Linux Academy

Study Guide

Linux Academy RHCSA 7 Prep

Contents

Basic Commands

- **`pwd`** • Show current working directory path
- **`cd`** • Change directory
- **`ls`** • List contents of directory
- **`sudo`** • Allows a super user to run a command with root priviledges
- **`mkdir`** • Create new directory
  - » **`-p`** • Create parent directories, if do not already exist
- **`rmdir`** • Remove directory
- **`rm -rf`** • Force remove a directory, recursively (includes all files inside)
- **`touch`** • Create new, empty files

Input-Output Redirection

- **`>`** • Redirect standard output to file
  - » `echo "test" > file.txt`
  - » Replaces file, if already exists
- **`>>`** • Redirects and appends standard output
  - » `echo "test" >> file.txt`
  - » Adds text to bottom of file
- **`|`** • Chain scripts, files and commands together by the STDOUT as STDIN for the next command
  - » `cat /etc/passwd | grep root`
- **`2>`** • Redirect standard error
- **`2>>`** • Redirect and append standard error
- **`/dev/null`** • Data sent to */dev/null* is lost
- **`2>&1`** • Redirect STDEDD to STDOUT
- **`<`** • Accept input from file
  - » `mysql < filedump.sql`
- **`less`** • File viewing application and STDOUT can often piped into for ease of reading

- **`head`** • Show first ten lines of file
  - » **`-n`** • Define number of lines
- **`tail`** • Show last ten lines of file
  - » **`-n`** • Define number of lines

File System Hierarchy Standard

- **`/etc`** • Contains configuration files for programs and packages
- **`/var`** • Variable data specific to system. This data should not be removed or changed when the system reboots. Logs files tend to be stored within the `/var` directory
- **`/run`** • Runtime data for processes since last boot
- **`/home`** • Location of home directories; used for storing personal documents and information on the system
- **`/root`** • *root* user home directory
- **`/tmp`** • Files are removed after ten days; universal read/write permissions
- **`/boot`** • Files needed to start the system boot process
- **`/dev`** • Contains information on essential devices

Grep and Regular Expressions

- **`grep`** • Prints lines that match defined pattern
  - » **`grep pattern file.txt`**
  - » **`-i`** • Case insensative
  - » **`-v`** • Shows lines *not* containing pattern
- Examples including regex:
  - » **`grep linuxacademy filename`** • Search for *linuxacademy* in *filename*
  - » **`grep "^linuxacademy" filename`** • Search for lines starting with *linuxacademy*
  - » **`grep "linuxacademy$" filename`** • Search for lines ending with *linuxacademy*
  - » **`grep "^[abd]" filename`** • Search for characters not contained in brackets
  - » **`grep [lL]inuxacademy filename`** • Search for pattern starting with either capital or lowercase *L*

  - » **`grep "^$" filename`** • Search for empty lines
  - » **`grep -v ^# filename`** • Search for uncommented lines
- **`egrep`** • Same as `grep`, but using extended regular expressions
- **`fgrep`** • Interpret pattern as list of fixed strings

Access Remote Systems Using SSH

- **Password authentication** • Allows user to log in with only a password; considered to be less secure than using key-based authentication
- **`ssh user@server`** • Connect to remote host
- **`ssh server command`** • Issue command on remote host without connecting
- **`scp filename user@server:~/`** • Secure copy file to server
- **`sftp user@server`** • Secure File Transfer Protocol
  - » **`?`** • Display all options
  - » **`ls`** • List files
  - » **`cd`** • Mode directories
  - » **`get`** • Download
  - » **`quit`** • Exit `sftp`

Log In and Switch Users in Multi-User Targets

- **Target** • Systemd configuration files used for grouping resources
- **Interactive shell** • Any shell that has a prompt for user interaction
- **`su`** • Log in as another user
  - » **`su user`** • Log in to an interactive, non-login shell
  - » **`su - user`** • Log in to a login shell
- **GNU Bourne-Again Shell** • **Bash**
  - » Interactive shell uses either `$` (user) or `#` (root) prompt
  - » Takes commands, which run programs
    - Made up of three parts:

    - Command name
    - Options or *flags* to pass into the command
    - Arguments

Archive and Compress Using tar, star, gzip and bzip2

- **`tar`** • Archive files; does not handle compression
  - » **`-c`** • Create new archive
  - » **`-t`** • List contents of archive
  - » **`-x`** • Extract files from archive
  - » **`-z`** • Compress or uncompress file in `gzip`
  - » **`-v`** • Verbose
  - » **`-j`** • Compress or uncompress file in `bzip2`
  - » **`-f`** • Read archive from or to file
  - » Examples
    - **`tar -cf helloworld.tar hello world`** • Archive *hello* and *world* files into *helloworld.tar* archive
    - **`tar -tvf helloworld.tar`** • List all files in *helloworld.tar* archive
    - **`tar -xf helloworld.tar`** • Extract files in archive
    - **`tar -czvf helloworld.tar.gz hello world`** • Archive and compress (using `gzip`) *hello* and *world* files into *helloworld.tar.gz* archive
    - **`tar -zxvf helloworld.tar.gz`** • Uncompress (in `gzip`) and extract files from archive
- **`star`** • Archiving utility generally used to archive large sets of data; includes pattern-matching and searching
  - » **`-c`** • Create archive file
  - » **`-v`** • Verbose output
  - » **`-n`** • Show results of running command, without executing the actions
  - » **`-t`** • List contents of file

  - » **`-x`** • Extract file
  - » **`--diff`** • Show difference between files
  - » **`-C`** • Change to specified directory
  - » **`-f`** • Specify file name
  - » Examples"
    - **`star -c f=archive.tar file1 file2`** • Archive *file1* and *file2* into *archive.tar* archive
    - **`star -c -C /home/user/ -f=archive.tar file1 file2`** • Move to */home/user* and archive *file1* and *file2* from that directory into *archive.tar*
    - **`star -x -f=archive.tar`** • Extract *archive.tar*
    - **`star -t -f=archive.tar`** • List contents of *archive.tar*
- **`gzip`** • Compression utility used to reduce file sized; files are unavailable until unpacked; generally used with `tar`
  - » **`-d`** • Decompress files
  - » **`-l`** • List compression information
  - » Examples:
    - **`gzip file1`** • Compress *file1* into *file1.gz*
    - **`gzip -d file1.gz`** • Unpack *file1*
    - **`gunzip filename`** • Unpack *filename*

Create and Edit Files

- **`vi`** • Text editor that is always installed and useable; replaced `vim`
- **`vim`** • Vi iMproved; full-featured version of `vi`
- **`nano`** • Simple text editor
- **`touch`** • Create empty file

Create, Delete, Copy and Move Files and Directories

- **`mkdir`** • Make directory

  - » **`-p`** • Create parent directories, if not already created
- **`cp`** • Copy files and directories
  - » **`-R`** • Copy directory recursively
- **`mv`** • Move files and directories
- **`rm`** • Remove files and directories
  - » **`-r`**/**`-R`** • Remove recursively
  - » **`-f`** • Force remove
  - » **`-i`** • Prompt before removal

Create Hard and Soft Links

- **`ln`** • Create links between files
  - » Without the `-s` flag, creates a hard link
  - » **`-s`** • Symlink files
- **symlinks** • Soft links that connects one file to another, symbolically; if the target file moves to changes, the symlink continues to try use the previous location and must be updated
- **Hard link** • Links directly to an inode to create a new entry referencing an existing file on the system

List, Set and Change Standard Permissions

- Two ways to define permissions on a standard Linux system:
  - » Using symbolic characters, such as *u*, *g*, *o*, *r*, *w* and *x*
  - » Using octal bits
  - » The RHCSA only requires knowledge of the symbolic
- **`chmod`** • Change mode; set the permissions for a file or directory
  - » **`u`** • User
  - » **`g`** • Group
  - » **`o`** • Other
  - » **`a`** • All
  - » **`r`** • Read

    - **w** • Write
    - **x** • Execute
    - **s** • Set UID or GID
    - **t** • Set sticky bit
    - **-X** • Indicate the execute permissions should only affect directories and not regular files
    - Octal bits:
        - 1 • Execute
        - 2 • Write
        - 4 • Read
- **chown** • Change owner and group permissions
    - `chown user:group filename`
    - **-R** • Set ownership recursively
- **chgrp** • Change group ownership
- **setuid** • Set user ID permissions on executable file
- **setgid** • Set group ID permissions on executable file
- **umask** • Set default permissions for new directories and files

Locate, Read and Use System Documentation

- **command --help**
- **info** • Read information files; provides more information than `man`
- **which** • Show full path of command; useful for scripting
- **whatis** • Display manual page descriptions
- **locate** • Locate files on system by name
- **updatedb** • Update `locate` command databases
- **man** • Documentation
    - Nine sections:
        - **1** • Executable programs and shell commands
        - **2** • System calls

- **3** • Library calls
- **4** • Special files
- **5** • File formats
- **6** • Games
- **7** • Miscellaneous
- **8** • root user commands
- **9** • Kernel routines

- **`apropos`** • Search man pages and descriptions for text

Boot, Reboot and Shut Down a System

- Reboot:
  - `reboot`
  - `systemctl reboot`
  - `shutdown -r now`
- Shutdown:
  - No power off
  - `systemctl halt`
  - `halt`
  - `shutdown -h now`
  - `init 0`
- Power off:
  - `systemctl poweroff`
  - `poweroff`
  - `shutdown -P`

Boot Into Different Targets Manually

- A **target** is a Systemd unit of configuration that defines a grouping of services and configuration files the must be started when the system moves into the defined target.
  - A grouping of dependencies starts when a target is called

- **`systemctl list-units --type=target`** • View all targets on system
- **`systemctl list-units --type=target --all`** • View all targets on disk
- Common targets:
  - » **emergency.target** • `su` login; mounts only the root filesystem, which is read-only
  - » **multi-user.target** • Support concurrent log ins of multiple users
  - » **rescue.target** • `su` login; basic Systemd init
  - » **graphical.target** • Support concurrent log ins of multiple users on a graphical interface
- **`systemctl get-default`** • Show default target
- **`systemctl set-default`** • Set default target
- Configuration files:
  - » */usr/lib/systemd/system*
  - » */etc/systemd/system*
- **`systemctl -t help`** • View unit configuration types
- **`systemctl status service`** • Find status of service
- **`systemctl --type=service`** • List configuration files of active services
- **`systemctl enable service`** • Enable service configuration to start at boot
- **`systemctl --failed`** • List failed services
- Select a different target at boot:
  - » Reboot system
  - » At Grub menu, press **E** to edit entry
  - » Go to *linux16* kernel and press **CTRL+E**
  - » Add `systemd.unit=target.target`
  - » **CTRL+X**

Interrupt Boot Process to Access System

- Start or reboot system
- Stop Grub autoselection
- Ensure the appropriate kernel is highlighted and press **E** to edit

- Navigate to the *linux16* line, press **E**
- Add line `rd.break`
- **CTRL+X**
- System boots into emergency mode
- Mount */sysroot* with read and write permissions
  - » **`mount -oremount, rw /sysroot`**
- Switch into chroot jail:
  - » `chroot /sysroot`
- Reset root password
- Clean up
  - » **`touch /.autorelabel`**
- **`exit`**
- **`exit`**

Identify CPU/Memory Intensive Processes, Adjust Priority, Kill Processes

- **`top`**
  - » **`k`** • Kill process
  - » **`q`** • Quit
  - » **`r`** • Renice
  - » **`s`** • Change update rate
  - » **`P`** • Sort by CPU usage
  - » **`M`** • Sort by memory usage
  - » **`l`** • Toggle load average
  - » **`t`** • Toggle task display
  - » **`m`** • Toggle memory display
  - » **`B`** • Bold display
  - » **`u`** • Filter by username

  - » **`-b`** • Start in batch mode
  - » **`-n`** • Number of updates before exiting
  - » Columns:
    - **PID** • Process ID
    - **USER**
    - **PR** • Priority
    - **RES** • Non-swap memory
    - **SHR** • Shared memory size
    - **%CPU** • Task's share of elapsed CPU time
    - **%MEM** • Current amount of used memory
    - **TIME+** • CPU time minus the total CPU time the task has used since starting
- Nice priority:
  - » **-20** • Highest priority
  - » **19** • Lowest priority
  - » Any user can make a task lower priority
- **`pgrep`** • Search processes
  - » **`-u`** • Username
  - » **`-l`** • Display process name
  - » **`-t`** • Define tty ID
  - » **`-n`** • Sort by newest
- **`pkill`** • Kill process
  - » **`-u`** • Kill process for defined user
  - » **`-t`** • Kill process for defined terminal
- Kill signals:
  - » **1** • SIGHUP • Configure reload without termination; also used to report termination of controlling process
  - » **2** • SIGINT • Cause program to terminate
  - » **3** • SIGQUIT • When user requests to `quit` a process

  - **9** • SIGKILL • Immediately terminate process
  - **15** • SIGTERM • Send request to terminate process; request can be interpreted or ignored
  - **18** • SIGCONT • Restart previously stopped process
  - **19** • SIGSTOP • Stop a process for later resumption
  - **20** • SIGTSTP • Send by terminal to request a temporary stop
- **`ps`** • Process status

Locate and Interpret System Log Files and Journals

- **`journald`** • Responsible for event logging; records events from log files, kernel messages, etc.
  - Data does not persist after reboot
  - Can be configured for persistence in */etc/journald.conf*
  - Temporary log location: */run/log/journal*
  - Persistent log location: */var/log/journal*
- **`journalctl`**
  - **`-n`** • Set number of lines to show
  - **`-x`** • Provide explanation text, if available
  - **`-f`** • Show last ten events; continues listening
  - **`-b`** • Show messages from current boot only
  - **`-p`** • Show message priority type
  - **`_SYSTEM_UNIT=service`** • Get events related to service
  - **`--since=yesterday`** • Get events since defined time
  - **`--until=00:00:00`** • Get event from before defined time
- Find information about system boot:
  - **`systemd-analyze`**
  - **`systemd-analyze blame`**

List, Create and Delete Partitions

- **`fdisk`** • Used to create master boot record-based partitions
- **`gdisk`** • Used to create GPT-based partitions

Create and Remove Physical Volumes, Logical Volumes

- **Physical volume** • The physical disk or disks; can be a partition or whole volume
- **Volume group** • A combination of physical volumes that work as a logical volume, with pooled space

LVM Set Up

- **`pvcreate`** • Create physical volume
- **`pvdisplay`** • Show available physical volumes
- **`vgcreate name /dev/disks`** • Create volume group
- **`vgdisplay`** • Show available volume groups
- **`lvcreate`** • Create logical volume
  - » **`-n`** • Volume
  - » **`-L`** • Size in bytes
- **`lvremove /dev/vg/volume`** • Remove volume
- **`pvremove /dev/disk`** • Remove physical volume

Configure System to Mount File System at Boot

- **`mkfs -t xfs /dev/xvdf1`** • Make file system
- **`blkid`** • List available block devices on system
- **`lsblk`** • List all attached block devices
- **`mount /dev/disk /mnt/mountlocation`** • Non-persistent mount
  - » Mounting with the UUID ensures the appropriate mount is used
  - » Add to */etc/fstab* to mount persistently
- **`tune2fs -L labelname /dev/disk`** • Mount with file system label (ext)
- **`e2label /dev/disk labelname`** • Mount with file system label (ext)

- **`xfs_admin -L labelname /dev/disk`** • Mount with file system label (XFS)
- **`mount LABEL=labelname /mnt/mountlocation defaults 1 1`** • Mount with label, non-persistent; edit */etc/fstab* for persistence
- **`mount -a`** • Mount all file systems in */etc/fstab*
- **`umount -a`** • Unmount all file systems in */etc/fstab*

Schedule Tasks Using at and cron

- **`at`** • Execute command at a later time
  - » ***/etc/at.allow*** • Configure users permitted to use `at` command
  - » ***/etc/at.deny*** • Configure users **not** permitted to use `at` command
  - » Accepts following time/date formats:
    - hh:mm
    - midnight
    - noon
    - teatime (16:00)
    - am/pm
    - Full dates
    - now + time
- **`atrm`** • Remove pending `at` task
- **`anacron`** • Execute commands periodically
  - » **`-f`** • Force execution, ignoring timestamps
  - » **`-u`** • Upload timestamps of all jobs; does not run jobs
  - » **`-n`** • Run jobs immediately, ignoring delays
  - » **`-t`** • Use specified configuration file, instead of default
  - » **`-h`** • Show help
  - » ***/etc/anacrontab*** • Configuration file
  - » ***/var/spool/anacron*** • Shows all timestamps for jobs
  - » Only root and superusers can use `acacron`

  - Syntax:
    - **period in days** • Frequency of execution
    - **delay in minutes** • Number of minutes to wait before job execution
    - **job-identifier** • Unique name of job used in log files
    - **command** • Command to execute
    - **start_hours_range** • Time frame when jobs can be run
    - **random_day** • Stagger job starts at random times

Configure System to Use Time Services

- **`timedatectl list-timezones`** • List all available time zones
- **`tzselect`** • Select appropriate time zone
- **`timedatectl set-timezone zone/location`** • Set time zone
- **`timedatectl set-time YYYY-MM-DD hh:mm:ss`** • Set time and date
- **`timedatectl set-ntp true`** • Use Network Time Protocol
- NTP can be managed by either `ntpd` or `chronyd`
  - Generally, `ntpd` is for servers, and `chronyd` is for systems with frequent restarts
  - **`chronyd`** is the default for RHEL7

Install and Update Software Packages

- **`yum`** • Package management tool
  - **`install packagename`** • Install package
  - **`search string`** • Search packages
  - **`search all string`** • Searches name, description and summary
  - **`list`** • List installed packages
  - **`list all`** • Listed installed and available packages
  - **`list installed`** • List installed and available packages
  - **`check-update`** • Lists packages with available updates
  - **`update packagename`** • Update defined package

  - **`update`** • Update all packages with available updates
  - **`info package`** • Provide information about package
  - **`provides /some/directory`** • Displays packages that match path
  - **`list kernel`** • List installed and available kernels
  - **`remove packagename`** • Removes defined package
  - **`history`** • Display summary of installations and removes
  - **`history undo idnumber`** • Reverse a transaction
  - Working with groups (packages of software):
    - **`yum grouplist`** • Show available groups to install
    - **`grouplist hidden`** • Show all available groups
    - **`groupinstall groupname`** • Install defined group
    - **`groupinfo groupname`** • Display all packages to be installed with the group
      - **`-`** • Package is not installed and will not be installed
      - **`=`** • Package is installed as part of group
      - **`+`** • Package is not installed, but will be installed at next update
      - No symbol means that the package is installed, but was not installed as part of the group
  - ***/var/log/yum*** • Log file

Enable Third-Party Repositories

- **`yum repolist`** • List repository ID, name and number of packages available
  - **`-v`** • List more information about repos
  - **`all`** • Show all repos
- **`yum repoinfo`** • Show information about both enabled and disabled repos
- ***/etc/yum.repos.d/reponame.repo*** • Location of repositories
- **`yum-config-manager`** • Set repositories
  - **`--enable reponame`** • Enable repo
  - **`--disable reponame`** • Disable repo
  - **`--add-repo repourl`** • Add repository from defined URL

RPM

- RPM Package Manager
- Always use `yum` when possible
- **`rpm`**
  - **`-i`** • Install
  - **`-v`** • Verbose
  - **`-e`** • Remove package
  - **`-h`** • Use hashmarks for progress
  - **`-U`** • Upgrade to install package
  - **`-F`** • Upgrade already-installed package
  - **`-q`** • Query for a package
  - **`-a`** • Display all packages
  - **`-qa`** • Display installed files
  - **`-ql`** • List files in installed package
  - **`-qd`** • List documentation for package
  - **`-qpl`** • List files in RPM package

Create, Delete and Modify Local User Accounts

- **`id`** • Print user and group IDs
- UID ranges:
  - **0** • root
  - **1-200** • System users for Red Hat processes
  - **201-999** • System users for processed that do not own files
  - **1000+** • Regular users
- ***/etc/passwd*** • User login and password information
- ***/etc/shadow*** • User login and password hash information
- **Primary group** • The main group for a user; all files created by a user are set under this group

- ***/etc/groups*** • Group member information
- **`getent group username`** • Show all groups for a user
- **`useradd`** • Create user
- **`usermod`** • Modify user
- **`userdel`** • Delete user

Change Password and Password Aging

- **`chage`** • Modify amount of days between password changes
  - » **`-d`** • Number of days since 1970-01-01 to define password change
  - » **`-E`** • Set password expiration date
  - » **`-I`** • Number of days of inactivity before password expiration
  - » **`-l`** • Show account aging information
  - » **`-m`** • Minimum number of days between password changes
  - » **`-M`** • Maximum number of days between password changes
  - » **`-W`** • Days of warning before password change

Create, Delete and Modify Groups

- **`groupadd`** • Add a group
  - » **`-g`** • Group ID
  - » **`-r`** • Create system group
- **`groupmod`** • Modify group
  - » **`-g`** • New group ID
  - » **`-n`** • New group name
- **`groupdel`** • Delete group
- **`chmod g+s directoryname`** • Set group permissions for directory, and all files created in that directory have the same permissions

Create, Mount, Unmount and Use VFAT, EXT4 and XFS File Systems

- **VFAT** • Extension of FAT file system, allowing log file names; often used in SAMBA shares or when sharing files between Linux and Windows computers
  - » **`mkfs.ext /dev/xvdf1`** • Create VFAT file system at location
  - » **`mount /dev/xvdf1 /mnt/location`** • Mount file system
  - » **`fsck.vfat /dev/xvdf1`** • Check for file system consistency
- **EXT4** • Common among Linux systems; journaling-based file system that can support up to 16TBs on Red Hat and up to 50TB in file system size
  - » **`mkfs.ext4 /dev/xvdf1`** • Create EXT4 file system on device
  - » **`mount /dev/xvdf1 /mnt/location`** • Mount the file system at location
  - » **`fsck /dev/xvdf1`** • Check for file system consistency
  - » **`dumpe2fs /dev/xvdf1`** • Get details of file system
  - » **`tune2fs /L labelname /dev/xvdf1`** • Label the device
- **XFS** • Known for parallel processing and high I/O throughput; journaled file system that supports up to 500TB file size on Red Hat 7 with 500TB in file system size
  - » **`mkfs.xfs /dev/xvdf1`** • Create XFS file system on device
  - » **`mount /dev/xvdf1 /mnt/location`** • Mount file system at location
  - » **`xfs_repair /dev/xvdf1`** • Check for file system consistency
  - » **`xfs_info /dev/xvdf1`** • Get details of file system
  - » **`xfs_admin /L labelname /dev/xdf1`** • Label the device