9.1 Linux Driver Utilities

Command	Function
lsusb	Displays information on all USB devices connected to the computer. This utility uses the following options: **-v** shows exhaustive information. **-s *bus_name*** shows information for a specific bus. **-t** displays the USB device hierarchy as a tree.
hwinfo	Displays information about the hardware on the computer. Be aware of the following options: **--*hardware_item_name*** probes for a specific hardware item. Common hardware names include: ◦ **bluetooth** ◦ **camera** ◦ **cdrom** ◦ **cpu** ◦ **disk** ◦ **dsl** ◦ **monitor** ◦ **mouse** ◦ **keyboard** ◦ **usb** **--short** shows an abbreviated list of information. **--listmd** displays RAID devices. *Not all distributions include the hwinfo command.*
lspci	Displays information for all PCI devices. Be aware of the following options: **-k** shows the kernel drivers that support the device. **-t** displays a tree diagram that shows connections between all busses, bridges, and devices.
lsdev	Gathers information about your computer's installed hardware from the interrupts, ioports, and dma files in the /proc directory. This gives you a quick overview of which hardware uses what I/O addresses and what IRQ and DMA channels. *There are no options for this utility.*

9.2 Kernel Module Locations

File	Description
/etc/modprobe.conf	Provides the modprobe utility with default commands for loading modules at boot time. Entries in the file include the following: • **install** loads a module at boot time. • **alias** specifies a name as an alias for a module name. This alias can be used with module utilities. • options specifies options used while loading a module, including: ◦ **irq** for IRQ information ◦ **io** for I/O port information.
/etc/modprobe.d	Contains multiple configuration files used by modprobe at boot time if the /etc/modprobe.conf file does not exist.
/usr/lib/modules/kernel-version	This directory stores your kernel modules which are available to all users.

9.2 Kernel Module Management

Command	Function
lsmod	Lists all loaded modules. The command pulls information from the /proc/modules file.
cat /proc/modules	Views the /proc/modules file. This file contains a list of all loaded modules.
modinfo	Views additional information about a module listed using the lsmod command.
depmod	Creates a file that identifies module dependencies. The file is placed at /lib/modules/kernel_version_number/modules.dep. This command first reads the /etc/modules.conf file to identify modules. It then probes each module to build a list of dependencies. Be aware of the following options: **-a** shows information for all modules. **-n** shows what would happen on the screen, but does not perform the action. **-v** uses verbose mode.
insmod	Installs a module. • This command does not look for dependencies. It will fail if a module being loaded has unresolved dependencies. • Include the full name of the module, including the .o or .ko extension.
modprobe	Loads a module along with any module dependencies. This utility also runs at startup to load modules. The /etc/modprobe.conf file provides modprobe with its configuration information. Be aware of the following options: **-l** lists all loaded modules. **-r** removes a module. This option checks for dependencies before unloading the module.
rmmod	Removes a module from the kernel. rmmod: • Cannot unload the module if it is in use. • Does not look for dependencies and can cause errors if a module depends on a module that is unloaded.

9.3 Components to Manage Devices

Component	Description
sysfs	The Linux kernel provides a virtual file system called sysfs which is mounted at /sys. sysfs is able to export information about hotplug devices so that other utilities can access the information.
Hardware Abstraction Layer (HAL) daemon	The HAL daemon (hald) provides all running processes with data about current hardware. Hald runs constantly.
Desktop Bus (D-Bus) daemon	The D-Bus daemon notifies running processes whenever a hotplug device is connected or disconnected from the system.
udev	The udev system is comprised of a few kernel services and the udevd daemon. The kernel informs the udevd daemon when certain events happen and the udevd daemon is configured to respond to events with corresponding actions. The event information comes from the kernel and the actions happen in userspace. The responses to the events are configurable in "rules". A udev rule can specify what name will be given to a device regardless of which port the device has been placed. This consistent naming of devices guarantees that scripts dependent on a specific device's existence will not be broken. **For example**, the udev daemon (udevd) creates a virtual file system that is mounted at /dev. It communicates with the Linux kernel through the uevent interface. When a hotplug device is added or removed, the kernel sends out a uevent message that is picked up by udevd. Based on the rules defined in the files in the /etc/udev/rules.d directory, udevd then: • Initializes the device. • Creates the appropriate device file in the /dev directory. • Configures the device using the ifup utility if the new device is a network interface. • Mounts the device using the information in /etc/fstab if the new device is a storage device. • Informs running processes about the new device.
udevadm	udevadm is the udev management tool. It expects a command and command specific options. It also controls the runtime behavior of udev, requests kernel events, manages the event queue, and provides simple debugging mechanisms. • **trigger [options]** Request device events from the kernel. This is primarily used to replay events at system coldplug time. This can also be used to add devices. • **control [options]** Gives you the ability to control the udev daemon. For example, rules are not re-triggered automatically on already existing devices. Hot-pluggable devices, such as USB devices, will probably have to be reconnected for the new rules to take effect.