**# Process Management in Linux**

## Introduction to Process Management

A process is an instance of a running program. Linux provides multiple utilities to monitor, manage, and control processes effectively. Each process has a unique \*\*Process ID (PID)\*\* and belongs to a parent process.

## Index of Commands Covered

### Viewing Processes

- `ps aux` – View all running processes

- `ps -u username` – View processes for a specific user

- `ps -C processname` – Show a process by name

- `pgrep processname` – Find a process by name and return its PID

- `pidof processname` – Find the PID of a running program

### Managing Processes

- `kill PID` – Terminate a process by PID

- `pkill processname` – Terminate a process by name

- `kill -9 PID` – Force kill a process

- `pkill -9 processname` – Kill all instances of a process

- `kill -STOP PID` – Stop a running process

- `kill -CONT PID` – Resume a stopped process

- `renice -n 10 -p PID` – Lower priority of a process

- `renice -n -5 -p PID` – Increase priority of a process (requires root)

### Background & Foreground Processes

- `command &` – Run a command in the background

- `jobs` – List background jobs

- `fg %jobnumber` – Bring a job to the foreground

- `Ctrl + Z` – Suspend a running process

- `bg %jobnumber` – Resume a suspended process in the background

### Monitoring System Processes

- `top` – Interactive process viewer

- `htop` – User-friendly process viewer (requires installation)

- `nice -n 10 command` – Run a command with a specific priority

- `renice -n -5 -p PID` – Change priority of an existing process

### Daemon Process Management

- `systemctl list-units --type=service` – List all system daemons

- `systemctl start service-name` – Start a daemon/service

- `systemctl stop service-name` – Stop a daemon/service

- `systemctl enable service-name` – Enable a service at startup

## Viewing Process Details

### Using `ps`

Show processes for a specific user:

```bash

ps -u username

```

Show a process by name:

```bash

ps -C processname

```

### Using `pgrep`

Find a process by name and return its PID:

```bash

pgrep processname

```

### Using `pidof`

Find the PID of a running program:

```bash

pidof processname

```

## Managing Processes

### Killing Processes

To terminate a process by PID:

```bash

kill PID

```

To terminate using process name:

```bash

pkill processname

```

Force kill a process:

```bash

kill -9 PID

```

Kill all instances of a process:

```bash

pkill -9 processname

```

### Stopping & Resuming Processes

Stop a running process:

```bash

kill -STOP PID

```

Resume a stopped process:

```bash

kill -CONT PID

```

### Changing Process Priority

View process priorities:

```bash

top # Look at the NI column

```

Change priority of a running process:

```bash

renice -n 10 -p PID # Lower priority (positive values)

renice -n -5 -p PID # Higher priority (negative values, root required)

```

### Running Processes in the Background

Run a command in the background:

```bash

command &

```

List background jobs:

```bash

jobs

```

Bring a job to the foreground:

```bash

fg %jobnumber

```

Send a running process to the background:

```bash

Ctrl + Z # Suspend process

bg %jobnumber # Resume in background

```

## Monitoring System Processes

### Using `top`

Interactive process viewer:

- Press `k` and enter a PID to kill a process.

- Press `r` to renice a process.

- Press `q` to quit.

### Using `htop`

A user-friendly alternative to `top`:

```bash

htop

```

Allows mouse-based interaction for process management.

### Using `nice` & `renice`

Run a command with a specific priority:

```bash

nice -n 10 command

```

Change the priority of an existing process:

```bash

renice -n -5 -p PID

```

## Daemon Processes

Daemon processes run in the background without user intervention.

List all system daemons:

```bash

systemctl list-units --type=service

```

Start a daemon:

```bash

systemctl start service-name

```

Stop a daemon:

```bash

systemctl stop service-name

```

Enable a service at startup:

```bash

systemctl enable service-name

```

## Conclusion

Process management is crucial for system performance and stability. By using tools like `ps`, `top`, `htop`, `kill`, and `nice`, you can efficiently control and monitor Linux processes.