RICOMPILARE UN KERNEL SLACKWARE PER ABILITARE IL SYMMETRIC MULTI-PROCESSING

evilsocket
http://www.evilsocket.net

.: Introduzione

In questi giorni mi sto trovando a dover moddare un sistema GNU/Linux per adattarlo il più possibile alle mie esigenze e voglio condividere con voi uno dei passaggi più delicati che ho dovuto affrontare .
Il mio pc, come tanti altri di ultima generazione, ha un processore Intel Centrino Duo, quindi ha due core ... giorni fa mi è venuto il dubbio “ma il mio sistema li vede e sfrutta entrambi i core oppure no ?” ... beh, la verifica è stata la seguente .

```
ls /proc/acpi/processor/
```

e come output

```
CPU0/
```

oooops, ma vedo solo una cpu !!! vabbè ho pensato, magari il sistema considera i due core come facenti parte di un unica cpu, quindi verifichiamo anche questo .

```
cat /proc/cpuinfo
```

e come output

```
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 6
model           : 14
model name      : Genuine Intel(R) CPU           T2300  @ 1.66GHz
stepping        : 8
cpu MHz         : 1662.574
cache size      : 2048 KB
physical id     : 0
siblings        : 2
core id         : 0
cpu cores       : 2
fdiv_bug        : no
hlt_bug         : no
f00f_bug        : no
coma_bug        : no
fpu             : yes
fpu_exception   : yes
cpuid level     : 10
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov
pat clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe constant_tsc pni monitor vmx
est tm2 xtpr
bogomips        : 3334.12
clflush size    : 64
```

ma qui vedo un solo core cavolo !!! quindi il mio sistema in realtà si comporta come una cpu a soli 1.66 Ghz ... e che cazzo mica mi sta bene sta cosa con tutto quello che l'ho pagato sto pc ! XD

Informandomi un po in giro, in particolare grazie all'aiuto di Matrix86 che ringrazio veramente tanto per avermi seguito in questa cosa, ho scoperto che di default il mio kernel non supporta il dual core, ne la tecnologia SMP che sfrutta al meglio questo tipo di cpu, tanto per conferma un

```
uname -r
```

mi restituiva

2.6.20-...-NOSMP

a questo punto mi sembrava necessario configurare il kernel e ricompilarlo una volta abilitate queste features .

.: A lavoro !

Come è ovvio, per ricompilare il kernel abbiamo prima bisogno dei SORGENTI del kernel (ma daaaaiii ? XD) sennò che cosa compiliamo la cicoria? :D

Quindi prima di tutto installate i sources della vostra distro .

Una volta fatto questo, posizioniamoci nella dir dei sorgenti

cd /usr/src/linux/

e facciamo partire l'utility grafica di configurazione

sudo make xconfig

(mi raccomando il sudo)

una volta che abbiamo la nostra utility davanti, selezioniamo sul pannello di sinistra la radice "**Processor type and features**" il che farà comparire sul pannello di destra le impostazioni relative al processore .

Assicuriamoci di abilitare la prima voce, ovvero "**Symmetric multi-processing support**" (il famoso SMP) poi, andando alla radice "**Processor family**", cambiamo il tipo cpu abilitando la voce "**Core 2/newer Xeon (MCORE2)**" .

Fatto questo, abbiamo concluso la configurazione dei parametri di compilazione del kernel, quindi salviamo le modifiche e chiudiamo l'applicazione .

Ora siamo pronti per la compilazione vera e propria .

Sempre stando nella dir /usr/src/linux/ diamo il comando (da root o con sudo)

sudo make -j4

il che farà partire il processo che trasformerà i sorgenti del nostro kernel in un immagine avviabile al boot .
Prendetevi un caffè, fumatevi una sigaretta etc etc perchè il processo durerà un po .. diciamo tra i 15 ai 30 minuti in base alla potenza della vostra macchina .

Finita la compilazione, se non ci sono stati errori, installiamo i nuovi moduli con

sudo make modules_install

e aspettiamo che anche questo processo finisca (tranquilli è molto + veloce dell'altro :P) .

bene, se anche questo passaggio è andato a buon fine, è ora di copiare i file compilati nelle cartelle appropriate, quindi

```
sudo cp -vf System.map /boot/System.map-SMP
sudo cp -vf arch/i386/boot/bzImage /boot/vmlinuz-SMP
```

Quasi finito ragazzi, dobbiamo solamente configurare il nostro bootloader (nel mio caso lilo, ma la procedura è + o – la stessa per grub) per caricare il nuovo kernel .

NOTA : Aggiungete il testo che vi sto per dare al file di configurazione DOPO i dati già presenti, senza cancellarli, in modo tale che, se qualcosa dovesse andar male con il nuovo kernel, avrete la possibilità di caricare quello vecchio e utilizzare normalmente il vostro pc .

Editate (da root) il file /etc/lilo.conf aggiungendo alla fine le seguenti righe

```
image = /boot/vmlinuz-SMP
  root = /dev/sda5
  label = Linux-smp
  read-only
```

dove sda5 è la partizione ext3 del vostro disco dove è montata la root directory del sistema .

Salviamo lilo.conf e controlliamo che tutto sia stato impostato correttamente con il comando

```
sudo lilo -t
```

e confermiamo i cambiamenti con

```
sudo lilo -v
```

A questo punto non rimane che riavviare il sistema con un bel

```
sudo reboot
```

e caricare da lilo il nostro kernel “Linux-smp” (o qualsiasi label abbiate specificato nel lilo.conf) .

Vi avviso che alcuni drivers e moduli potrebbero non funzionare con il nuovo kernel, basterà tutta via ricompilarli ed installarti di nuovo per rendere il tutto di nuovo perfettamente funzionante .

Una volta nel sistema verificate la presenza di entrambi i core con i famosi

```
ls /proc/acpi/processor/
```

e

```
cat /proc/cpuinfo
```

Spero il paper vi sia piaciuto e mi raccomando, non smanettate con le impostazioni del kernel che non conoscete, potrebbe rendere il sistema inutilizzabile :P .

evilsocket