

Questo breve tutorial spiega come installare i drivers ATI su distribuzioni Debian e derivate (come Ubuntu).

Premetto che il mio sistema è Debian SID/Unstable, con scheda video ATi Radeon 9600xt 256mb, XFree 4.3 e processore a 32bit. Il processo è stato eseguito con i drivers versione 8.8.25, gli ultimi disponibili al momento della scrittura.

Cosa serve?

L'ultima versione dei drivers ufficiali scaricabili da <http://ati.com/support/driver.html>; Alien e relative dipendenze, che servirà per convertire il pacchetto RPM in DEB.

Nel caso non si abbia alien installato sul proprio sistema, da root eseguire:

```
#apt-get install alien
```

Convertire il pacchetto:

Da root spostarsi nella cartella in cui si è scaricato il pacchetto RPM dei drivers ATi e da riga di comando eseguire:

```
# alien --to-deb nome_pacchetto.rpm
```

sarà così creato un nuovo pacchetto DEB nella stessa cartella.

Installazione pacchetto convertito:

Il pacchetto dei drivers ufficiali ATI installa tra le altre cosa la propria libreria libGL, che va in conflitto con quella già presente nel sistema al momento dell'installazione. Per ovviare al problema sarà necessario usare un parametro aggiuntivo per l'installazione del pacchetto. Da root in una shell posizionata nella stessa cartella del pacchetto DEB appena convertito eseguire:

```
# dpkg -i -force-overwrite
```

siccome nella conversione, si sono persi gli script automatici per la compilazione dei moduli, in seguito alle differenze di automazione tra RPM e DEB, saranno necessarie ulteriori operazioni.

Compilazione moduli DRM per il proprio kernel:

Prima di procedere assicurarsi di avere installato sul proprio sistema il pacchetto con gli headers per il kernel in uso. Questo pacchetto ha una nomanazione del tipo *kernel-headers-x* dove x indica la versione del kernel ed eventuali altre descrizioni del pacchetto (numero build, architettura, etc) del kernel.

Se il vostro sistema è pronto con gli headers installati, spostarsi nella cartella */lib/modules/fglrx* col comando:

```
# cd /lib/modules/fglrx/build_mod
```

ed eseguire:

```
# sh make.sh
```

se l'operazione è andata a buon fine, andare nella cartella padre:

```
# cd ../
```

e procedere con l'installazione del modulo:

```
# sh make_install.sh
```

Lo script lanciato con l'ultimo comando proverà anche a modprobare il modulo appena installato; questo potrebbe fallire se un'altro modulo che usi la scheda video (esempio "radeon", il drivers del progetto DRI fornito con X). Dunque prima di eseguire l'ultimo script consiglio di rimouvere l'eventuale modulo già caricato col comando:

```
# modprobe -r radeon
```

Terminate queste operazioni dopo il modprobe del nuovo modulo fglrx, eseguire da shell:

```
# dmesg
```

che dovrebbe dare un output finale del tipo:

```
fglrx: module license 'Proprietary. (C) 2002 - ATI Technologies, Starnberg, GERMANY' taints kernel.
```

```
[fglrx] Maximum main memory to use for locked dma buffers: 930 MBytes.
```

```
ACPI: PCI interrupt 0000:01:00.0[A] -> GSI 16 (level, low) -> IRQ 209
```

```
[fglrx] module loaded - fglrx 8.8.25 [Jan 14 2005] on minor 0
```

```
allocation failed: out of vmalloc space - use vmalloc=<size> to increase size.
```

```
[fglrx] AGP detected, AgpState = 0x1f00021f (hardware caps of chipset)
```

```
agpgart: Found an AGP 2.0 compliant device at 0000:00:00.0.
```

```
agpgart: XFree86 passes broken AGP3 flags (1f00031e). Fixed.
```

```
agpgart: Putting AGP V3 device at 0000:00:00.0 into 8x mode
```

```
agpgart: SiS delay workaround: giving bridge time to recover.
```

```
agpgart: Putting AGP V3 device at 0000:01:00.0 into 8x mode
```

```
[fglrx] AGP enabled, AgpCommand = 0x1f000312 (selected caps)
```

```
[fglrx] free AGP = 54800384
```

```
[fglrx] max AGP = 54800384
```

```
[fglrx] free LFB = 116387840
```

```
[fglrx] max LFB = 116387840
```

```
[fglrx] free Inv = 134217728
```

```
[fglrx] max Inv = 134217728
```

```
[fglrx] total Inv = 134217728
```

```
[fglrx] total TIM = 0
```

```
[fglrx] total FB = 0
```

```
[fglrx] total AGP = 16384
```

se così è, e non sono presenti messaggi allarmanti, allora tutto è andato per il versio giusto, si può procedere col resto.

Creazione file di configurazione di X:

Eseguire da root il comando:

```
# fglrxconfig
```

seguire le istruzioni, alla fine verrà creato il file /etc/X11/XF86Config-4. Se si aveva un file di

configurazione del tipo XF86Config, rinominarlo come
XF86Config.VECCHIO_CHE_FUNZIA:P, in caso di futuri problemi che non si sa mai...
A questo punto riavviare X e lanciare
fgl_glxgears

un simpatico cubo rotante con altrettanto simpatici ingranaggi ruotanti su ogni sua faccia
dovrebbe allietarci ed essere il preludio al 3D più spinto mai visto su una LinuxBOX! Ora si
può giocare.

Autore: Michele Comignano
Guida pubblicata su: <http://linux.le.it>

--
Vito Russo - Project Manager russo.le.it
<http://russo.le.it>
<http://linux.le.it>
<http://coding.le.it>
<http://windows.le.it>
<http://bsd.le.it>