Linux Academy

Study Guide

LPIC-2

Contents

Prerequisites

- Linux - CentOS 6 (or other Sysvinit distribution)

Topic 200 - Capacity Planning

200.1 - Measure and Troubleshoot Resource Usage (System Availability - uptime)

- `uptime`
  - Shows how long a system has been running, can be formatted to display in more "human readable" format
  - `-V`, `--version` • Version information on the utility
  - Information on uptime container in `/var/run/utmp`, binary file

200.1 - Measure and Troubleshoot Resource Usage (CPU and Disk - iostat)

- `iostat`
  - Provides information on both CPU and memory utilization since the last system boot
  - `-c` • Display CPU utilization report
  - `-h` • Add formatting that may be easier for humans to read
  - `iostat -c 2 5` • Provides CPU information for the user, system, iowait and idle every two seconds, repeated five times
  - `-d -` • Display the device utilization report
    - Shows the detected block devices (CDROM, SCSI, SATA, IDE, LVM, etc) and their disk statistics
    - Statistics includes kB read per/s, written per/s, kB read total, kB written total

200.1 - Measure and Troubleshoot Resource Usage (CPU and Disk - sar)

- `sar`
  - Provides similar information to `iostat`, but displays it as it occurs over time (generally 10

minute intervals)

- `sar | head` • Shows CPU utilization for user, system and idle (in sum for all CPUs) every default time period
- `-d` • Shows disk utilization for detected block devices (CDROM, SCSI, SATA, IDE, LVM, etc) in TOTAL
    - Displays the device, transfers per second, reads per/s, writes per/s, average request size (in sectors), average length of the queue, average time device has taken to serve request (await) and average service time for request (svctime)
- Part of the "sysstat" package on CentOS 6 and most other sysvinit distributions, may not be installed by default
- Default log files written to are in `/var/log/sa` and each file is *sa#*, where the # is the day number of the report
- First time, must be enabled and started to create the log directory and first file
    - `chkconfig sysstat on`
    - `service sysstat start`

200.1 - Measure and Troubleshoot Resource Usage (Memory - free)

- `free`
    - Displays a summary of the virtual memory (including physical RAM and swap) utilization
    - Statistics displayed are total, used, free, shared, buffered/cached and available memory
    - `-m` • Display stats in megabytes
    - `-g` • Display stats in gigabytes
    - `-h` • Tries to shorten the displayed numbers for readability
    - `-s ##` • Polls ## seconds for display

200.1 - Measure and Troubleshoot Resource Usage (Memory - vmstat)

- `vmstat`
    - Provides more detail on memory than the `free` utility
    - Values include amount of virtual memory used, idle memory, buffered memory use, cached memory use, inactive memory, active memory, swap in/out, blocks in/out, interrupts per/s, context

switches per/s

  - `-a` • Display active and inactive memory totals

  - `#` • Delay in seconds between output of values

  - `-t` • Add a timestamp to any output of values

  - `-s` • Display a single table summary of all memory statistics (cannot be repeated with intervals)

200.1 - Measure and Troubleshoot Resource Usage (Disk and Files - lsof)

- `lsof`

  - Typically, used to list open files

  - Can be used to provide some disk I/O information when applying some filters as shown

  - `lsof | head -10` • Provides the top ten files that are open

  - `lsof | wc -l` • Provides a count of the total number of open files on run

  - `-u [username]` • Display open file statistics for that particular user

    - Statistics displayed include the command run, the PID, the terminal ID, the username, the file descriptor, type of file, the device, the size and the node name (directory)

    - `ls -l /dev | grep "###, +"` • Displays all devices matching that major device number, pass the full device ID from the command to get one specific device

200.1 - Measure and Troubleshoot Resource Usage (Process Management - ps, pstree, top)

- `ps`

  - Reports a snapshot of the current processes on a system for the current user (or, as root, the entire system)

  - Statistics displayed include owner of the process, the PID, the parent PID, CPU, start time of process, TTY/Terminal ID, total CPU time over life of process, full command name

  - `-e` • Select and display all processes

  - `-f` • Full format listing, is typically combined with other switches (like `-e`) to display specific processes in more detail

- `pstree`

    - Reports a snapshot of the current process tree on a system for the current user (or, as root, the entire system)
    - Shows indented and ascii-formatted graphical connections between parent and child processes in a tree
    - `-a` • Include command line arguments passed to a process
    - `-A` • Use ascii characters for the tree
    - `-G` • Use VT100 line drawing characters (if possible), makes the display and connections visually clearer
    - `-h` • Highlight the current process
    - `-p` • Include PIDs
    - `-Z` • Include SELinux security context for each process (if SELinux enabled)
- `top`
    - Displays a constantly updated list of tasks/processes on a system
    - `top` is interactive, keyboard shortcuts change/update the values displayed in real time
        - **l** • Load average
        - **t** • Task/cpu stats
        - **m** • Memory information
        - **f** • Add/remove columns
        - **o** • Change order of column display
        - **R** • Reverse/normal sort order
        - **H** • Show threads
        - **n #** • Set maximum number of tasks to display
        - **d** • Set update interval
        - **k** • Kill task
        - **r** • Renice task
        - **q** • Quit

200.1 - Measure and Troubleshoot Resource Usage (Network and Bandwidth - netstat)

- `netstat`

    - Displays network connections, routing tables, interfaces and related statistics
    - `-r` • Display routing table
    - `-s` • Display a summary of network connection statistics, by protocol
    - `-l` • Display sockets that are in a listening state (for incoming connections)
    - `-lt` • Display only TCP sockets listening
    - `-lu` • Display only UDP sockets listening
    - `-n` • Do not attempt to resolve numeric IPs to name (reverse DNS), speeds up responses
    - `-c` • Continuously update the display
    - `-p` • Includes additional information (application name and its PID)

200.1 - Measure and Troubleshoot Resource Usage (User Information - w)

- `w`
    - Shows who is logged on to the system and what they are doing
    - `-s` • Short format display
    - `-i` • Display IP address instead of hostname of connection (if possible)
    - `user` • Display information on indicated user only

200.2 - Predict Future Resource Needs (Gather System Usage Statistics - collectd Intro and Install)

- `collectd`
    - Responsible for collecting system statistics; daemon runs constantly
    - Uses plugins to determine the specific statistics that it collects
    - Is not designed as a display mechanism or analyzation engine, but rather a collector of that data
        - There are a number of third-party tools that can display the collected data, that are outside of the scope of the LPIC2 course
    - `/etc/collectd/collectd.conf`
        - Daemon configuration file
        - Most settings are commented out, by default

    - Most global settings will be fine, specific configuration changes will be more likely to be made on the plugins
    - Amount of time to wait between queries for `collectd` statistics is defined by the INTERVAL value in the config file
  - **Install collectd** • `sudo apt-get install collectd`

200.2 - Predict Future Resource Needs (Gather System Usage Statistics - collectd Key Files and Locations)

- `collectd`
  - Enable the data collection on system startup:
    - `ckconfig collectd on`
  - Start the service:
    - `service collectd start`
  - `/usr/share/collectd/types.db`
    - Contains one line for each data-set specification, consisting of two field delimited by space or tabs
    - First field defines the name of the data-set
    - Second field defines the list of data-source specifications (space or comma delimited)
      - Format is inspired by RRDtool's data-source specification
  - `/usr/lib/collectd/rrd/[hostname]`
    - Set of directories, one for each plugin enable for that host
    - Each directory contains one or more RRD formatted file containing the statistics for that data point as defined in the plugin configuration

200.2 - Predict Future Resource Needs (Gather System Usage Statistics - collectd Display Statistics for Capacity Planning)

- `collectd`
  - Install the needed packages/dependencies to log the collected data and display it:
    - `apt-get install collectd lighttpd git php5 php5-cgi`
  - Configure `/etc/php5/cgi/php.ini`
    - Find line `cgi.fix_pathinfo=1` and uncomment it, save and exit the file

- Enable modules for lighttpd
  - `sudo lighttpd-enable-mod fastcgi`
  - `sudo lighttpd-enable-mod fastcgi-php`
- Restart collectd:
  - `service collectd restart`
- Start collectd graph panel
- Clone the cgp (collectd graph panel)
  - `cd /var/www`
  - `git clone http://git.nethuis.nl/pub/cgp.git`
  - `restart lighttpd1`
    - `service lighttpd start`
- View website: http://hostname/cgp/

200.2 - Predict Future Resource Needs (Gather System Usage Statistics - Awareness of Other Monitoring Solutions)

- Nagios
  - Collection of products that include: Nagios XI (Full Enterprise Package), Log Server, Network Analyzer, Fusion, Incident Manager, Core, Reactor
  - Nagios Core is open source, including the collectd-nagios plugin
  - Remaining packages have open source and proprietary components or services
- MRTG
  - MRTG (Multi Router Traffice Grapher)
  - Monitors network traffic load
  - Reads traffic and creates statistics from that information on routers and creates logs for later use
  - Enterprise extensions allow graphing of the data via a web interface
- Cacti
  - Also focused on network monitoring
  - Front end for the RRDtool (Round Robin Database tool we installed above)
  - Provides for the display an graphing of data from the tool and can do limited data collection

- Provides a user management facility

Topic 201 - The Linux Kernel

201.1 - Kernel Components (Source and Documentation for 2.6.x and 3.x Kernels)

- 2.6.x kernel source code
  - `yum install kernel-devel`
  - `/usr/src/kernels/[kernel version]`
- 2.6.x kernel documentation
  - `yum install kernel-doc`
  - `/usr/share/doc/kernel-doc-[kernel verion]/Documentation`
  - **Alternate location (some distributions)** • `/usr/src/linux/Documentation`
- 3.x kernel source code
  - `apt-get source linux-image-$(uname -r)`
  - `/usr/src/linux-[kernel version]`
- 3.x kernel documentation
  - Installs with the source code
  - `/usr/src/linux-[kernel version]/Documentation`
  - **Alternate location (some distributions)** • `/usr/share/doc/kernel-doc[kernel version]/Documentation`

201.1 - Kernel Components (2.6.x and 3.x Kernel - Terms)

- Kernel images
  - zImage
    - Older systems, designed so the kernel would be stored in "low memory" (the first 640kb of RAM), was limited to 512kb as a result in order to support older systems (compressed image size)
  - bzImage (stands for "big zImage")
    - Low memory limitations are largely a thing of the past and most systems permit kernels to be loaded in "high memory" (around the 1mb RAM barrier), most modern kernel images are

above 512kb in size (even compressed)

- Compression types • gzip prior to 2.6.x, bzip2 since then
- Name
    - Typically named with the standard of "vmlinuz-[version]-[architecture]" or "vmlinux-[version].[architecture]"
        - **Example** • *vmlinuz-3.13.0-36-generic* (Ubuntu) or *vmlinuz-2.6.32-642.3.1.el6-x86_64*

201.2 - Compiling a Kernel (Getting Everything Ready)

- Clean the kernel
    - This removes any configurations that are left from other compilations
    - Packages needed:
        - `yum groupinstall "Development Tools"`
        - `yum install ncurses-devel qt-devel hmaccalc zlib-devel binutils-devel elfutils-libelf-devel`
            - All of these tools can be several GB in size, depending on the distribution type already in place (minimum, network, desktop, server, etc)
        - `apt-get install ncurses-devel make gcc`
    - Three types of "clean":

1. `clean` • Delete most generated files while leaving enough to build modules
2. `mrproper` • Delete the current configuration and ALL generated files
3. `distclean` • Remove everything including backup files, leftover patch files, etc.

- Typical clean command:
    - `make clean` (run as +root+, in the kernel source directory)

201.2 - Compiling a Kernel (2.6.x / 3.x General Kernel Compilation)

- Distribution specifics
    - There may be certain packages you need to install depending on your distribution, your distribution documentation will tell you if there are packages specific to compiling your kernel

    - In general, kernel source, make utilities, zlib utilities and gcc and g++ are REQUIRED by all distributions and kernel versions

- 2.6.x and 3.x differences
    - Other than the source and documentation locations that are listed above the, PROCESS of compiling the kernel once the tools and utilities are all in place is the same for each distribution and kernel version
- Creating the configuration file
    - Main types:
        - `make config` • The most involved type, you will be asked questions on everything before compile can start (100s of them), not generally used anymore except in very specialized circumstances
        - `make menuconfig` • Most common method, will read `.config` file in base source directory as template if it exists, reads existing kernel configuration if not
        - `make xconfig` • Same as `make menuconfig` but with the X Window QT graphical libraries
        - `make gconfig` • Same as `make menuconfig` but with the Gnome Desktop and GTK+ libraries
- Create the configuration file:
    - `make menuconfig` (ncurses terminal version)
    - device drivers
        - block devices
            - normal floppy support (remove/exclude)
    - new .config file in place
- Compile the new kernel:
    - `make bzImage` (Note: This can take a LONG time and consume a bunch of memory and disk)
- Make the kernel modules:
    - `make modules` (Note: This can take a LONG time and consume a bunch of memory and disk)
- Make the kernel module installation:
    - `make modules-install` (Note: This can take a while, although not quite so long)
- Copy the new bzImage to the `/boot` directory:
    - `cp /usr/src/linux-[kernel version]/arch/[architecture]/boot/bzImage /boot/vmlinuz-[kernel version].[architecture]`

- Create the `initramfs` file (which calls the "dracut" utility):
  - `mkinitrd /boot/initramfs-[kernel version].[architecture].img [kernel version]`
    - **Example** • `mkinitrd /boot/initramfs-3.13.0-36.x86_64.img 3.13.0-36`
- Modify the boot configuration (GRUB) to test
  - Copy the default configuration, referencing your new values
  - Do NOT make it the default on first boot for testing

201.3 - Kernel Runtime Management and Troubleshooting (LKM - Loadable Kernel Modules)

- LKM
  - Modules that are loaded into memory as needed to supplement the functionality of the base kernel
- Location:
  - `/lib/modules`
    - Directory should match the value of the `uname -a` command when choosing from multiple potential directories
- `uname`
  - Print system information
  - `-a` • Print all information
  - `-s` • Just give the kernel name
  - `-n` • Print the network node hostname
  - `-r` • Print kernel release (often used in downloading kernel specific version info like docs or source)
  - `-v` • Full kernel version
  - `-m` • Machine hardware name
  - `-p` • Processor type
  - `-i` • Hardware (platform)
  - `-o` • Operating system
- Actual module location for specific kernel:

- `/lib/modules/[kernel version]/kernel`
- Subdirectories contained here break the modules into the general categories that they pertain to (i.e. all sound modules can be found in `/lib/modules/[kernel version]/kernel/sound` directory)

201.3 - Kernel Runtime Management and Troubleshooting (depmod and modules.dep)

- Manually copying a module onto the system (pre-compiled)
  - Place it into the appropriate category (based on documentation: block, sound, fs, etc)
- Update the module dependencies on the system or the module will not be loadable
  - `depmod`
  - Looks at all modules under the `/lib/modules/[kernel version]/kernel` directory
  - Generates files called `/lib/modules/[kernel version]/modules.dep and, if needed, /lib/modules/[kernel version]/+map`
  - Only executed on the current RUNNING kernel by default, specify another kernel version as an argument if desired
    - **Example** • `depmod 3.10.0.1-22`
- `modules.dep`
  - A listing for the modules on the system
  - Each module with two ":" delimited fields:
    1. Name of the kernel module
    2. Module it depends on (if no value in 2nd field, no dependency found)

201.3 - Kernel Runtime Management and Troubleshooting (Listing, Adding and Removing Modules)

- `lsmod`
  - List all LOADED modules
  - Three columns of output:
    1. Name of the module
    2. Size of the module (bytes)
    3. Listing of how many other items (modules, processes, etc) are using the module

- `rmmod`
    - Removes a module from memory
    - Make sure there are no active items using the modules or you could have problems
    - `rmmod lp`
        - Removes the line printer module from memory, `lsmod | grep lp` shows it is gone
- `insmod`
    - The old method of adding a module to memory
    - Requires that you know the location and exact name of the module
    - Requires that you know the location and exact name of any modules it depends on, loading them before
    - `insmod /lib/modules/[kernel version]/kernel/drivers/char/lp.ko`
        - Loads the line printer module back into memory, confirmed with `lsmod | grep lp`
- `modprobe`
    - New, easier way to add modules into memory
    - You only need to know the name of the module, not its path or dependencies, if there are any and they exist, they will be loaded automatically
    - `modprobe lp`
        - Loads the line printer module back into memory, confirmed with `lsmod | grep lp`
- `modinfo`
    - Displays a modules parameters and other applicable information on it
    - Items marked as "parm" are able to be passed to the module upon load
    - Example • `modprobe lp reset=1`
        - Add the line printer module to the kernel, indicating that it should reset
- `/etc/modprobe.d`
    - Directory containing files that allow you to make changes to how modules are loaded
    - Each file lists modules and three primary settings for them
        1. **alias** • Think of as another way of referring to said module (nickname)
        2. **install** • Commands and/or parameters to execute when a particular module is loaded into memory

3. **remove** • Commands and/or parameters to execute when a particular module is removed from memory (often added to modules that may require that other services be stopped and/or when you want specific logging to occur when the module unloads)

201.3 - Kernel Runtime Management and Troubleshooting (Viewing and Changing Kernel Parameters in /proc/sys and Using sysctl)

- `/proc/sys`
  - A memory-based filesystem mounted under `/proc` wherein everything is related to the kernel (including modules), everything is stored in memory (RAM) and NOT on the filesystem itself
  - Values can be changed without a reboot or reload by echoing values into them (overwriting) for either kernel parameters OR module parameters
  - Example • `cat "0" > /proc/sys/dev/cdrom/autoclose`
    - Changes the default value of "true" on the autoclose module to "false"
  - Modifying values this way is temporary until system restart
- `sysctl`
  - Displays (and allows you to change) kernel and module parameters on the fly
  - `-a` • Display all kernel and module parameters loaded
    - **Example** • `sysctl -a | grep cdrom`
    - Displays all `cdrom` module parameters
  - Change a parameter by passing it to `sysctl` by full name and new value, repeating the autoclose example for CDROM
    - **Example** • `sysctl dev.cdrom.autoclose=0`
    - Changes the default value from true to false just as the above `/proc/sys` method
  - Modifying values this way is temporary until system restart
- `sysctl.conf`
  - Permanent changes to overwrite default values are stored here
  - Format is the full device and then equal to the new value
    - Example • `dev.cdrom.autoclose = 0`

201.3 - Kernel Runtime Management and Troubleshooting

(Displaying Information About System Hardware)

- `lspci`
    - Lists all hardware devices that are attached to the system directly (permanent attachment, not temporarily, like USB, although the USB controller itself will be listed)
    - Each single line describes one hardware device in the following format:
        - slot, position, bus number, type, manufacturer, model info, powerspec (if reported), revision (if provided)
    - `-v` • Show additional details (verbose mode), typically shows additional flags and parameters that are reported
    - `-vv` • Show even more details, including more flags, memory locations or subsystem designations
    - `-vvv` • Most verbose mode, show everything the command is able to parse from each connected device
    - `-k` • Display the kernel modules that handle the device (both any that are in use, and any that might be available)
- `lsusb`
    - Older systems (prior to 3.x kernels) may not have this command available
    - Produces output similar to `lspci` but restricted to connected USB devices, output shows the following
    - Bus ID number, device number (on that bus ID), major manufacture identification number, minor manufacturer device identifier, manufacturer name (of device or primary detected interface)
    - `-v` • Verbose display, typically not used alone because of the amount of information that would be displayed
    - `-d` • The device that you want to see more information on; see the `lsusb` output and obtain the vendor ID and product ID (in ####:#### format)
        - **Example** • `lsusb -v -d 1d6b:0002`
            - Displays a vast array of information about the root USB 2.0 hub on a system
- `lsdev`
    - Another command to display system hardware information
    - Displays the device (location/slot/port), the dma used, the irq used and any I/O ports reported
    - Information from this command is obtained from the `/proc/dma, /proc/ioports and /proc/interrupts` directories

- Logging hardware information
  - `/var/log/message` (Red Hat/CentOS systems)
    - Contains kernel messages whenever a newly-detected hardware device is added or removed
  - `/var/log/syslog` (Debian/Ubuntu systems)
    - Contains kernel messages whenever a newly-detected hardware device is added or removed

201.3 - Kernel Runtime Management and Troubleshooting (The Device Filesystem - udev)

- `/dev`
  - Contains device files that are attached to the Linux system
  - In the past, would contain a number of files for devices that COULD be attached to the system
- `udevd` (daemon)
  - During boot, used to probe connected devices and add the appropriate files to the `/dev` filesystem
  - After boot, this daemon adds the necessary files to `/dev` filesystem automatically
- `/etc/udev/rules.d`
  - Directory that contains rules that allow you to adjust how these device files are created when detected
  - The numbers in the front of each name determine the order in which these rules are applied
    - The higher the number, the higher the priority (meaning that a rule starting with 60 could be overridden with the rules in a file starting with 90)
  - Each file contains a set of key pair like values, comma separated
- `udevadm`
  - Allows you to monitor what happens when a device is detected and added to the system
    - **Example** • `udevadm monitor`
      - Prints all the events received for UDEV pertaining to the KERNEL (until killed with CTL-C)
- `udevmonitor`
  - Same functionality but for older Linux systems

Topic 202 - System Startup

202.1 - Customizing SysV-init System Startup (Linux Standard Base Specification)

- LSB - Linux Standard Base Specification
  - Joint venture between the Linux Foundation and several Linux distributions
  - An effort to standardize the software system structure (to include the full filesystem hierarchy) of the Linux OS
  - Based on the POSIX specification
  - Binary backward compatibility is maintained with previous versions
    - Designed to produce an ABU (application binary interface) for software vendors to rely on
      - This is achieved (backward compatibility) by adding only new interfaces and not removing old ones until they have been deprecated at least three revisions (about six years)
      - NOTE: LSB 5.0 released in June 2015 does break compatibility with older versions and is the first to incorporate the changes made by FHS 3.0 (Filesystem Hierarchy Standard)
  - Full support on all versions of Red Hat/CentOS/RPM-based distributions, while Debian-based distributions are more limited as a result of the decision to officially support delivery of software packages ONLY on in RPM format (which would need translation to .deb)

202.1 - Customizing SysV-init System Startup (SysVInit Boot Process)

- sysvinit
  - Older system start specification, still only specification on LPIC exams as of 2016
- Alternatives:
  - upstart (previous Ubuntu versions)
  - systemd (most modern distributions as of 2016 have moved to this service management method)
- init
  - First process called during the boot, called the init process
  - Used to start all other processes and services
  - What is started is dependent on the runlevel being booted to
  - Runlevels (Red Hat/CentOS)
    - 0 - Halt (Stop)
    - 1 - Single User Mode

        - 2 - Multiuser mode (no GUI or network)
        - 3 - Multiuser mode (no GUI, with network)
        - 4 - Not defined
        - 5 - Multiuser mode (GUI and network)
        - 6 - Reboot
    - Runlevels (Debian Based)
        - 0 - Halt (Stop)
        - 1 - Single User Mode
        - 2 - Multiuser mode (GUI and network)
        - 3 - Multiuser mode (GUI and network)
        - 4 - Multiuser mode (GUI and network)
        - 5 - Multiuser mode (GUI and network)
        - 6 - Reboot
    - Other runlevels
        - 7,8 and 9 - No standards, most distributions do not define them by default
    - General runlevel uses
        - 1 - fixing serious issues, minimal service availability
        - 3 - general "server" runlevel where GUI is not involved
        - 5 - generally considered "desktop" mode with all services plus GUI availability
- `/etc/inittab`
    - The file used by the init process to determine the default boot level
        - Multiple values can appear
        - Fields appearing in a line
            - `id:runlevel:keyword:comand`
        - Typical default runlevel definition
            - **Example** • `id:3:initdefault:`
- `/etc/rc.d/rc.sysinit`
    - A script that is written to allow the system to boot in a basic state

  - Generally the same as "single user" or "init 1" mode
  - Mounts filesystems, including mounting RAID or LVM block devices
  - Starts key services
- `/etc/rc.d/rc`
  - Script that is run whenever the runlevel of a system changes
    - Primary sections are responsible for running START or KILL scripts
- +$runlevel+
  - Variable that contains the value initially passed into the `/etc/init.d/rc` script to determine the runlevel to which to change
- `/etc/rc.d/rc#.d`
  - Directory corresponding to the desired runlevel that contains start scripts (beginning with a capital "S" and then a number between 1 and 99) or kill scripts (beginning with a capital "K" and then a number between 1 and 99)
  - Each of the start/kill scripts is actually a link to the appropriate service file in `/etc/init.d`
  - This structure facilitates easily adding/removing services from each runlevel by adding/removing links that will not affect the underlying service itself
- Naming convention for start/stop files
  - S[1-99]servicename
  - K[1-99]servicename
    - Numbers designate the order that each start or kill script is executed in, it allows a dependency build
      - **Example** • S13hostname is executed only AFTER the S10network script is called (because we do not want to set a hostname before starting the network)

202.1 - Customizing SysV-init System Startup (/etc/init.d and Modifying Runlevels Scripts)

- `/etc/init.d`
  - Where the actual service scripts are installed in a sysvinit system
  - Linked to by the various rc#.d directories and the contained start/kill scripts
  - Can be used to directly start a service daemon if desired
    - **Example** • `sudo /etc/init.d/nginx start`

  - Location where custom scripts are located that can be added to the runlevels
- `update-rc.d` - NOTE: DEBIAN AND UBUNTU SYSTEMS ONLY
  - Program to create the links from the `/etc/rc#.d directories to the named script in /etc/init.d`
  - **Example** • `update-rc.d mynewservice defaults`
    - Creates links in the `/etc/rc0[,1,2,3,4,5,6].d directories to this script called +mynewservice+ in /etc/init.d`, as this is represented by "defaults" in the argument
    - Defaults can be replaced by one or more values
    - **Example** • `update-rc.d mynewservice start 66 3 5 . stop 76 3 5 .`
      - mynewservice will have start scripts added to runlevels 3 and 5 beginning with number 66
      - mynewservice will have kill scripts added to runlevels 3 and 5 beginning with number 76
    - `update-rc.d [servicename] remove`
      - Disables/removes the service from the configured runlevels
- `chkconfig (within your /etc/init.d/[service script]`) - NOTE: REDHAT AND CENTOS SYSTEMS ONLY
  - Alternative method for adding the script to various runlevels
  - `# chkconfig [runlevels] [start script number] [kill script number]`
    - **Example** • `# chkconfig 35 92 98`
      - Adds to runlevels 3 and 5 directories with a start script number of 92 and kill script number of 98
  - `chkconfig --add mynewservice`
    - Adds/modifies the values in the appropriately-referenced `rc.d` directories
- `chkconfig` (used to determine startup status of service)
  - `--list` • Displays the current status of each script and when it is started or stopped
  - `on/off` • Enables or disables the service on boot (unless indicated, for default runlevels)
  - `--level #` • Applies the on/off only to the specified runlevel

202.1 - Customizing SysV-init System Startup (Changing Runlevels)

- `runlevel`
  - Displays the current runlevel
- `init`
  - Command to change runlevel to indicated value
    - **Example** • `init 3`
      - Changes from whatever runlevel the system is in to runlevel 3, executing all appropriate start/kill scripts along the way
- `telinit`
  - Same as init, with some additional options
  - `-t ##` • The amount of time to wait to change to the indicated runlevel
    - **Example** • `telinit -t 10 5`
      - Change to runlevel 5 after waiting 10 seconds

202.2 - System Recovery (Understanding the Boot Process)

- Process phases
  1. BIOS (UEFI)
  2. bootloader
  3. kernel
  4. post-kernel
- BIOS (UEFI)
  - Happens on "power on" of a system
  - Comes with the systems hardware and is not part of the Linux OS
  - **BIOS (Basic Input/Output System)** • legacy
    - Considered firmware
    - Certain limitations around devices recognized, disk sizes, etc.
  - **UEFI (Unified Extensible Firmware Interface)** • modern
    - Software that extends firmware
    - Can run with BIOS
    - Can read from modern block devices (LVM, RAID)

        - Can parse GUID-based partition tables

- Bootloader
    - Several types of bootloaders
        1. GRUB (legacy grub)
        2. GRUB 2 (modern)
        3. LILO (legacy bootloader)
    - Installed on internal hard disks or external drives (USB)
    - Stored on the MBR (Master Boot Record), generally
        - MBR is at the start of a drive, within the first 215 bytes of the disk
        - Partition table is stored in this same location
            - Legacy MBR may also be in the first sector of a partition, usually when more than one boot loader is present
    - Almost always installed during the boot process
    - Can be installed later and/or reinstalled (usually from a rescue disk (CD, USB, etc))
- Kernel
    - Accomplishes several things:

1. Configures system in order to assign memory addresses to software processes

2. Performs hardware recognition (probing)

3. Uncompresses the initrd (older)/initramfs (modern) images that contain kernel modules needed

4. Initializes LVM or RAID and mounts the root (/) filesystem (as read-only at first)

5. Hands off to post-kernel

- Post-kernel
    - One of three post-kernel service management systems
        1. sysvinit (LPIC-2 exam requirement)
        2. upstart (older Debian/Ubuntu systems)
        3. systemd (most modern distributions, new standard moving forward)

202.2 - System Recovery (GRUB - Legacy Structure)

- NOTE: using CentOS 5.x for legacy GRUB demonstration
- GRUB (versions 1 or less)
  - Replaced LILO (the LInux LOader)
  - Cannot use UUIDs
  - Supports fewer operating systems (than GRUB 2)
  - Does not support LVM or RAID (as a boot loader partition)
  - Simpler configuration files with less options
- `/boot/grub/grub.conf`
  - Configuration file for legacy GRUB and its menu
  - Key fields
    - **hiddenmenu** • If present, the menu will not be display, but a countdown timer may be
    - **timeout** • The value (in seconds) that the menu will wait for a choice of boot images; if none are chosen, then the marked default choice will be booted
    - **splashimage** • A graphic (in `.xpm graphic format, compressed with gzip, must be named +.xpm.gz`) in a specific size (640x480 and 14 colors or less) to be displayed with the boot menu
    - **default** • The choice automatically booted to when no other menu choice is chosen or available (numbered from 0 and up)
  - Structure of menu choices (typical options to know)
    - Title (containing the title of the choice to be booted if chosen)
    - root - indicates the device containing the boot files (kernel and initrd/initramfs)
      - **Example** • `root (hd0,0)`
        - Would be the first disk and first partition on it
        - Note that although the root device itself may actually be something like */dev/hda1*, numbering changes in GRUB
          - In drives, letter "a" being first, would be device "0" and partition "1" being first, would be partition "0"
            - **Example** • */dev/hdb2* would translate to *root (hd1,1)*
    - **kernel** • relative path to the device indicated in the "root" parameter and the name of the kernel file and parameters indicated "ro" for read only and "root=" containing the kernel the location of the root filesystem (which can be on an LVM or RAID device, just the /boot mount

does not support those special block devices)

      - **quiet** • optional parameter for the kernel line that suppresses verbose boot messages
      - **rhgb** • optional parameter for the kernel line that indicates usage of the Red Hat graphical boot feature (progress bar during boot) if available (Red Hat/CentOS systems)
      - Most options that are added during boot are added on this (the "kernel") line
        - **Example** • `s` - added to the end of the kernel line, causes it to boot into single user mode

- Boot alternatives (from menu)
  - Pressing **enter** on the boot menu boots the target highlighted
  - Moving to another item and pressing **enter** boots that target
  - Highlighting a choice and then pressing **e** allows you to edit the title
    - Adding or changing any of the lines, highlighted one at a time
      - **Example** • Adding "s" to the end of the kernel line boots that kernel into single user mode
      - Optionally, if you want to change JUST the kernel line, instead of **e**, press **a** to go right to the kernel line on the menu
    - Once changes are made, press **enter** to return to the previous screen
    - Cancelling changes can be done with **esc**
  - Once changes are made and you want to boot, press **b**
- GRUB command line
  - Press **c** to manually enter the GRUB command line and manually add boot parameters to your system
  - Can sometimes be consider "rescue mode" on an otherwise unbootable system
- Secure GRUB with password
  - Prevents someone with console access from rebooting into single user mode and resetting root password
  - Plain text passwords can be used
  - MD5 encrypted passwords are preferred
  - Password is contained in `/boot/grub/grub.conf`, which is root-viewable (so would require filesystem access to see)
  - MD5 password generation for GRUB

- `grub-md5-crypt`
    - Prompts for a password, repeat the entry, presented with a encrypted password string
    - Copy the MD5 string provided
- Above "title" add (to prevent any menu choice before password)
    - `password --md5  [encrypted string]`
- Optionally, placing password within a particular title will just prevent booting that choice without password
- Rebooting will show menu choices that indicate you have to press **p** to enter a password to boot that choice

- `grub-install`
    - Allows install of GRUB on default or other drive
    - Not needed in GRUB legacy when menu changes are made, read on boot from `grub.conf`

202.2 - System Recovery (GRUB2 - Modern Structure)

- NOTE: CentOS/Red Hat and Debian/Ubuntu systems can be used here
- `/boot/grub2/grub.cfg` • Red Hat/CentOS GRUB 2 configuration file
- `/boot/grub/grub.cfg` • Debian/Ubuntu GRUB 2 configuration file
- Note that these files are generated files and are almost never manually changed since changes would be reverted on next boot to current values
- `/etc/default/grub`
    - Configuration changes are made to this file
    - `GRUB_DEFAULT` • The default boot target (or optionally set to SAVED)
        - Best to use the target title (name) because the order of targets can change after kernel upgrades and other system events
        - If set to SAVED, also requires
            - `GRUB_SAVEDEFAULT` • true or false (remembers last target chosen and makes it default on next boot)
    - `GRUB_HIDDEN_TIMEOUT_QUIET` • true or false to hide the timeout countdown
    - `GRUB_TIMEOUT` • The timeout to choose before default is booted
    - `GRUB_CMDLINE_LINUX_DEFAULT` • Optional parameters for the kernel line, applied to all targets (regular and not recovery targets)

  - `GRUB_CMDLINE_LINUX` • Optional parameters for the kernel line, applied to all targets (regular and recovery)
  - (Optional) `GRUB_GFXMODE` • Resolution to display graphical grub if video vard supports VBE
  - Changes made to these files are not applied until one of two commands is run (depending on distribution)
    - `grub2-mkconfig` • Red Hat/CentOS
    - `update-grub` • Debian/Ubuntu
      - Either command then reflects the changes made in the `grub.cfg` file in the location specific to that distribution type
- `/etc/grub.d`
  - Directory containing files for specific types of changes (i.e. titles for boot targets)
  - Each of the files, numbered and by type, are "executed" when you run the appropriate grub update command
  - The values of these scripts are then "compiled/concatenated" to form the `grub.cfg` file, which is how the menu system, defaults and options are read on boot
  - `/etc/grub.d/40_custom`
    - Generally where custom kernel targets are manually added
    - Copy/paste the values from `grub.cfg as a template, then add your changes and view the new grub.cfg` file
- Single user/recovery mode in GRUB 2
  - Edit the line of the target you want as in GRUB legacy
  - At the end of the kernel line, add `init=/bin/sh`
  - Boot the system with **Ctrl-x**
  - Remount the root filesystem
    - `mount -o remount,rw /`
  - Change the password
  - Assuming SELinux is enabled, make sure to mark it to rescan file changes on next boot
    - `touch /.autorelabel`
  - Reboot the machine
    - `/sbin/reboot`

202.2 - System Recovery (Filesystem Recovery)

- `fsck`
  - Regardless of GRUB legacy or GRUB 2
  - Generally needed because a filesystem was not properly shutdown (i.e. power outage)
  - Metadata is stored in memory for files, so if that data is not flushed to disk (part of normal shutdown or reboot), it can cause anything from corruption to "lost inodes"
  - **Example** • `fsck /dev/sdb1`
    - Runs the utility against the filesystem on the second disk's first partition
    - Any questions regarding fixing a filesystem should be answered "yes"
  - **Example** • `fsck -y /dev/sdc1`
    - Runs the utility against the filesystem on the third disk's first partition
    - You will not be prompted to fix files, assumes yes
- NOTE: Do not run on a running/live filesystem
  - Causes issues instead of fixing them
  - Unmount the target filesystem before running

202.3 - Alternate Bootloaders (Awareness of LILO, Syslinux, PXELinux)

- LILO
  - LInux LOader
    - Older and replaced by GRUB
    - `/etc/lilo.conf` • Main configuration file
      - `boot` • Determines the boot drive
      - `default` • Default image to boot without manual entry
      - `install` • Determines whether menu or lilo text prompt appears
      - `prompt` • Same as install
      - `timeout` • Value before system boots to default target
      - `image` • The kernel image (full path `/boot/[kernel name]`)
      - `root` • Root filesystem (i.e. `/dev/hdb1`)
      - `label` • Name of the target

    - Changes do not take affect until `lilo` command is executed (which writes to the MBR)
- SYSLINUX
  - Associated with the Syslinux Project. Intended as one of several boot loaders that do specific things (but are not intended to boot typical Linux distributions).
  - Designed to be placed on a disk
    - **Example floppy install** • 'syslinux –install /dev/fd1'
  - `/syslinux/syslinux.cfg`
    - Directory created on the device to which it is installed
  - Key Options
    - **DEFAULT** • Default boot target
    - **LABEL** • Name of target
    - **SAY** • Text to print when chosen
    - **KERNEL** • Name of the kernel file
    - **APPEND** • Additional options passed to kernel line
- EXTLINUX
  - Associated with the Syslinux Project. Intended as one of several boot loaders that do specific things (but not intended to boot typical Linux distributions).
  - Sames as SYSLINUX, but supports EXT2/3/4 filesystems on USB
  - Installed to a mounted filesystem (rather than raw)
    - **Example** • 'extlinux –install /mymount `(where /mymount` is where the intended device is mounted on the system)
  - Configuration options in the `extlinux.conf` file are identical to those found in the SYSLINUX configuration file
- ISOLINUX
  - Associated with the Syslinux Project. Intended as one of several boot loaders that do specific things (but are not intended to boot typical Linux distributions).
  - Designed to make bootable CD-ROM (ISO 9660 filesystems)
    - Extension to make bootable called "El Torito Bootable CD Specification"
  - Install SYSLINUX to use
  - Directory that contains all packages and dependencies (called `cd-root`)

  - Create `isolinux directory under that (cd-root`)
  - Copy `isolinux.bin, included with SYSLINUX package, into the isolinux` directory
  - Create `isolinux.cfg` file in that same directory (format is the same as SYSLINUX)
  - Create a `kernel directory under the cd-root` and copy the kernel into this directory
  - From `cd-root`, execute a command like the following example:
    - `mkisofs -o /tmp/mycdout.iso -b isolinux/isolinux.bin -c isolinux/boot.cat -no-emul-boot -boot-load-size 4 -boot-info-table -- ../cd-root`
  - Which creates the file `/tmp/mycdout.iso`
  - Which can be burned to CD and booted from

- PXELINUX
  - Associated with the Syslinux Project. Intended as one of several boot loaders that do specific things (but are not intended to boot typical Linux distributions).
  - Install SYSLINUX
  - Create directory called `/tfpboot`
  - Copy `pxelinux.o into /tfpboot` (file comes with SYSLINUX package)
  - Copy kernel and initrd images into `/tfpboot`
  - Copy library called `ldlinux.c32 into /tfpboot` (file comes with SYSLINUX package)
  - Create `/tfpboot/pxelinux.cfg` file
    - Syntax identical to SYSLINUX

Topic 203 - Filesystem and Devices

203.1 Operating the Linux Filesystem (Displaying Filesystem Mounting Information)

- `mount`
  - Used to mount filesystems of various types (limited by the filesystem type of the disk/drive/share and filesystem types that the existing kernel supports)
  - Running by default displays all current mounts on the system
    - Options display are: device, mount point, filesystem type and any options used
  - `-n` • don't update the `/etc/mtab` file

- `/etc/mtab`
    - Filesystem mounts are stored in detail here (and is what is read by the `mount` command)
    - This file is automatically updated any time a filesystem is mounted/unmounted
        - Managed by `mount/umount` commands
- `/proc/mounts`
    - Also contains filesystem information
    - Managed by the kernel and may be more accurate (updates cannot be excluded by mount command as "mtab" file can)

203.1 Operating the Linux Filesystem (Mounting and Unmounting Filesystems Manually)

- `umount`
    - Used to unmount filesystems from their mount points
    - `-f` • force the filesystem to unmount (caution - filesystems in use that are unmounted can cause problems including system instability)
- `fuser`
    - Used to identify processes that are using files or sockets
        - **Example** • `fuser /mnt/data`
            - Would display processes (PID) that are using the filesystem (locked files, open files, etc)
            - **Example output** • `/mnt/data:          5211e`
                - The PID represented by the number
                - **c** • Current directory (i.e. you are trying to unmount a directory you are currently in)
                - **e** • Executable running
                - **f** • Open file
                - **F** • Open file (being written to)
                - **r** • Foot directory (you are trying to unmount the root filesystem)
                - **m** • mmap or shared library
    - `-v` • verbose display mode

        - `-k` • Kill any processes accessing the file or mount
- `mount`
    - Used to mount filesystems of various types (limited by the filesystem type of the disk/drive/share and filesystem types that the existing kernel supports)
    - `-t` • allows you to indicate the type of filesystem to be mounted (ext2/3/4, xfs, fat32, ntfs, etc)
        - NOTE: Any filesystem referenced must have an associated avaialble kernel module OR have support compiled into the kernel directly
        - **Example** • `mount -t ext3 /dev/xvdj1 /mnt/data`
            - Mounts the device `/dev/xvdj1 to /mnt/data` with filesystem type "ext3"
    - `-o` • Can be used to provide a list of options to the mounted filesystem to determine permissions, modes, etc. (options are command delimited)
        - `remount` • Remounts the indicated filesystem to the same location
        - `ro` • Allows the mounting of a filesystem in "read only" mode
        - `rw` • Allows the mounting of a filesystem in "read/write" mode (default)
        - `username` • Provides a username to use (used in remote filesystem mounts)
        - `password` • Provides a password to use (used in remote filesystem mounts)
    - `-a` • Used to read the `/etc/fstab` file and mount filesystems listed there not currently mounted

203.1 Operating the Linux Filesystem (Mounting Filesystems Automatically with /etc/fstab)

- `/etc/fstab`
    - Determines local and remote filesystems that are mounted automatically on boot
    - Each line determines a filesystem with all necessary options, structure contains the following:
        - **Example** • `/dev/xvdj1    /mnt/data    ext3    defaults    0 0`
            - **Device to mount** • The actual physical/virtual/remote device to mount
            - **Mount point** • The local directory to which to mount the filesystem
            - **Filesystem type** • The filesystem type of the device (must be supported by kernel module or compiled into kernel)
            - **Mount options** • A comma delimited list of options (see mount options below)

        - **Dump level** • A numeric value indicating whether the "dump" command would perform a backup on the filesystem (rarely used in distributions now)

        - **Filesystem check value** • A Boolean to determine whether a `fsck` needs to be run on boot on this filesystem before mounting

- `e2label`

    - Responsible for setting a custom name/label to a device that can be used during mounting

    - Prevents problems associated with device changes (plugging into another port can change `/dev/sda1 to /dev/sdc1, for example) in the /etc/fstab` file

    - **Example** • `e2label /dev/xvdj1 mydata`

        - Sets the device `/dev/xvdj1 as it is now to the name mydata`

        - ++Example `/etc/fstab` change++ • `LABEL="mydata"    /mnt/data    ext3    defaults    0 0`

- `blkid`

    - Used to display the UUID (Universally Unique IDentifier)

    - A more "stable" method of mounting devices as this value will never change

    - **Example** • `blkid | grep xvdj1`

        - Displays the UUID of the `/dev/xvdj1 device (i.e. /dev/xvdj1: UUID="4b0f5600-652f-4466-a4b8-52ebc752cf62" TYPE="ext3"`)

        - ++Example `/etc/fstab` change++ • `UUID=4b0f5600-652f-4466-a4b8-52ebc752cf62    /mnt/data    ext3    defaults    0 0`

- `/etc/fstab` mount options

    - **Defaults** • Translates to options rw,suid,dev,exec,auto,nouser,async,relatime (see individual options below)

    - **rw** • Read/write filesystem

    - **suid** • Enables the suid permission set to be applied to the filesystem (nosuid will be disabled)

    - **dev** • Allow device files to be used on the filesystem

    - **exec** • Permit executable files (noexec to disable allowing executable files)

    - **auto** • Used when the `-a option is passed to the mount` command, so the filesystem will be automounted as a result

    - **nouser** • Require `sudo`/superuser to mount the filesystem ("user" to allow a single user to mount, "users" to allow any user to mount/umount)

- **async** • Allows metadata to be stored in memory as new files/directories are created (sync forces all writes at once but can cause performance issues)
- **relatime** • Enables updating the file/directory access timestamp (can cause performance issues, norelatime will be disable)

- `sync`
  - Largely unused in modern Linux distributions
  - Forces all filesystems to have all data in cache/memory immediately written to disk
    - Used to be a common practice before rebooting or powering of a system (often run multiple times just prior)
- Additional mount options can be found in the `mount` man page for each major filesystem

203.1 Operating the Linux Filesystem (Swap Space)

- `swapon`
  - Enable devices and files for swapping (paging)
  - `-a` • Enable ALL swaps
  - `-s` • Display active swap space files/partitions
- `swapoff`
  - Disable devices and files for swapping (paging)
  - `-a` • Disable ALL swaps
- Swap in `/etc/fstab`
  - **Example of swap partition** • `/dev/sda1 swap swap defaults 0 0`
  - **Example of swap file** • `/mnt/swap swap swap defaults 0 0`
- NOTE: Creating swap files is covered later based on the exam objective order for the LPIC-2 exam

203.2 Maintaining a Linux Filesystem (Filesystem Types and Creating Them)

- Key filesystem types

1. **ext2** • Older, can be used on USB devices, does not support journaling
2. **ext3** • Journaled filesystem (designed to make filesystem repair faster by being able to replay metadata events)
3. **ext4** • journaled filesystem, can handle larger than 2TB files (up to 16TB), max filesystem size increased from 16TB tp 1EB (exabyte)

4. **xfs** • Non-journaled, high performance filesystem optimized for large data sets with many read/write operations, supports larger individual files (8EB) and filesystem sizes (16XB)
5. **btrfs** • B-tree structured filesystem, allows snapshots (point in time backups, in real time)

- `fdisk`
    - Used to format or probe disks and partitions
    - `-l` • List all detected devices and partitions
    - Used alone, with a device, begins the process of partitioning that device (i.e. `fdisk /dev/xvdj`)
- `mkfs`
    - General command to create a filesystem of the indicated type on a device partition
    - `-t` • allows you to indicate the filesystem type to create (ext2/3/4,xfs,etc)
    - **Example** • `mkfs -t ext3 /dev/xvdj1`
        - Creates an ext3 formatted drive on the "xvdj1" partition
    - Cannot be run on a device alone, must include partition (i.e. `/dev/xvdj1 vs. /dev/xvdj`)
    - Alias for the actual command determined by the filesystem type passed in through the `-t` parameter
        - `mkfs -t ext4 = mkfs.ext4, mkfs -t xfs = mkfs.xfs`, etc

203.2 Maintaining a Linux Filesystem (Change and View EXT Based Filesystems)

- `fsck`
    - In addition to the earlier material, as it applies to EXT filesystems, is just an alias for the appropriate filesystem command like `mkfs`
        - `fsck /dev/xvdj1 (an ext3 filesystem) = fsck.ext3 /dev/xvdj1`
- `dumpe2fs`
    - "Dumps" information about the indicated device (in detail)
    - `-h` • Exclude block group descriptors (vast amount of information not generally needed)
    - Important information displayed includes
        - **Features of the filesystem** • Used to determine what the filesystem in question supports (journaling for example)
        - **inode count** • Each file has one, this tells you the max files the filesystem supports
        - **Reserved block count** • How much is reserved for the superuser

        - **Mount options (default)** • Both options passed during mount as well as any that are compiled into the filesystem

- `tune2fs`
    - Allows changes to the filesystem
    - Many options are NOT changeable with this command (number of inodes, block size, etc)
    - `-m` • Change the amount of space reserved for superuser
    - `-o` • Change mount options (will error if mount option is compiled in and cannot be changed)
    - `-L` • Apply to LABEL name (if exists)
    - `-U` • Apply to UUID (if exists)
- `debugfs`
    - Allows you to perform certain debugging tasks in an interactive manner
    - **Example** • `debugfs /dev/xvdj1`
        - Presents you with a "debugfs:" prompt, from which you can execute commands
            - `ls` • List the files in the current directory
            - `stat [filename]` • Obtain information about named file
            - `help` • Display command options (many commands are standard Linux filesystem commands and operate in the same manner)
    - Can be used to recover deleted files under certain circumstances (on unmounted filesystems)
        - At "debugfs:" prompt, list deleted files with `lsdel`
        - Recovery, if file is found, with `undel`
            - NOTE: The amount of time that passed between the deletion of the file and its recovery affects the ability to recover, the longer between deletion and recovery attempt, the more likely the file is unrecoverable
            - NOTE: This process does not work well on some virtual machines and their filesystems

203.2 Maintaining a Linux Filesystem (Change and View XFS Based Filesystems)

- `xfsprogs` • Red Hat/CentOS XFS support and utilities package
- xfs is the default filesystem for Red Hat/CentOS versions 7+
- `xfsdump`
    - Used to backup an xfs filesystem

    - -f • File dump method
        - **Example** • xfsdump -f /mnt/bkup /mnt/xfsdata
            - Creates a file-based backup of the /mnt/xfsdata directory (mounted with an xfs filesystem) in the /mnt/bkup directory (regardless of mount or format)
            - Prompts for a label (name of your backup) and media label (left over from when tapes were the preferred medium, for file base, provide the name of the device you are copying to or something else to indicate destination)
    - -l # • The level of backup to be performed (0 - full, 1 - changed files since last backup, 2 - files changed since last 0 or 1 backup)
- xfsrestore
    - Used to restore an xfsdump backup
    - -f • File restore method
        - **Example** • xfsrestore -f /mnt/bkup /mnt/xfsnewlocation
            - Would restore the xfsdump from /mnt/bkup to the new location
    - -t • List the files in the backup while restoring the files
    - -i • Enter interactive mode to build a file list to restore
        - Creates a "->" prompt where you can execute common file commands (ls, cd, add, delete, extract, quit, help)
        - add [/path/to/file/filename.ext] • Adds the indicated file to the restore list
        - End and begin the restore in the command prompt with the extract command
- xfs_info
    - Lists basic xfs filesystem info on the indicated device
- xfs_repair (and xfs_check)
    - xfs-specific repair utility on the indicated device
    - -n • Just perform a check on the filesystem, do not start repair (older distributions had separate command xfs_check)

203.2 Maintaining a Linux Filesystem (Creating Swap Files and Partitions)

- mkswap
    - Prepares a partition (or specially-formatted file) to be used as swap space
        - **Example partition** • mkswap /dev/sdc1

        - **Example file** • `mkswap /opt/swap/swap_file`
    - Either can be added to `/etc/fstab`, as discussed in the swapon/swapoff lesson
- `dd`
    - Convert and copy a file (in this context, used to create a specially formatted file to use for swap space)
    - Requires the following
        - `if` • Provides the input (or device) that will be used to create the initial file
        - `of` • The file to be created, full path and name
        - `bs` • Block size
        - `count` • The size, in MB (default), of the file
        - **Example** • `dd if=/dev/zero of=/opt/swap/swap_file bs=1M count=1024`
            - Uses the `/dev/zero device (a constant stream of zeros) to create a swap file named /opt/swap/swap_file` with a block size of 1MB and a total size of 1GB (1024MB)
    - `mkswap` would then be run on the file
- Driver monitoring (smart devices)
    - Self-monitoring, analysis and reporting
        - Developed to better monitor and report on hard drive issues before they happen (does not apply to virtual devices)
    - `smartd`
        - Daemon needed to enable
    - `smartctl`
        - Provides information on the indicated device
        - `-H` • Overall health report on the device
        - `-i` • More details on the device

Topic 204 - Advanced Storage Device Administration

204.1 Configuring RAID (RAID Levels)

- Exam testable for RAID levels 0, 1, 5
- Common RAID levels are

1. **RAID 0** • Treating multiple drives as a single device (striping), high performance but losing one disk in the pool will corrupt the entire RAID group

2. **RAID 1** • Taking one or more disks and mirroring them to one or more disks in another group, lower performance (as each write is made twice), only 50% of installed capacity is available, but redundancy is high making recovery more likely from a failure

3. **RAID 5** • Requires three or more drives to implement, pool of available storage is reduced by parity usage on each drive (parity data is written across each disks in a round robin fashiion), recovery from a single disk loss is by recreating the failed drive contents from parity data on the other drives

- Other RAID levels (not testable)

  - **RAID 4** • Three drives, one used to store all parity data, which is used to recover any failed drive in the access pool

  - **RAID 10** • Combines RAID 1 and RAID 0, two (or more) drives are put into a RAID 1 for redundancy and then merged into RAID 0 for larger capacity and performance

  - **RAID 50** • Combines RAID 5 and RAID 0, two (or more) drives are put into a RAID 5 for redundancy and then merged into RAID 0 for larger capacity and performance

  - **RAID 0 + 1** • Two RAID 0 devices are used to form a RAID 1 device (often used in software RAID)

- **Hardware RAID** • RAID devices created and managed by a special hardware plugin interface card, typical of higher performance server systems

- **Software RAID** • The kernel substitutes management of the RAID array and no special hardware is needed (this is what is tested during the LPIC-2 exams)

- When using software RAID, disk types that are valid members

  - Entire disks

  - Partitions

  - Logical volumes

    - Try to keep the disk/partition/volume sizes the same in order to avoid wasted space (the smallest size will determine the overall available capacity of the RAID group)

204.1 Configuring RAID (Preparing Your Devices for Software RAID)

- `fdisk`

  - Used to partition disk devices

  - `-l` • List all devices and their partitions detected on the system

- `/dev/[device]` • Begin the partitioning process of the indicated device name
- Within the `fdisk` utility, interactive mode (specific to creating partitions and setting partition type)
  - `n` • Create new partition
  - `p` • Define it as primary (vs. extended) - NOTE: best for Software RAID
  - `e` • Define it as extended (vs. primary)
  - `#` • The number of the primary or extended partition to define

204.1 Configuring RAID (Configuring Your RAID Device)

- `mdadm`
  - Used to create software RAID devices
  - Can be used to create any supported RAID level
  - `-C` • Create a new array
  - `-l` • The type of array to create
  - `-n #` • The count of devices/partitions/volumes to be placed into service in the array
  - `-x` • Optional parameter to indicate a "spare" device for use in failures
    - **Example** • i`mdadm -C /dev/md5 -l raid5 -n 4 /dev/sdb1 /dev/sdc1 /dev/sdd1 /dev/sde1 -x 1 /dev/sdf1`
      - Creates a RAID 5 array device called "md5" containing four disks in the pool with one extra available for failure recovery
  - `--detail [/dev/[arrayname]]` • Displays all RAID information for indicated array device
- `/proc/mdstat`
  - File containing some RAID details (less information that the `--detail` command indicated above)
  - Shows the state (active/inactive) of the RAID device
- Format the now active RAID array as a normal filesystem and mount as desired
  - **Example** • `mkfs -t ext4 /dev/md5`
    - Formats the array as an ext4 filesystem, it can then be mounted as any other drive or placed into `/etc/fstab`
- `mdadm`

  - Can be used to manually active an array
  - **Example** • `mdadm --detail --scan --verbose`
- `/etc/mdadm.conf`
  - Create this file using the output from the above command (the output from that command is what is read during boot from that file by the kernel and then activates the array)

204.1 Configuring RAID (Managing Failure and Recovering RAID Devices)

- Software RAID, device will fail
  - `mdadm /dev/[array name]`
    - Indicates a device has failed and recovery in progress, array is clean or has failed
  - `mdadm --fail /dev/[array name] /dev/[device to fail]`
    - Forces the failure of a device so the recovery process starts
    - The device indicated shows failed and is marked as such
    - % of rebuild/recovery with `mdadm /dev/[array name]` shows until complete
      - As long as the array can recover with available resources, underlying filesystem will remain unaffected
- Add a new device as a spare in a RAID set
  - `mdadm --add /dev/[array name] /dev/[device to add]`
    - Adds the device to the indicated RAID array
    - NOTE: Only one device can be added at a time, but many can be added as spares as needed
- Remove a device from a RAID set
  - `mdadm --remove /dev/[array name] /dev/[device to remove]`
    - NOTE: A failed device remains in a configured RAID array until removed with the above command
- Delete an array completely
  - `unmount the RAID filesystem`
  - `mdadm --stop /dev/[array name]`
  - Remove the `/etc/mdadm.conf` file (or entry if multiple RAID devices exist)
- Changes made to an array do NOT update the `/etc/mdadm.conf file automatically,`

`be sure to rerun the command when all changes are complete (mdadm --detail --scan --verbose > /etc/mdadm.conf`)

204.2 Adjusting Storage Device Access (iSCSI Network Storage - Target Configuration)

- iSCSI target
    - The 'server' that is making the storage available to clients (called 'initiators')
- Installation
    - If needed, the package `scsi-target-utils` needs to be installed
- `/etc/tgt/targets.conf`
    - Contains the target storage device and initiator parameters for use
- Naming convention for targets
    - `iqn.year-month.domain_reverse_order:target_name`
        - For example, for a target called +myiscsi+ on a server host called +target.example.com+ on the 1st of December in 2016, the name would be `iqn.2016-12.com.example.target:myiscsi`
- Example `/etc/tgt/targets.conf` (using above example on device iscsi)

    -
      ```
      '<target iqn.2016-12.com.example.target:myiscsi>
         backing-store /dev/iscsi
         iniator-address [network IP of client(s)]
      </target>'
      ```
- Start the service
    - `/etc/init.d/tgtd start`
- Verify enabled and available
    - `tgt-admin --show`
- Filesystem
    - The device used in the 'backing-store' should neither be partitioned or formatted, you are making the storage device and not the filesystem, available for use

204.2 Adjusting Storage Device Access (iSCSI Network Storage - Initiator Configuration)

- iSCSI iniator

    - The 'clients' that will be utilizing the indicated storage
- Installation
    - If needed, the package `scsi-initiator-utils` needs to be installed
- `iscsiadm`
    - Utility used to discover iSCSI storage provided by a target server
    - `-m discovery` • Used to discover available targets from server
    - `-t sendtargets` • Displays available targets discovered
    - `-p [IP Address of target server]` • The IP of the target portal (IP:PORT), if no port used, default 3260 will be
    - NOTE: on first run of this command, the 'iscsid' service will start
- Automatic connection (assuming client IP is configured as allowed in `/etc/tgt/target.conf` on target server)
    - `/etc/init.d/iscsi restart`
        - Restarts iscsi services, connecting to discovered/permitted devices
- `/etc/iscsi/iscsid.conf`
    - Configuration file for service, no changes are generally needed and it outside the scope of what you will be tested on in the exam
- Status of devices connected
    - `/etc/init.d/iscsi status`
- `udev filesystem`
    - Maps the target(s) to device files in the `/dev` filesystem (which is displayed in the status command above)

204.2 Adjusting Storage Device Access (iSCSI Network Storage - Mounting and Using the Device)

- `fdisk`
    - Partition the device as any other disk device (device name pulled with previous status command)
- Format
    - Any valid filesystem type can be used to create a filesystem after partitioning

- Enable the iscsi service (for reboot)
    - `chkconfig iscsi on`
- `/etc/fstab`
    - Entry for iSCSI device that has been partitioned and formatted is like any other
    - NOTE: Advisable to add the `_netdev` options since it waits for the network to start before attempting to mount the device
        - Example entry in `/etc/fstab`
            - `/dev/sdd1       /mnt/iscsi      xfs       _netdev      0 0`
                - Mounts the device called `/dev/sdd1 on the /mnt/iscsi directory as an 'xfs' filesystem AFTER the network starts (indicated by the _netdev` option)

204.3 Logical Volume Manager (Physical Volume Group Creation)

- `lvm`
    - Logical Volume Management • Used to combine multiple storage devices into a single volume that can be expanded or reduced within the combined storage limits of the sum of the devices
    - Can be done with full devices or single partitions
- Snapshots
    - lvm allows snapshots to be taken of managed volumes, solving traditional problems associated with backups by using the snapshot to eliminate open/incomplete files during backups
- `pvcreate`
    - Takes multiple storage devices and/or partitions and combines them into a single storage device; all or part of the storage can be used at any time and it can be expanded or reduced as needed (including adding additional devices to the volume group)
    - `pvcreate /dev/xvdf /dev/xvdg /dev/xvdj`
        - Creates a physical volume that we can then add to our volume group to create an lvm storage volume
- `pvdisplay [device name]`
    - Displays all the drive characteristics of the indicated device

204.3 Logical Volume Manager (Volume Group Creation)

- `vgcreate`
  - Used to take one or more physical volumes created with the `pvcreate` command and add them to a volume group with a common name
  - `vgcreate VG0 /dev/xvdf /dev/xvdg`
    - Creates a volume group called 'VG0' containing the storage resources of each device
  - `-s` • Allows you to specify the extent size (default 4mb), which determines the maximum number of files your volume can contain (and can affect performance in smaller numbers)
    - NOTE: For LVM v1, there is a limit of 65,534 extents, thus the largest filesystem a volume can contain at 4mb (default) size is 256gb
    - NOTE: For LVM v2, this limitation was removed
    - Largest extent size is 16GB
    - Extents CANNOT be adjusted after the fact
- `vgdisplay [volume group name]`
  - Displays all the characteristics of the indicated volume group containing the devices as added during creation
  - `-v [volume group name]` • Displays any physical or logical volumes associated with the indicated volume group

204.3 Logical Volume Manager (Logical Volume Group Creation)

- `lvcreate`
  - Allows for the creation of a logical volume from a volume group
  - `-n [name]` • Assigns the indicated name to the logical volume
  - `-l [#]` • Create with defined number of extents
  - `-L [#M/G/T]` • Create with the amount of space in megabytes/gigabytes/terabytes (provided it is within the available limitations of the physical volumes and associated volume group)
    - NOTE: If exceeding the limit of available space, automatically resizes to take only max size
    - NOTE: If the indicated amount is not a multiple of the extent size of the volume group, rounds up to the nearest number that is
  - `-i [#]` • Create the logical volume using the indicated number of physical volumes (striping) rather than a chunk off the first available device

- Naming conventions
  - `/dev/VG[#]`
    - The actual device for the volume group, once logical volumes exist, will be one or more entries under this (for example, `/dev/VG[#]/lv[#]`)
    - NOTE: The lv[#] devices are links to the actual device names, which is `/dev/dm-[#]` or `/dev/mapper/VG[#]-lv[#]/dm-[#]`
      - For example, with a volume group named 'VG0' and a logical volume named 'lvtest0', the full path to the devices would be `/dev/mapper/VG0-lvtest0, which would link to ../dm-[#]`
- Create a filesystem
  - Partitioning is not necessary
  - `mkfs -t ext4 /dev/mapper/VG0-lvtest0` (would create an 'ext4' filesystem on the indicated logical volume)
- `mount`
  - Create a mount point and mount using standard mount command
    - `mount -t ext4 /dev/mapper/VG0-lvtest0 /mnt/logvoltest` (would mount the 'ext4' filesystem in our example logical volume on the `/mnt/logvoltest` directory)
- Re-running `vgdisplay -v VG0` in our example now shows the logical volumes associated with the volume group 'VG0'
- `lvdisplay [/dev/mapper/VG0-lvtest0]` • Displays all logical volume information for the indicated logical volume storage
- `lvremove [logical volume]`
  - Used to remove the indicated logical volume (and all associated data)
    - NOTE: Can only be run when the filesystem is unmounted

204.3 Logical Volume Manager (LVM Maintenance - Extending, Reducing and Resizing Logical Volumes)

- `pvcreate [additional device]`
  - Add additional indicated device to the available list of physical volumes
- `vgextend [volume group] [physical device]`
  - Allows the addition of a new physical volume to the indicated volume group
  - NOTE: Devices can only be added one at a time

- `lvdisplay [/dev/mapper/VG0-lvtest0]` • Displays all storage information, including current and maximum available size, of the indicated logical volume
- `lvextend`
  - Allows the extension of the logical volume up to the maximum available volume group and associated physical volume limitations
  - `-L [#M/G/T]` • Extend the indicated volume by the amount indicated
  - `-l [#]` • Extend the indicated volume by the number of extents indicated
- `resize2fs [/dev/mapper/VG0/lvtest0]`
  - Used to resize ext2/3/4 logical volumes as configured with `lvextend or lvreduce`
  - NOTE: Extending a filesystem can be done while the logical volume is still mounted without risk to the underlying data
- Reducing a logical volume
  - Must be done in the following order to avoid data corruption:
    1. Unmount the filesystem
    2. Force `fsck` to check the filesystem
       - `fsck -f [/dev/mapper/VG0-lvtest0]`
    3. Use `resize2fs` to reduce the filesystem
       - `resize2fs [/dev/mapper/VG0-lvtest0] [#M/G/T]`
    4. Use `lvreduce` to reduce the size of logical volume
       - `lvreduce -L -[#M/G/T] [/dev/mapper/VG0-lvtest0]`
         - NOTE: The amount used in `lvreduce` should match the reduction from the `resize2fs` command
    5. Remount the filesystem

204.3 Logical Volume Manager (LVM Maintenance - Snapshots)

- `lvcreate`
  - Can be used to generate snapshots; backups can then be taken from the mounted snapshot, which will be a static filesystem and not have traditional backup issues (open files, locked files, etc)
  - `-n [name]` • Assigns the indicated name to the logical volume snapshot

- `-l [#]` • Create with the number of extents

- `-L [#M/G/T]` • Create with the amount of space in megabytes/gigabytes/terabytes (provided it is within the available limitations of the physical volumes and associated volume group)

- `-s` • Create a snapshot rather than a new logical volume

  - Using our example, `lvcreate -L 20G -s -n mysnap0 /dev/mapper/VG0-lvtest0` would create a snapshot called "mysnap0" as logical volume `/dev/mapper/VG0-mysnap0`

- The snapshot exists in the same `/dev/VG[#]` or `/dev/mapper/[VG#-lv#]` structure

- Can then be mounted

  - For example, `mount -o ro /dev/mapper/VG0-mysnap0 /mnt/backup` would mount the snapshot called "mysnap0" in the `/mnt/backup` directory, read only

- `lvremove`

  - Can be used to remove the indicated snapshot

Topic 205 - Network Configuration

205.1 Basic Networking Configuration (Network Commands - route)

- `route`

  - Tells a network packet how to get from the point of origination to its destination (usually through either a gateway or a static route)

  - Executed alone, displays routing table (destination, gateway, genmask, flags, metric (speed), ref and use iface)

  - Can be used to add/delete routes (static or default) on the indicated host and adapter

  - Add a route with a gateway

    - `route add -net 10.1.1.0 netmask 255.255.255.0 gw 10.1.1.1` (would add a gateway to IP 10.1.1.1 that handles all outgoing traffic from the 10.1.1.0 network)

  - Add a route with a default gateway (for all network traffic without another matching rule)

    - `route add default gw 10.1.1.1`

  - Remove a static route

    - `route del -net 10.1.1.0 netmark 255.255.255.0 gw 10.1.1.1`

  - Remove a default gateway

    - `route del default`

205.1 Basic Networking Configuration (Interfaces - ifconfig)

- `ifconfig`
  - Used to display the configuration of active network devices
  - Displays flags (UP, BROADCST, MULTICAST, RUNNING), mtu (maximum transmission unit - default 1500), inet xx.xx.xx.xx (IPv4 address), netmask xx.xx.xx.xx, broadcast address (determined by combination of network address and netmask), inet6 xx:xx:xx:xx:xx:xx (IPv6 address), txqueuelan # (the speed of the device), RX packets (received correctly), RX errors (received errors), TX packets (transmitted correctly), TX errors (transmission errors)
  - If executed with no parameters, displays all active devices and the above configuration information of each
  - Can specify a specific adapter
    - i.e. `ifconfig eth0` only displays information for the 'eth0' adapter
  - Can be used to configure an adapter that is not configured (temporarily, will not be saved on reboot)
    - i.e. `ifconfig [adapter] [IPv4 address] netmask [netmask value] broadcast [broadcast address]`
    - Broadcast can be omitted as it can be calculated based on the IP and netmask
  - Promiscuous mode
    - A mode that can be applied to a network interface that allows it to listen for/capture all traffic on the network rather than just traffic meant for itself (is often used to facilitate network monitoring)
    - `ifconfig promisc/-promisc`
    - Turns promiscuous mode on/off
  - `arp` (address resolution protocol)
    - `ifconfig [adapter] arp/-arp`
    - Turns arp on/off on the specified adapter

205.1 Basic Networking Configuration (Network Commands - ip)

- `ip`
  - Replaces the arp, ifconfig and route commands on modern Linux distributions (systemd)
- Equivalent command examples
  - `route add default gw 10.1.1.1` = `ip route add default via 10.1.1.1`

- `route del default` = `ip route del default`
- `route add -net 10.1.1.0 netmask 255.255.255.0 gw 10.1.1.1` = `ip route add 10.1.1.0/24 via 10.1.1.1`
- `route del -net 10.1.1.0 netmask 255.255.255.0 gw 10.1.1.1` = `ip route del 10.1.1.0/24 via 10.1.1.1`
- `route` = `ip route show`
- `ifconfig` = `ip addr show`
- `ifconfig eth0 promisc/-promisc` = `ip link set eth0 promisc on/off`
- `ifconfig eth0 apr/-arp` = `ip link set eth0 arp on/off`
- `ifconfig etho 10.1.1.120 netmask 255.255.255.0 broadcast 10.1.1.255` = `ip addr add 10.1.1.120 /24 broadcast 10.1.1.255 dev eth0`
- `arp` = `ip neigh show`
- `arp -i eth0 -d 10.1.1.120` = `ip neigh del 10.1.1.120 dev eth0`

205.1 Basic Networking Configuration (Network Commands - arp)

- `arp`
  - Address resolution protocol, part of the ISO network model (a seven layer model that handles how network devices communicate), arp is associated with level 2 in this model as it is mapped to MAC addresses (a unique identifier for any network device), kept in a 'table' on both systems with network adapters and routers/switches/load balancers
  - Alone, displays all active adapters and their arp configuration (Address, HWType, HWAddress, Flags mask, Iface)
  - `-i [interface]` • Specify the adapter interface with which to work
  - `-d [hostname]` • Deletes the arp information for the specified host
  - `-v` • Verbose in displaying arp information
- Entries can be added or deleted (rarely needed)
- Usually done because a system has had an adapter replaced (a new one would not require any manual intervention as the configuration and subsequent traffic generated in/out would create the appropriate arp entry), deleting the old arp entry allows a new value to be replaced sooner than the expiration default (usually 30sec to 1min)

205.1 Basic Networking Configuration (Network Commands - iwconfig and iwlist)

- NOTE: These commands are not covered in the videos, these are here for study awareness only and other than a general 'what command is used to do XYZ for wireless,' you will not see detailed configuration questions for wireless on the LPIC-2 exam

- `iwconfig`
    - Displays all the parameters of detected wireless interfaces
    - Default interface name is 'wlan#' where '#' is the number of the interface
    - Executed by itself will display all active adapters, but display only parameters for devices categorized as wireless (mode, frequency of connection, detected transmission power, RTS, power management)
- `iwlist`
    - Used to probe available (visible) wireless connection access points
    - `scan` • Option to scan for visible wireless access points and provides information about speed, connection and encryption type, etc

205.2 Advanced Network Configuration and Troubleshooting (Gather Information From Remote Systems - ping and nmap)

- `ping`
    - Commonly used to send a request to a remote system to discover if the remote system responds on the network address indicated (using ICMP requests - Internet Control Message Protocol)
    - `-c #` • Ping the remote system [#] of times
    - `-f` • Flood the remote system to determine any issues with packet loss/traffic handling (NOTE: Use with caution as this is often seen as malicious use of the utility)
    - `-i #` • Wait [#] of seconds between pings (default is one second)
    - `-n` • Numeric (IP) output only, no host names will be looked up
    - `-v` • verbose output
- `nmap`
    - Utility to determine what ports/services may be available on a remote system
    - Run alone, displays default availability information on the remote system (nmap version, up/down, count of closed ports, open port, open port state, service of port if known, summary of report)
        - NOTE: By default, ONLY TCP ports are scanned and reported on
    - `-sP [network range]` • Perform a basic 'ping' scan on all the IPs in the indicated range to determine which are available/respond to the request
    - `-sU` • Scan and report on UDP (User Datagram Protocol) ports/services

- `-sV` • Provide version information (when available or can be determined) on the port and services that are scanned and are open
- `-O` • Can be used to determine (of provide a "guess" of sorts) about the remote operating system in use
- `-p [range]` • Specify a range of ports to which to limit the scan
- `--iflist` • Show local network interfaces and the associated routing table

205.2 Advanced Network Configuration and Troubleshooting (View Network Activity - netstat, lsof and nc)

- `netstat`
  - Outputs network connections (from the localhost perspective), routing, interface stats, connections and multicast information
  - `-s` • Displays a summary of the localhost network information (rather than a remote scan), summarized by protocol type (messages received, sent, ICMP info)
  - `-i` • Restrict summary information to the active network interfaces only
  - `-r` • Restrict summary information to the routing table only
  - `-at` • Restrict summary information to TCP ports only
  - `-au` • Restrict summary information to UDP ports only
  - `-e` • Display more verbose information (`-ee and -eee` for even more)
  - `-l` • Display a list of all LISTENING sockets
  - `-n` • Do not attempt to translate to names
- `lsof`
  - Lists open files and the processes, user associated with them (included are the command run, PID, user, file desc, type, device (number), size, node and name of the file open)
  - `-n` • Numeric (IP) output only, no host names will be looked up
  - `-p` • Do not attempt to resolve port numbers to application/port names
- `nc`
  - Netcat, used for a wide variety of TCP/UDP information, can even be used to listen for connections as part of a test
  - `-l [port number]` • Starts listening for incoming connections on that port (provided it is available, if not, the command will fail to start listening)
  - `[host/server name] [port number]` • The 'client' side of the netcat listener, connects

to the IP/host/server of the remote system on the indicated port

- `-6` • Enable IPv6 connectivity

- `-k` • Allows the server to continue listening after a client disconnects

- `-u` • Use UDP (TCP is the default)

- `-w [timeout]` • Sets a timeout value for the connection to automatically close after connected

- `-z [hostname] [port range]` • functions like `netstat` in that it displays open ports on the indicated hostname

205.2 Advanced Network Configuration and Troubleshooting (Troubleshoot Network Issues - tcpdump)

- `tcpdump`

  - Allows a real time view/summary of ALL TCP connections on the local system (often referred to as a 'packet sniffer')

  - `-c [#]` • Limits the output of the command to the indicated number of packets

  - `-i [interface] [protocol] [port] [src/dest]` • Limit the output of the command to ONLY the data related to the indicated interface and/or (optionally) the indicated protocol (TCP or UDP), port or source/destination

  - `-w [filename]` • Write the output of the command to a file for later viewing/processing

  - `-r [filename]` • Reads the `tcpdump` content from an earlier run sent to the indicated file

    - Used most commonly with other utilities like `cut`, `wc`, `grep`, `sed`, `awk` or other processing commands to review specific data/IP/traffic information

205.3 Troubleshooting Network Issues (Configuration Files - resolv.conf, hosts, nsswitch.conf and hostname)

- `/etc/resolv.conf`

  - Contains the local (client) DNS configuration (nameservers, domains)

  - Multiple nameservers can be defined, one per line (up to three are generally supported, more may be supported depending on the distribution version)

    - Names 8.8.8.8 (for example, would define the Google Public DNS server at IP 8.8.8.8 as one of the nameservers the client can use for name resolution)

      - Nameservers are used in the order they appear in the file

  - `search [domain]` • Tells the local (client) DNS to attempt to resolve any 'short' hostnames by assuming the addition of the domain indicated

        - `ping test`, for example, would attempt to ping hosts name 'test' and 'test.[domain]' as indicated in the 'search' line

        - Limited to SIX domains using the 'search' directive or 256 characters on each line

    - `options [timeout]:` • Amount of time to wait for DNS response before timing out

    - `options [attempts]:` • Number of attempts to query a name server before indicating failure

- `/etc/hosts`

    - Local file that contains local host name/IP definition maps, generally is the first place a system will look for system names when lookups are necessary

    - **General format** • `[IPv4/IPv6 Address] [short hostname] [FQDN/alias]`

    - Most common entry is for the 'localhost'/'localhost.localdomain' at IP 127.0.0.1

- `/etc/nsswitch.con`

    - Determines the order that name resolution will follow, that can include db, files, nis, nisplus, dns

        - `hosts: files dns`, for example, would indicate that the client system name resolution will check for local files like `/etc/hosts` and associated values and then nameservers as defined in the `/etc/resolv.confi` file

- `hostname`

    - Command to display or temporarily set the local host name

    - By itself, it displays the current hostname

    - `[name]` • As an option, temporarily (i.e. until system restart or the `hostname command` is run again) changes the hostname of the system

        - NOTE: Use caution in changing the hostname as some running processes may be linked to the system's configured hostname when they were started

    - In Debian based distributions, also exists as a file in `/etc/hostname` containing the hostname itself

    - In other distributions, is defined within the configuration file `/etc/sysconfig/network`

205.3 Troubleshooting Network Issues (Configuration Files - Network Device Configuration Files)

- `/etc/network/interfaces`

    - From the file header, this describes the network interfaces that are available on your system (NOTE: This is on Debian-based distributions)

    - Typical entries are loopback, eth[#] and associated options
    - Each interface can be listed as 'auto' configured (automatic on boot, like DHCP) or 'static' and other directives are provided for address, netmask and gateway in order to be read on boot

- `/etc/sysconfig/network-scripts`
  - Directory that contains the start up and termination scripts associated with each network device on the system (NOTE: this is on Red Hat and CentOS-based distributions)
  - `ifcfg-eth0`
    - This file would contain the device configuration for the interface called 'eth0' on the system
    - Configuration items of note are DEVICE, HWADDR, TYPE, UUID, ONBOOT, BOOTPROTO
      - ONBOOT, BOOTPROTO and DEVICE are required, whereas the others are optional
      - Example:

        -
          ```
          DEVICE=eth0
           HWADDR=09:01:33:A2:A4:EE
           TYPE=ethernet
           UUID=a2dashj8221-2gv0-8712-0987-de9878992fece2
           ONBOOT=yes
           BOOTPROTO=dhcp
          ```
    - Adding static address would look like the following example

      -
        ```
        DEVICE=eth0
         IPADDR=10.0.2.111
         NETMASK=255.255.255.0
         BROADCAST=10.0.2.255
         ONBOOT=yes
         BOOTPROTO=none
        ```

205.3 Troubleshooting Network Issues (Log Files)

- `/var/log/syslog`
  - The main log file on Debian-based distributions
- `/var/log/messages`
  - The main log file on Red Hat- or CentOS-based distributions
- `dmesg`
  - Shows the 'kernel ring buffer', or messages from the kernel during boot, normal runtime and shutdown/reboot

    - Often used with other utilities to parse or pull specific information from the file or watch system events in real time
        - `dmesg | tail -f`, for example, displays the full contents and then continue to display events as they come in and are written to the logs

Topic 206 - System Maintenance

206.1 Make and Install Programs from Source (Unpack, Configure, Compile and Install)

- NOTE: Distribution specific source package downloading is NOT a topic covered on the LPIC-2 exam
- **For CentOS/Red Hat systems** • Install the "Development Tools" via `groupinstall`
- **For Debian/Ubuntu systems** • Install the development tools package "build-essential"
- **Sample package** • "curl" version 7.32.0 (URL: http://curl.haxx.se/download/curl-7.32.0.tar.gz)
- Unpacking the source code
    - Create a local directory (i.e. "compile")
    - Source can be in gzip, bzip2, or 7z format (compressed) or just tar
        - `tar zxvf curl-7.32.0.tar.gz` • gunzip and untar the source files and directories for the indicated curl version)
- `configure`
    - Pre-build application that reads the required applications, source and library links that need to be installed and available for the compilation to succeed
    - Errors will indicate that something is not installed, will stop at the FIRST error
        - Can be resolved by installing the package, application or source that is referred to in the error
        - Re-run the configuration command
- `make [target]`
    - Like in the kernel compile, the "make" command can be run to compile (without target listed, will run the default compile target)
        - `sudo make install` would install the successfully compiled "curl" utility in our example in the default directory locations specified by the make

206.2 Backup Operations (Standard Tools - dd, tar and rsync)

- `dd`

  - Backs up entire devices to a file or to another partition/device
    - Note that the output device would need to be at least as big as the input device
  - `dd if=/dev/xvdj1 of=/dev/xvdj2` would take the entire partition '/dev/xvdf1' and back it up EXACTLY AS IS to `/dev/xvdf2`
  - `dd if=/dev/xvdj1 of=/mnt/today.img` would take the entire partition '/dev/xvdf1' and back it up to a file image called `/mnt/today.img`
    - Commonly used to back up CD/DVD discs to an ISO image (`dd if=/dev/sr0 of=/mnt/dvd.iso`)
- `tar`
  - Tape archive
  - Default tape device names, if they exist, would be */dev/st+* and / or */dev/nst+*
    - If asked which command can manipulate a tape device, the `mt` command can be used to skip, rewind, erase or display information about the indicated tape device (this will be the extent of actual tape device questions you may see)
  - `-c` • Create the indicated tar file
  - `-f` • Filename to create
  - `-t` • View the contents of a tar file
  - `-x` • Extract the tar file
  - `-v` • Display verbose output for the command
  - `-A` • Append to the existing tar file
  - `-W` • Attempt to verify the contents after writing
  - `-z` • Compress the tar file during creation with the gzip command
  - Examples
    - `tar -cvf mytar.tar /etc/+` creates a tar file called "mytar", containing the contents of the `/etc`/ directory
    - `tar -tvf mytar.tar` shows the contents -- files and paths -- contained in the indicate tar file
    - `tar -cvzf mytar.tar.gz /home/user/+ c`reates a gzip compressed tar file called "mytar.tar.gz" containing the contents of the `/home/user/` directory
    - `tar -zxvf mytar.tar.gz` uncompresses the files and directories in the tar file, preserving relative paths

- `rsync`
  - A different type of "backup" tool that provides features more suited to traditional backup, particularly to or from remote systems
  - Allows a certain amount of synchronization based on file changes between backups and provides a method of easily including/excluding files and directories in a list
  - Possible to use as a daemon offering remote connections (out of scope for the LPIC-2 exaxm)
  - `-a` • Archives the indicated files/directories to a local or remote location
  - `-e` • Execute the indicated command as part of the synch (commonly ssh)
  - `-q` • Quiet the output of non-error messages
  - `-r` • Recursive (recurse the synch into directories)
  - `-b` • Make a backup
    - `--suffix=SUFFIX` • The type of backup (extension) to make
  - `-u` • Update the synch, copying ONLY files that have changed or been added since last copy
  - `-l` • Preserve and copy symlinks
  - `-H` • Preserve hard links
  - `-n` • Dry run (no changes will be made, testing purposes)
  - `-v` • Verbose messaging
  - `-z` • Compress during transfer
  - Examples
    - `rsync [options] [source] [destination]` (basic syntax)
    - `rsync -zvh local.tar /mnt/backup/` (synchronize/copy a single file in the local directory to a different location, compressing the file during transfer)
    - `rsync -avz /mnt/dir1 /mnt/dir2/` (transfer one local directory to another in archive mode)
    - `rsync -avz /mnt/dir1 user@10.0.0.101:/home/user` (transfer the local /mnt/dir1 contents, in archive mode, to a remote system at IP 10.0.0.101 as the user 'user' and stored in the /home/user directory)
    - `rsyncy -avz user@10.0.0.101:/home/user/ /mnt/dir1` (the reverse of the above example, synchronize the remote contents, locally)

206.3 Notify Users on System-Related Issues (Messages with issue, wall and motd)

- `/etc/issue`
  - File that allows the display of a custom message as part of the login prompt
  - Common options
    - `\d` • Displays the current date
    - `\s` • System name
    - `\m` • Displays the results of the `uname -m` command
    - `\n` • Node name (equivalent of `uname -n`)
    - `\r` • OS release number (`uname -r`)
    - `\u` • Display number of users logged in
    - `\v` • Version of the OS
    - **text** • Can be entered, with line breaks and spacing, for custom messaging
  - Only displayed when a user logs in via the command line login LOCALLY
- `/etc/issue.net`
  - Equivalent of the 'issue.net', but displayed for remote network logins
  - Special characters are not valid in this configuration file, text only
    - NOTE: SSH servers will not display by default, a change to `/etc/ssh/sshd_config` is needed
      - # Banner none (change)
      - Banner /etc/issue.net
      - Restart sshd daemon
- `/etc/motd`
  - Post login message (message of the day)
  - Typically used to welcome, warn or inform the users of potential system issues, changes or upcoming information
  - By default, is generally empty
  - Text entry, no formatting, no special characters or command substitution
  - Will be displayed by default if not empty
- `wall`

- Broadcast a message to all users that are LOGGED IN to the system
- Text message only
- Wall messages in any terminal can be turned on or off by a user
  - `mesg n` • Disables
  - `mesg y` • Enables
  - `mesg` • Displays current value
- `-n` • Disable banner display in broadcast
- Message is limited to 20 lines of text
- Called by `shutdown` command if a quoted text message is included in the line
  - `shutdown +30 "Shutdown coming..."` (will shut the system down in 30 minutes, but broadcasts the message immediately)