Chapter 5: Process Management

1. What Is a Process?

Definition (Interview-Oriented)

A process is a **running instance of a program** in memory.

Example:

- `nginx` binary on disk → program
- `nginx` running → process

Interview-ready line:

A process is a program in execution with its own memory, PID, and resources.

2. Process Lifecycle (Important Concept)

A process goes through these stages:

1. Created
2. Ready
3. Running
4. Waiting (sleep)
5. Terminated

Linux kernel manages this lifecycle using a scheduler.

Interview insight:

Process scheduling is handled by the kernel, not the user.

3. PID (Process ID)

Definition

PID is a **unique numeric identifier** assigned to each process.

Example:

```
ps -ef
```

Interview note:

PID 1 is always the init/systemd process.

---

4. Parent and Child Processes

Explanation

- Every process (except PID 1) has a parent
- Parent creates child processes

Example:

- Shell is parent
- Commands you run are child processes

Interview line:

Linux follows a parent-child process hierarchy.

---

5. Zombie Process (VERY IMPORTANT)

Definition

A zombie process is a process that has **finished execution but still has an entry in the process table**.

Why Zombie Processes Occur

- Child process exits
- Parent does not collect exit status
- Kernel keeps the entry

Zombie processes:

- Do not consume CPU
- Do not consume memory
- Still consume PID slots

Interview-ready explanation:

Zombie processes exist because the parent failed to reap the child.

---

6. Orphan vs Zombie Process

Orphan Process

- Parent dies before child
- Child is adopted by PID 1 (systemd)
- Cleaned up automatically

Zombie Process

- Child dies
- Parent still alive but not handling it
- Remains until parent exits

Comparison Table

Feature	Orphan	Zombie
Parent alive	No	Yes
Consumes resources	No	No
Auto cleaned	Yes	No

---

7. Context Switching

Definition

Context switching is the process of saving the state of one process and loading another.

Why It Happens

- CPU can run only one process per core at a time
- Kernel switches between processes rapidly

Interview insight:

Excessive context switching reduces performance.

---

8. nice and renice

Definition

`nice` **defines process priority.**
Lower value = higher priority.

Range:

- `-20` (highest)
- `19` (lowest)

Example

```
nice -n 10 command
renice -5 -p 1234
```

Interview line:

nice controls CPU scheduling priority, not memory or disk.

---

9. Finding High CPU Processes

Common Commands

```
top
ps aux --sort=-%cpu
```

Interview Approach

When CPU is high:

1. Identify process
2. Check logs
3. Check application behavior
4. Decide whether to kill or restart

---

10. Killing a Process Safely

Signals Explained

- SIGTERM (15): Graceful stop
- SIGKILL (9): Force stop

Commands

```
kill PID
kill -9 PID
```

Interview insight:

Always try SIGTERM before SIGKILL.

---

11. ulimit

Definition

`ulimit` **controls resource limits for users and processes.**

Examples:

- Max open files
- Max processes
- Max memory

Example

```
ulimit -a
```

Interview explanation:

ulimit prevents a single user or process from exhausting system resources.

---

12. OOM Killer (Advanced but Important)

Definition

OOM (Out Of Memory) Killer is a kernel mechanism that **terminates processes when system memory is exhausted**.

How It Works

1. System runs out of memory
2. Kernel selects a process
3. Process is killed to free memory

Interview insight:

OOM Killer protects the system from complete crash.

---

13. Real-Life Production Scenarios

Scenario 1: High CPU Usage

- Use `top` or `ps`
- Identify PID
- Check application logs
- Adjust priority or restart

Scenario 2: Zombie Processes Increasing

- Identify parent process
- Restart or fix parent application

Scenario 3: Application Killed Automatically

- Check OOM logs
- Increase memory or optimize app

Chapter 5: Interview Takeaways

After this chapter, you should be able to:

- Explain process lifecycle clearly
- Differentiate zombie vs orphan processes
- Explain context switching
- Use nice/renice properly
- Debug high CPU and memory issues
- Explain OOM Killer confidently