– General Linux 2 –

Manage/Query Kernel and Kernel Modules at Runtime

(Linux Professional Institute Certification)

`$Id: gl2.105.1.slides.tex,v 1.1 2003/10/25 04:33:06 anguslees Exp $`

Manage/Query kernel and kernel modules at runtim

Objective

Candidates should be able to manage and/or query a kernel and kernel loadable modules. This objective includes using command-line utilities to get information about the currently running kernel and kernel modules. It also includes manually loading and unloading modules as appropriate. It also includes being able to determine when modules can be unloaded and what parameters a module accepts. Candidates should be able to configure the system to load modules by names other than their file name.

Manage/Query Kernel and kernel modules at runtim

Key files, terms, and utilities

`/lib/modules/`*`kernel-version`*`/modules.dep`

`/etc/modules.conf` & **`/etc/conf.modules`**

depmod

insmod

lsmod

rmmod

modinfo

modprobe

uname

Manage/Query Kernel and kernel modules at runtim

Resources of interest

- *linux-source/*Documentation/

Kernel Module Basics

- Kernel modules are chunks of kernel code which can get loaded and unloaded at runtime.
- Some modules depend on code in other modules.
- Good for shipping pre-built kernels – but not loading all available code; good for changing between conflicting modules (eg. OSS and ALSA), etc.

uname – Who the kernel thinks it is

Modules are stored in `/lib/modules/‘uname -r‘/kernel/`

-a print all information

-s kernel name (“Linux”)

-n “node name” (hostname)

-r kernel release (“2.4.20-3-686”)

-v kernel version (compile time)

-m machine hardware name (“i686”)

-o operating system (“GNU/Linux”)

lsmod – Currently loaded modules

lsmod uses `/proc/modules` to show you which kernel modules are currently loaded.

```
Module                  Size  Used by
bsd_comp                5888  1
ppp_async              10624  1
ppp_generic            29072  6 bsd_comp,ppp_async
slhc                    7040  1 ppp_generic
ipv6                  230624  10
ds                     14596  0
pcmcia_core            62688  1 ds
irda                  174400  0
parport_pc             34088  1
lp                     10560  0
parport                40552  2 parport_pc,lp
```

modprobe – Loading modules

Load a module and its dependencies:

`modprobe` *`modulename [module options]`*

eg: `modprobe ftape ft_fdc_base=0x123`

Unload a module if it's unused:

`modprobe -r` *`modulename`*

eg: `modprobe -r ftape`

modinfo – Module options

“modinfo ftape” gives:

```
parm:           ft_fdc_base:Base address of FDC controller.
parm:           ft_fdc_irq:IRQ (interrupt channel) to use.
parm:           ft_fdc_dma:DMA channel to use.
parm:           ft_fdc_threshold:Threshold of the FDC Fifo.
   ⋮
author:         (c) 1993-1996 Bas Laarhoven (bas@vimec.nl), (c) 1995-
1996 Kai Harrekilde-Petersen (khp@dolphinics.no), (c) 1996, 1997 C
Justus Heine (claus@momo.math.rwth-aachen.de)
description: QIC-117 driver for QIC-40/80/3010/3020 floppy tape dr
license:    GPL
```

modprobe maintenance

Configuration in `/etc/modules.conf` (aka `/etc/conf.modules`).

```
# bogus example
options ftape ft_fdc_base=0x123 fg_fdc_irq=5 ft_fdc_dma=1
alias char-major-27 ftape
```

depmod builds `modules.dep`, describing module dependencies.

Loading modules - low-level

Low level commands to actually insert or remove a module:

`insmod` *`filename module options`*

`rmmod` *`modulename`*

You will probably never use these directly.