4.4.7 SysVinit Facts

When Linux is powered on, it must first determine which process to start, the order in which they should be started, and what to do when a process is terminated. For years, many Linux distributions used a traditional SysVinit-based operating system to manage this initialization process. Newer distributions are now migrating to systemd, which is a replacement daemon designed to start processes in parallel. This lesson focuses on the traditional SysVinit init process.

This lesson covers the following topics:

- Init script configuration
- Init script commands

Init Script Configuration

Init is the program on Linux systems that spawns all other processes. It runs as a daemon and typically has a process identification number (PID) of 1. It is the parent of all processes. Its primary role is to create processes from a script stored in the file /etc/inittab file. These scripts specify whether specific daemons start at a specified runlevel.

Init scripts:

- Are stored in the following locations:
  - **/etc/rc.d/init.d**
  - **/etc/init.d**
- Have symbolic links that are stored in subdirectories that correspond to the runlevel under which each script should start..
- Can be started and stopped manually.
- Are started at boot using the init script. This code includes:
  - The default-start line defines the runlevels in which the script starts by default.
  - The required-start line defines services that must be running before this service can start.
  - The should-start line defines the services that are recommended to start before this service starts.

Init script directories also contain other important scripts.

- **rc** starts and stops specified daemons.
- **halt** stops or reboots the computer. It runs when the init 0 or init 6 commands are invoked.
- **bootrc** or **sysinit** is run by the init process when a computer starts.
  These scripts perform tasks that include:
  - Loading the kernel module
  - Checking the file system
- **boot.local** or **rc.local** runs specific tasks at startup as specified by the administrator.

When using init scripts, keep the following in mind:

- At boot time, init uses the **/etc/inittab** file to determine the default runlevel, such as runlevel 5.
- When a runlevel is specified, init looks at the directory associated with the runlevel to determine which processes to start. The directory for runlevel 5 is named rc5.d. Additional directories named rc0.d through rc6.d specify which processes to start for each runlevel.
- Each rc directory contains symbolic links that point to a specific init script.
  - Link names starting with an **S** start a script for the runlevel.
  - Link names starting with a **K** kill a running process when the computer changes runlevels.
- Init follows the links and runs the scripts to start or stop processes.

Init Script Commands

Use the following commands to manage daemons and the init scripts.

Command	Function	Examples
service *daemon_name*	Manages the current state of a daemon. Options include: - **start** starts a daemon that is not currently running. - **stop** halts a running daemon. - **restart** stops and restarts a daemon. - **reload** requests that a daemon read and apply its configuration files without stopping. - **status** shows the status of a single daemon or daemons. - **--status-all** shows the status of all daemons.	**service atd start** starts the atd daemon. **/etc/rc.d/init.d/httpd start** starts the httpd daemon. **service httpd stop** halts the httpd daemon. **/etc/rc.d/init.d/httpd restart** restarts the httpd daemon. **/etc/init.d/httpd reload** implements a new configuration for the httpd daemon without

	As an alternative method, use the absolute path to the daemon script and the option to configure the daemon (e.g., **/etc/rc.d/init.d/httpd stop**).	halting the service. **service httpd reload** reloads the httpd daemon. **/etc/init.d/httpd status** shows whether the httpd daemon is running. **service --status-alll**
insserv	Configures default runlevels for a daemon on a BSD distribution. **insserv** references the INIT INFO script section of each daemon to determine the default runlevels for the daemon and dependent daemons. Be aware of the following options: - *script_name* starts at the runlevels specified in the init block script code. - **-r** keeps a script from starting at any runlevel. - **-d** restores a daemon to the default runlevels defined in the scripts.	**insserv httpd** causes the httpd daemon to start at the runlevels specified in the script. **insserv -r httpd** stops the httpd daemon from starting when a computer boots.
chkconfig	Configures default runlevels for a daemon. Be aware of the following options on a System V distribution: - **--add** adds a new service to be managed by **chkconfig**, and makes sure the service has a start or kill entry at every runlevel. - **--del** removes a service from **chkconfig** management and removes symbolic links to the service from /etc/rc*0-6*.d. - **--level** *s*pecifies the level to which a service should belong. - **--level on**|**off**|**reset** starts, stops, or resets the named service in the specified runlevel. The **on** and **off** options affect levels 2, 3, 4, and 5 when the runlevel is omitted. - **--list** lists services and their runlevels. Be aware of the following options on a BSD distribution: - **-l** lists services and their runlevels. - **-s** *s*pecifies the level to which a service should belong.	**chkconfig --add atd** starts the atd daemon. **chkconfig --del ldap** removes the ldap daemon. **chkconfig --level 5 lpd** specifies level 5 for the lpd daemon. **chkconfig --level 345 nfslock off** turns the nfslock daemon off in runlevels 3, 4, and 5. **chkconfig ypxfrd on** turns the yp transfer daemon on in levels 2, 3, 4, 5.