Managing Libraries In Linux

- Configure a system wide library path**: /etc/ld.so.conf**
    - File includes include ld.so.conf.d/*.conf
        - Tells system to load all files in /etc/ld.so.conf.d that have extensions that end in .conf.
        - ldconfig – must run this program after updating library path.
- /lib and /usr/lib are always included in the libraries even if they are not located in the ld.so.conf file.
- By default library path is /etc/ld.so.conf. However, you can change the library path in your environment by updating the **LD_LIBRARY_PATH** environment variable. Export **LD_LIBRARY_PATH=/usr/local/mylib:/opt/mylib**
    - Adds two new locations to the library search path
    - Does not require ldconfig to be ran for change to take place. Adding libraries to the main ld.so.conf files always do

Correcting Library Path Problems

- Most common problems are programs looking in wrong locations for library and library names having minor .x.x.x differences.
    - Fix by creating symbolic links to expected locations and new file names
        - After creating symbolic link run ldconfig to apply library changes.
        - Be sure to add the location of the symbolic links in the library paths so they can be searched. Or use LD_LIBRARY_PATH environment variable to set the location

Library Management Commands

- **Ldd** path-to-program: shows library dependencies on a certain program.
- **Ldconfig** –updates cache and links used to locate libraries. (Reads changes
    - Flags:
        - -N :Does not rebuild cache but updates symbolic links
        - -n Processes only specified directories
        - -X update cache but not links
        - -C new cache file
        - -r New root dir
        - -p display current cache
- from /etc/ld.so.conf) this is NOT needed when you change the LD_LIBRARY_PATH environment variable.

Getting Information Out Of the Kernel

Command: **uname**

- **uname –n or –nodename** displays the system network hostname/node name
- **uname –s or –kernel-name** displays the name of the Kernel
- **uname –a or –all** – Prints all information
- **uname –o or –operating-system** displays the operating system
- **uname –m or –machine** displays information about your machine/cpu
- **uname – v or –kernel-version** – displays your Linux kernel version, but only holds kernel build and release time. NOT the actual kernel version.
- **uname – r or –kernel-release** Displays the kernel release number.
- **Uname –I or –hardware-platform** – Display information about your current hardware platform, can be unknown especially in cloud based environments.

PS – Displays information about a selection of the active system processes.

Ps is used to display Linux system processes. It is a very extensive command and its full version can have around 80 columns of display data. By default PS command truncates the data to fit your console window. Display all columns you will want to dump the information to a text file. By default ps only displays process that were run from its own terminal. -Ae will display all processes on a system. -u displays processes given by a specified user, H -F group processes and use indentation to show the hierarchy of relationships between processes. Ps w > all.txt tells ps not to truncate to system and output to file. Typing PS with no options/flags (default view) you will only see processes that were **RUN FROM IT'S OWN TERMINAL**.

  - **You can change some default settings in PS by altering the PS_PERSONALITY environment variables.**
  - Display all processes on a system
    - -e displays all processes
    - -x display process user owner
  - –U or –User, -i Display processes by a specific user
  - –p select by process id
  - –u show user list
  - ps aux BSD style show users list and all processes
  - –-forest display forest
  - h used to print process tree.
- PS can use a combination of bsd and other ps version such as GNU UNIX and BSD.
- PID (process id) Id of the process can be use to search or kill the process.
- Username – Username of the user who runs the program.

- Parent Process ID – Process parent, the parent process is the process that launches the child process
- TTY Teletype code that identifies the terminal not all processes have these.
- CPU time - Represents the total CPU time used
- %CPU – Represents the percentage of cpu time the process is using when it's running.
- CPU priority – Priority set by nice. Lower priority the more priority the process has with the CPU.
- %mem – Same concept as CPU percentage of memory the process is using
- Command – the command that launched the process if any.

TOP

- -d specifies delay between updates
- -p lists of to 20 specific PID's,
- -n display certain number of updates then quit,
- -b batch mode.

Commands while running top:

- k - kills processes,
- q-quits processes,
- r - change process priority,
- s - change update rate,
- P- sort by cpu usage,
- M- sort by memory usage.
- M – displays memory usage.

Top will also show uptime, and memory info. and load average (all the same as w)

Jobs – show running processes for that specific shell instance

Fg push a program from the background to the foreground

Bg push a program from the foreground to the background

& use after a program name to launch an instance in the background i.e nano test &