'avvio il kernel linux può adesso sapere in quale directory trovare i moduli corretti. Ma, i file in '/lib/modules/2.4.6/' sono sovrascritti e rimangono vecchi

moduli se avevate precedentemente già compilato il kernel 2.4.6 e adesso lo ricompilate. Pio, i vecchi moduli possono rimanere in giro anche se non sono più necessari nel nuovo kernel. Normalmente questo non è un problema, ma è sempre una buona idea prendersi il tempo di cancellare i vecchi moduli prima di installare quelli nuovi.

per evitare problemi installando il kernel dovete anche assicurarvi che la configurazione del lilo '/etc/lilo.conf' sia corretta, e dovete copiare il kernel e il file 'System.map' nella locazione corretta. Dopo tutto questo dovete anche eseguire il comando 'lilo'. Un'alternativa è usare 'loadlin', che permette l'avvio la kernel linux sotto Dos/Windows. Entrambe le opzioni sono discusse sotto.

Configurazione lilo

Potete normalmente trovare il file di configurazione del lilo nella directory '/etc' con il nome '/etc/lilo.conf'. Apritelo in una finestra di testo con un semplice editor ASCII: Potete avere installato un potente edito come XEmacs (allora, usate 'xemacs /etc/lilo.conf &'), un semplice editor a tutto schermo come kedit o gedit (allora, usate 'k(g)edit /etc/lilo.conf') o un primitivo editor testuale come pico o nano (allora, usate 'pico /etc/lilo.conf'). Il file lilo.conf mostrerà qualcosa come:

```
boot      = /dev/hda
vga       = normal
read-only
menu-scheme = Wg:kw:Wg:Wg
lba32
prompt
timeout = 300
message = /boot/message

    other = /dev/hda1
    label = win98

image = /boot/bzImage
label = linux-2.4.6
root = /dev/hda3
append = "parport=0x378,7 hdd=ide-scsi"

image = /boot/vmlinuz.suse
label = suse
root = /dev/hda3
append = "hdd=ide-scsi"
initrd = /boot/initrd.suse
```

Il contenuto dettagliato del file lilo.conf può differire da quello sopra su ogni sistema e tra distribuzioni. Quindi adesso io utilizzerò questo file. Le prime 8 righe sono già corrette e non avete necessità di cambiarle. Il comando 'boot' nella prima riga indica il disco fisso fisico da cui avviarsi, cioè, 'boot' punta alla locazione del 'master boot record'. Nel mio caso l'avvio parte da /dev/hda, il primo disco fisso fisico. L'opzione 'vga' indica che l'avvio usa la modalità testo VGA standard con 80x25 caratteri. L'opzione 'read-only' significa che il processo di avvio monterà inizialmente in sola lettura la partizione linux. Durante l'avvio di linux le partizioni sono controllate alla ricerca di errori: solo dopo vengono rimontate con i permessi di lettura e scrittura. La linea 'menu-scheme' imposta i colori del menu di avvio in modalità testo 'lilo'. Con 'lba32' è possibile avviare il sistema operativo dopo il 1024esimo cilindro, ammesso che questo sia supportato dal BIOS. Tutti i moderni sistemi supportano 'lba32'. I problemi con questo possono essere risolti con un aggiornamento del BIOS, cosa quasi necessaria per i grandi dischi fissi oggi disponibili. Il comando 'prompt' costringe il 'lilo' a mostrare un

prompt che permette all'utente di scegliere il sistema operativo desiderato. L'opzione 'timeout' mostra il numero di millisecondi che 'lilo' aspetta per un input dopo il prompt prima di avviare il sistema operativo standard. Se 'lilo.conf' non ha un sistema operativo standard, come nell'esempio, il processo di avvio partirà con il primo sistema operativo incontrato. Nel mio caso è Windows98, così le persone che non conoscono linux finiranno nell'ambiente Windows. L'opzione 'message' mostra un messaggio mentre 'lilo' è in esecuzione. Sotto SuSE è Tux, l'attraente pinguino mascotte di linux, con (naturalmente) il testo 'SuSE Linux 7.1'. Potete vedere questo messaggio digitando 'xv /boot/message' o 'gv /boot/message' (a volte perfino 'gimp /boot/message') : 'xv' e 'gv' (ghostview) sono programmi shareware con i quali potete vedere vari tipi di formati immagine. Notate che il file /boot/message non esiste nei sistemi senza una schermata grafica di login (ad esempio vecchie distribuzioni), in questo caso il 'boot message' è un messaggio di testo. È in linea di principio possibile mostrare la vostra immagine preferita durante l'avvio, ma non ho ancora provato questa possibilità. Tutte le opzioni di 'lilo' sono naturalmente documentate nelle pagine 'man', a cui potete accedere con i comandi 'man lilo' e 'man lilo.conf'.

Le altre opzioni controllano l'avvio dei vari sistemi operativi. Al massimo potete avviare sedici differenti sistemi operativi o kernel. Normalmente questo è troppo. Scegliete il sistema operativo con la linea 'label='. Il default per Windows98 (e anche per le vecchie versioni di Windows e DOS, ma non per Windows NT o Windows 2000) è di avere di essere nella prima partizione primaria. Quindi questi sistemi operativi necessitano solo di una linea 'other' e una linea 'label'. La seconda sezione, che inizia con 'image=/boot/bzImage', avvia il nuovo kernel 'linux-2.4.6'. La mia directory radice è '/dev/hda3'. La linea 'append = "parport=0x378,7 hdd=ide-scsi"' dice al kernel l'indirizzo e l'interrupt per la porta parallela (porta 0x378, interrupt 7), e specifica che il mio masterizzatore 'hdd' deve essere indirizzato tramite l'emulazione SCSI. Il nome dei CD dipende dal sistema: nel mio è 'hdd' ma nel vostro potrebbe essere un'altro. L'uso di un'interrupt è principalmente una questione di preferenze personali. Un interrupt velocizza la stampa, ma potete tralasciare questo comando se non avete un'interrupt disponibile per la porta (stampante) parallela. Il default per linux è il cosiddetto 'polling', che permette al kernel l'utilizzo della porta parallela senza un'interrupt. L'ultima sezione comincia con 'image = /boot/vmlinuz.suse', contiene la configurazione del lilo fatta durante il processo di configurazione della SuSE: Ho aggiunto manualmente la linea 'append="hdd=ide-scsi"'. Il file 'boot/vmlinuz.suse' è il kernel standard che viene installato con la distribuzione. Preferibilmente, dovete SEMPRE salvare questo kernel per le emergenze. La linea 'initrd = /boot/initrd.suse' si applica solo per all'installazione standard del kernel: specifica il caricamento della cosiddetta immagine 'ramdisk', un disco virtuale che viene caricato in memoria (o memoria ad accesso casuale RAM). Il 'ramdisk' contiene i moduli necessari per un corretto avvio di linux: un kernel di distribuzione deve naturalmente essere in grado di accedere ad un'enorme varietà di hardware, cosa fattibile solo utilizzando molti moduli.

Adesso è chiaro dove salvare il nuovo kernel linux prima di eseguire lilo. In questo esempio, i comandi corretti sono:

```
cp /usr/src/linux/arch/i386/boot/bzImage /boot
cp /usr/src/linux/System.map /boot/System.map-2.4.6
lilo
```

Sotto SuSE 7.3 potete anche fare una seconda copia con:

```
cp /usr/src/linux/System.map /boot
```

(il System.map originale è già stato rinominato)

Se avete già un kernel con il nome 'bzImage' e volete tenervelo, potete copiare il nuovo kernel su '/boot/bzImage-2.4.6' e fare la corrispondente modifica in /etc/lilo.conf: cambiare /boot/bzImage in /boot/bzImage-2.4.6. La compilazione ricrea sempre il file System.map, che contiene il nome e la configurazione delle variabili importanti del kernel. Il comando 'depmod -a' crea un file che contiene tutte le

dipendenze dei moduli del kernel, che sono, le varie relazioni tra il kernel e i moduli, tra i moduli stessi, e anche le informazioni del file '/etc/modules.conf'. Molte distribuzioni linux, inclusa SuSE, si assicurano che 'depmod -a' sia eseguito durante l'avvio, ma è saggio assicurarsi esplicitamente che il file '/boot/System.map' esista, e che corrisponda alla propria versione del kernel. Il comando 'lilo' installa la nuova configurazione di un vecchio kernel, o di uno nuovo. Se avete già un file '/boot/bzImage' che viene sovrascritto con uno recentemente compilato ma senza eseguire lilo, vi ritroverete con un nuovo kernel che non può avviarsi. Il vecchio, originale kernel della distribuzione che avevate salvato funziona ancora, perchè non è stato sovrascritto. Tenendo il vecchio kernel rende responsabile la compilazione e il test dei nuovi kernel.

Usare loadlin

Se usate 'loadlin' per avviare linux sicuramente sapete dove trovare questo comodo programma, in C:\loadlin. Virtualmente tutte le distribuzioni linux hanno 'loadlin' nel primo CD nella directory 'dosutils'. Dovete anche copiare il nuovo kernel nel disco fisso in 'C:\loadlin', forse con un nome unico. Avviando Windows98 in modalità DOS potete avviare linux con il comando:

loadlin bzImage

In circostanze normali molte configurazioni che sono salvate nel kernel stesso, come la locazione della partizione radice, sono corrette se avete compilato il vostro kernel, e in questo caso il comando sopra indicato è sufficiente per avviare linux senza problemi. Digitando (in DOS) il comando 'loadlin | more' avrete una schermata di aiuto che comprende un collegamento al 'loadlin-HowTo' su Internet. Con 'loadlin' potete provare un nuovo kernel anche se siete riluttanti a farlo con 'lilo': 'loadlin' e 'bzImage' assieme stanno in un floppy da 1.44 MB. Non c'è assolutamente nulla che possa andare male se prima avviate il DOS con un dischetto di avvio (che supporti EMM386), e poi avviate linux dal dischetto con 'loadlin' e 'bzImage'. Ma, dovete naturalmente avere un dischetto di avvio DOS.

SuSE e i driver audio ALSA

SuSE utilizza i driver audio standard ALSA (Advanced Linux Sound Architecture). Questi driver hanno una migliore qualità rispetto ai driver del progetto OSS che sono standard nel kernel. Se vi interessa l'audio sotto linux dovete assolutamente usare i driver ALSA. Questi driver non sono parte del (vecchio) codice sorgente del kernel, il che implica che i driver devono essere compilati ed installati separatamente. Il codice sorgente per i driver ALSA sono localizzati nella parte 'zq' della distribuzione SuSE: leggete se non sapete cosa significa. Installate ALSA tramite YaST o YaST2 così avrete il codice sorgente disponibile nella directory '/usr/src/packages/'. Per compilare ed installare i driver ALSA eseguite:

```
rpm -bb /usr/src/packages/SPECS/alsa.spec  
cd /usr/src/packages/BUILD/alsa/alsa-driver-<version number>/  
./configure  
make install
```

La prima linea installa il codice sorgente, inclusi i driver, nella directory '/usr/src/packages/BUILD/'. Inoltre le librerie e le utility ALSA sono direttamente compilate come file rpm. Sfortunatamente, i driver ALSA non sono compilate di default. Dovete compilarli ed installarli separatamente a mano utilizzando i due comandi sotto. Il comando './configure' trova le necessarie configurazioni e i file nel vostro sistema e li inserisce in un file di configurazione. Il comando 'make install' compila tutti i driver ALSA e li installa allo stesso tempo per utilizzarli con il kernel nella directory '/lib/modules/2.4.6/misc/'. Adesso, quando avviate SuSE il desiderato

driver audio viene installato automaticamente. Ammetto che la procedura suggerita è un po' complessa, dovete sapere esattamente cosa fare per trovare i driver necessari e includere il supporto audio in un nuovo kernel nel modo più diretto.

Se non usate SuSE, o se volete utilizzare una versione più recente dei driver ALSA, potete scaricare questi driver e le librerie e le utility corrispondenti da <http://www.alsa-project.org>. La prima pagina di questo sito dà le ultime notizie sul progetto ALSA (ad esempio, nel febbraio 2002 l'integrazione dei driver ALSA nella serie ufficiale dei kernel 2.5) e i collegamenti ai vari file da scaricare. Sotto mostro come compilare i driver ALSA: potete seguire gli stessi passi per le librerie e le utility. Scompattate i driver in una directory conveniente, ad esempio, '/usr/local/'. Spostatevi in questa directory, in questo caso '/usr/local/alsa-driver-<version-number>' e eseguite i comandi sopra indicati cominciando con './configure'. È possibile che dobbiate fare alcuni ulteriori passi per avere un driver funzionante se la vostra distribuzione non usa i driver ALSA come standard. Sfortunatamente, questi problemi sono fuori dallo scopo di questo già esteso articolo, ma potete aver ulteriore aiuto dalle FAQ (Frequently Asked Question = Domande Frequenti) ALSA che potete scaricare.

Supporto per pcmcia (laptops)

Il supporto per pcmcia è standard nei kernel dal 2.4.x in avanti. Comunque, il PCMCIA HOWTO ufficiale afferma che la versione PCMCIA del kernel preferibilmente non dovrebbe essere usata. Per adesso, i sorgenti pcmcia, inclusi gli script e i driver possono funzionare con tutti i kernel: 2.0, 2.2, e 2.4. Ho sperimentato che i driver pcmcia smettono di funzionare dopo una ricompilazione del kernel, e devono essere ricompilati anche loro. Ci sono due soluzioni e dipendono dalla vostra distribuzione. Installare e compilare il codice sorgente in un file rpm vi dà un file che potete installare. L'altra soluzione è scaricare, scompattare, compilare e installare la più recente versione pcmcia da <http://sourceforge.net/projects/pcmcia-cs/>, con questo:

```
cp /etc/rc.d/pcmcia /etc/rc.d/pcmcia.SuSE
cp ~/pcmcia-cs-3.1.?.tar.gz /usr/src
cd /usr/src
tar -zxf ./pcmcia-cs-3.1.?.tar.gz
make config
make all
make install
cp /etc/rc.d/pcmcia.SuSE /etc/rc.d/pcmcia
```

La prima e l'ultima riga risolvono uno specifico problema di SuSE. Lo script di inizializzazione pcmcia di SuSE '/etc/rc.d/pcmcia' è sovrascritto dal comando 'make install', il che causa un fallimento dello script sotto SuSE. Il problema è risolto copiando lo script originale dopo 'make install'. Se avete erroneamente sovrascritto lo script originale dovete reinstallare il pacchetto pcmcia cominciando dalla parte 'a1', copiare lo script originale in un'altro file, eseguire ancora 'make install' e finalmente ricopiare lo script originale.

Il nuovo file rpm per il supporto pcmcia lo trovate sotto SuSE come segue:

```
rpm -i /cdrom/suse/zq1/pcmcia-3.1.?.rpm
cd /usr/src/packages
rpm -bb ./SPECS/pcmcia-3.1.?.spec
cd /RPMS/i386/
rpm -i --force ./pcmcia-3.1.?.rpm
SuSEconfig
```


Nella prima linea presumo che abbiate installato i driver pcmcia dal sesto o settimo CD e che il CD-ROM sia già montato su /cdrom. Il comando 'rpm -i' installa il codice sorgente e il comando 'rpm -bb' compila il file rpm pcmcia. Poi installate questo particolare file rpm come tutti gli altri file rpm. Notate che dovete usare l'opzione '--force' perchè altrimenti il programma rpm vi dirà (correttamente) che 'pcmcia' è già installato, e quindi ignorerà il nuovo file. Come sempre dovete eseguire il programma SuSEconfig (notate le maiuscole e le lettere minuscole) quando avete installato manualmente i file rpm sotto SuSE, per attivare le modifiche di configurazione. L'ultimo è fatto automaticamente dal programma di setup YaST o YaST2 della SuSE dopo che hanno installato o modificato i nuovi pacchetti. Quindi, non è più necessario attivarli manualmente.

Per usare propriamente il supporto pcmcia dovete lasciare attivato 'network support' durante la compilazione, ma dovete disabilitare tutti gli altri driver per schede di rete. E, come già discusso in Figura 11, dovete naturalmente abilitare 'TCP/IP support' se volete usare Internet.

Installazione 'modutils'

Come già menzionato il kernel usa dei piccoli programmi in 'modutils' per gestire i moduli del kernel. Questi programmi includono:

```
insmod (che installa un modulo),  
rmmod (per rimuovere un modulo, e )  
lsmod (mostrare tutti i moduli in uso),
```

tra molti altri. Con comandi tipo 'man lsmod' potete trovare come funzionano i vari comandi, cosa che non verrà mostrata qui.

La compilazione e l'installazione dei 'modutils' è diretta. Semplicemente fate:

```
cd /usr/src  
cp ~/modutils-2.4.6.tar.bz2 . ( assumendo che il file stia nella )  
                          ( vostra directory, '~' )  
bzip2 -d modutils-2.4.6.tar.bz2 ( unzip: richiede del tempo )  
tar -xvf modutils-2.4.6.tar  
cd modutils-2.4.6 ( andate nella directory dove i 'modutils' sono )  
                  ( stati scompattati )  
./configure (trova i specifici file di configurazione )  
make ( compila 'modutils': siccome è piccolo la compilazione può )  
      ( essere sorprendentemente veloce )  
make install (installa 'modutils' nella directory '/sbin/' )
```

Questo è tutto ciò che dovete fare per avere i 'modutils' pronti per l'uso. Notate che in questo esempio i 'modutils' sembrano avere lo stesso numero di versione del kernel, ma non è sempre questo il caso.

Il kernel funziona correttamente?

Il nuovo kernel è configurato, compilato e probabilmente anche installato tramite lilo. RIavviate il sistema e vi chiedete: come posso vedere se il nuovo kernel funziona correttamente? Potete facilmente trovare l'hardware che non funziona quando provate delle cose, ma inoltre il kernel fornisce molte utili informazioni sullo schermo durante l'avvio, anche prima che parta il login grafico. Questa informazione contiene le

configurazioni automaticamente rilevate (porte, IRQ, ecc.), ma anche i messaggi di errore nel caso in cui alcuni driver non possano essere inizializzati direttamente, sia come moduli sia come parti del kernel. Guardare questi messaggi da un primo avviso di certi problemi. Per linux SuSE questo è molto facile: ogni parte del kernel che si avvia scrive sul lato destro dello schermo il messaggio (verde) 'done' se funziona bene, o 'failed' se qualcosa va male. Poiché i messaggi possono scorrere sullo schermo ad una velocità furiosa, ogni messaggio di errore può essere ricapitolato sopra il prompt di login. Trovate il prompt di login sotto '<Ctrl>+<Alt>+F1' se avete automaticamente la schermata grafica di login. In questo modo avete almeno un suggerimento di dove occorre il problema se effettivamente il messaggio di errore è importante per voi. I messaggi di errore che somigliano a 'Cannot find module' o 'Cannot load module' indicano usualmente che non avete incluso alcune parti che non potevate tralasciare. Correggere la configurazione del kernel e ricompilarlo normalmente risolve il problema. Notate che non è necessario ricompilare l'intero kernel. Se avete dimenticato alcuni moduli è sufficiente rieseguire 'make modules' e 'make modules_install'. Non è comunque un problema se ricompilate l'intero kernel. Molto codice del kernel è già compilato, e le sole parti che devono essere ricompilate sono quelle mancanti. Ricapitolando, semplicemente modificando un po' il kernel e ricompilarlo è molto veloce, in contrasto con la compilazione iniziale del kernel. Dove potete tranquillamente prendere del caffè.

Un'altro modo di ispezionare i messaggi di avvio del kernel è con il comando 'dmesg'. Eseguendo semplicemente questo comando si richiamano i messaggi precedentemente scorsi fuori dallo schermo. Indirizzandoli in un file, con 'dmesg > temp', vi permette di leggere i messaggi con il vostro modo (con 'più tempo' o con il vostro editor favorito).

Conclusione

Con questa guida in mano potete cominciare i vostri esperimenti con il kernel in un modo ragionevole. Sperando che la soglia di partenza per lavorare con il kernel sia abbastanza bassa da farvi desiderare di partire. Molto del vostro tempo verrà speso per decidere la corretta configurazione del kernel, mentre durante la compilazione potete tranquillamente giocare a 'freecell' o fare altri lavori al computer.

Se siete ancora nei guai senza via di scampo, potete usare le varie mailing list riguardanti linux e i siti web dove potete fare delle domande. Questi esistono in tutte le lingue, non solo in inglese. In un tempo ragionevolmente breve avrete una risposta soddisfacente che vi permetterà di risolvere i vostri problemi. Il miglior modo di trovare queste liste e i siti web è con un motore di ricerca.

Webpages maintained by the LinuxFocus Editor team

© Philip de Groot

"some rights reserved" see linuxfocus.org/license/

<http://www.LinuxFocus.org>

Translation information:

nl --> -- : Philip de Groot <philipg/at/authors.linuxfocus.org>

nl --> en: Nino R. Pereira <pereira/at/speakeasy.org>

en --> it: Fabio Baseggio <base/at/trevinet.it>