

9.3.2 Hotplug and Coldplug Device Facts

This lesson covers the following topics:

- Device categories
- Components to manage devices

Device Categories

Be aware of the following device categories when managing hardware:

Category	Description
Coldplug	<p>Devices that are connected prior to system boot are said to be “coldplugged.” Coldplug devices should only be removed or replaced when the power to the computer is off. Attempting to remove these devices while the power is on can damage the device or the computer.</p> <p>Coldplug devices include:</p> <ul style="list-style-type: none"> ▪ RAM (Random Access Memory) chips ▪ CPU (Central Processing Unit) ▪ Expansion cards, such as Peripheral Component Interconnect (PCI) or PCI Express cards ▪ Standard hard disk drives
Hotplug	<p>Hotplug devices can be removed while the computer is on. Linux uses software designed to detect these changes as the devices are added and removed.</p> <p>Hotplug devices include:</p> <ul style="list-style-type: none"> ▪ USB devices ▪ FireWire devices ▪ Hot-swappable hard disk drives

Components to Manage Devices

Linux uses the following 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	<p>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.</p> <p>For example, the udev daemon (udev) 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</p> <p>Based on the rules defined in the files in the /etc/udev/rules.d directory, udevd then:</p> <ul style="list-style-type: none"> ▪ 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.

Commands	Description	Examples
trigger <i>[options]</i>	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.	udevadm trigger --action=add --subsystem-match=pci This adds the pci subsystem udevadm trigger --type=devices This command reapplies the rules to a device that is already connected,
control <i>[options]</i>	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.	udevadm control --reload-rules This reloads the rules files.