3. Student Activity

Student Activity: Practicing Foreground and Background Processes in Linux

Welcome to the hands-on activity session! In this activity, you will practice managing foreground and background processes in Linux. Follow the steps below to reinforce your understanding of these concepts. Make sure you have access to a Linux terminal to perform these exercises.

---

1. Understanding Foreground Processes

Example 1: Viewing a File

1. Open your terminal.
2. Run the following command to view a file in the foreground:

   ```
   $ cat /etc/passwd
   ```

   Observe how the terminal is occupied until the command finishes displaying the file contents.

Example 2: Running a Command

1. Run a command that takes some time to complete:

   ```
   $ find / -name "*.conf"
   ```

   Notice that you cannot use the terminal for other tasks until this command completes.

Example 3: Using a Text Editor

1. Open a text editor in the terminal:

   ```
   $ nano myfile.txt
   ```

   The terminal is occupied by the text editor until you exit it.

---

2. Working with Background Processes

Example 1: Running a Command in the Background

1. Run a command in the background using the ampersand (`&`):

   ```
   $ sleep 30 &
   ```

   Observe that you can continue using the terminal while the `sleep` command runs in the background.

Example 2: Background File Copy

1. Start copying a large file in the background:

   ```
   $ cp largefile.txt /tmp/ &
   ```

   You can perform other tasks in the terminal while the file is being copied.

Example 3: Background Download

1. Use `wget` to download a file in the background:

   ```
   $ wget http://example.com/largefile.zip &
   ```

   The download will continue in the background, allowing you to use the terminal for other commands.

---

3. Managing Foreground and Background Processes

Example 1: Moving a Foreground Process to the Background

1. Start a command in the foreground:

   ```
   $ tail -f /var/log/syslog
   ```

2. Pause the process using `Ctrl + Z`.
3. Move it to the background:

```
$ bg
```

Example 2: Bringing a Background Process to the Foreground

1. List background jobs:

```
$ jobs
```

2. Bring a specific job to the foreground using its job number:

```
$ fg %1
```

Example 3: Terminating a Background Process

1. Identify the PID of a background process using `ps`:

```
$ ps
```

2. Kill the process using its PID:

```
$ kill <PID>
```

---

4. Checking and Killing Processes

Example 1: Listing Processes

1. List all processes running in your terminal:

```
$ ps
```

2. List all processes on the system:

```
$ ps aux
```

Example 2: Killing a Process

1. Identify a process to kill using `ps`:

   ```
   $ ps aux | grep sleep
   ```

2. Kill the process using its PID:

   ```
   $ kill <PID>
   ```

Example 3: Forcefully Killing a Process

1. If a process does not terminate, forcefully kill it:

   ```
   $ kill -9 <PID>
   ```

---

5. Creating and Running a Shell Script

Example 1: Writing a Simple Script

1. Open a text editor and write the following script:

   ```
   #!/bin/bash
   echo "Starting script..."
   sleep 3
   echo "Script completed."
   ```

2. Save it as `simple_script.sh`.

Example 2: Making the Script Executable

1. Make the script executable:

   ```
   $ chmod +x simple_script.sh
   ```

Example 3: Running the Script

1. Execute the script:

   ```
   $ ./simple_script.sh
   ```

   Observe the output and how the script runs.

Conclusion

By completing these exercises, you should have a better understanding of how to manage foreground and background processes in Linux. Practice these commands regularly to become proficient in process management and shell scripting. If you have any questions or need further clarification, feel free to ask!