Filename: comptia-linuxxk0004-8-4-1-troubleshooting_services
Show Name: CompTIA Linux+ (XK0-004)
Topic: Managing System Components
Episode Name: Troubleshooting Services
Description: In this episode, Zach and Don switch gears and focus on what to do when things go wrong. They highlight some of the conditions where a process may execute improperly and then demonstrate some of the tools you can use to troubleshoot and correct the misbehaving process. They wrap up by showing how to terminate an unresponsive process.

Troubleshooting Services

[?] What are some of the common problems we can have with processes under Linux?

- Process Hangs
    - Freezes
    - May still consume resources
    - Holds files open
- Process Terminates
    - Incomplete execution
    - May corrupt data
- Resource starvation
    - Other processes are consuming too many resources

[?] How does a normal process behave?

- 5-State process life cycle
    1. Running
        - The process is executing
    2. Interruptible Sleep
        - The process is waiting on a resource
        - Can wake at any time
    3. Uninterruptible Sleep
        - The process is waiting on a resource
        - Will only wake when the resource is available
    4. Zombie
        - The process has completed, but the parent is holding it open
    5. Stopped
        - The process has terminated
        - Successfully or unsuccessfully

[?] What is the best way to view the processes as they are running?

- `ps`
    - Displays the process table
    - BSD Style options
        - `a` - List all user-triggered processes
        - `u` - List user name alongside process
        - `x` - Include processes without terminals
    - Unix Style options

        - `-e` - List all processes
        - `-U <user_name>` - Display processes for a particular user

[?] How would we know if a process is misbehaving?

- Resource hog
    - `top`
    - Displays process performance data
    - Shortcut Keys
        - `M` - Sort processes by memory usage
        - `P` - Sort processes by CPU usage
        - `u` - Display processes belonging to the user specified at the prompt
        - `k` - Terminate the process for which you specify the PID
        - `q` - Exit the process list
- Slow boot
    - `systemd-analyze blame`
    - Displays processes that took the most time to boot
- Holding files open
    - `lsof`
    - Lists files opened by all processes
    - `lsof | grep "file\.txt"`

[?] Are there other ways to end a process other than the top command?

- Always try to exit gracefully
    - Process might be in the background
    - `Ctrl-Z`
        - Temporarily pauses the job
    - `jobs`
    - `bg #` sends job to background
    - `fg #` brings job to foreground
    - `%<job_number>`
    - `&`
- Try adjusting the process priority
    - Nice value
    - -20 through +19
        - Lower value means higher priority
        - +10 is the default
    - Set priorities when starting
        - `nice -n 5 mysqld`
    - Change priorities while running
        - `renice -n <pid>`

[?] If all that fails, is it Hulk smash time?

- `pgrep`
    - Quickly find the process ID
    - `pgrep httpd`
- `kill`

    - Ends a process by PID
    - `kill 1004`

- `killall`
    - Ends a process by name
    - `killall firefox`