

Process Management, Scheduling, and Package Management in Linux

PART A: PROCESS MANAGEMENT AND SCHEDULING

1. What is a Process?

A **process** is a **program in execution**.

Example: - ls command → program - When executed → process

A program is **passive** (stored on disk), while a process is **active** (running in memory).

2. Program vs Process (Simple Comparison)

Program	Process
Stored on disk	Resides in memory
Static	Dynamic
No PID	Has PID
Not scheduled	Scheduled by CPU

3. Process Lifecycle

A process goes through multiple states:

New → Ready → Running → Waiting → Terminated

State Explanation

- **New:** Process is created
 - **Ready:** Waiting for CPU
 - **Running:** Executing on CPU
 - **Waiting:** Waiting for I/O
 - **Terminated:** Execution completed
-

4. Process Attributes

Each process has metadata known as **process attributes**.

Important Attributes

Attribute	Description
PID	Process ID
PPID	Parent Process ID
UID	User ID
State	Current process state
Priority	Scheduling priority
Nice value	User-defined priority
CPU usage	CPU time consumed
Memory usage	RAM usage

5. PID and PPID

- **PID**: Unique identifier for a process
- **PPID**: PID of parent process

Example:

```
ps -ef
```

6. Process Creation in Linux

Processes are created using: - `fork()` – creates a child process - `exec()` – replaces process image

Parent-child relationship:

```
Parent (bash)
└─ Child (ls)
```

7. Process States in Linux

Check process states:

```
ps aux
```

State	Meaning
-------	---------

State	Meaning
R	Running
S	Sleeping
D	Uninterruptible sleep
Z	Zombie
T	Stopped

8. Zombie and Orphan Processes

Zombie Process

- Process completed
- Entry remains in process table
- Parent did not collect exit status

Orphan Process

- Parent terminated
 - Child adopted by PID 1 (init/systemd)
-

9. Process Control Commands

Command	Purpose
ps	View processes
top	Real-time process view
htop	Enhanced top
kill	Send signal
nice	Set priority
renice	Change priority

10. Signals in Linux

Signals are used to control processes.

Signal	Meaning
SIGTERM (15)	Graceful termination
SIGKILL (9)	Force kill
SIGSTOP	Pause
SIGCONT	Resume

Example:

```
kill -9 1234
```

PART B: PROCESS SCHEDULING AND PRIORITIZATION

11. What is Process Scheduling?

Process scheduling decides: - Which process gets CPU - For how long

Linux uses **preemptive multitasking**.

12. CPU Scheduling Basics

- Single CPU → one process at a time
- Multiple CPUs → parallel execution
- Scheduler switches processes using **context switching**

13. Linux Scheduler (CFS)

Linux uses **CFS – Completely Fair Scheduler**.

Key Idea

- All processes get **fair CPU time**
- Virtual runtime (vruntime) used

14. Priority and Nice Value

Priority

- Kernel internal value

Nice Value

- User-space priority
- Range: -20 (highest) to +19 (lowest)

15. Setting Priority Using nice

Start a process with priority:

```
nice -n 10 sleep 1000
```

16. Changing Priority Using renice

```
renice -5 -p 1234
```

17. Real-Time Scheduling Policies

Policy	Description
SCHED_FIFO	First In First Out
SCHED_RR	Round Robin
SCHED_OTHER	Default

Used in: - Automotive - Robotics - Embedded systems

18. Viewing Scheduling Information

```
chrt -p 1234
```

PART C: PACKAGE MANAGEMENT

19. What is Package Management?

Package management handles: - Software installation - Dependency resolution - Updates and removal

20. What is a Package?

A **package** is a compressed archive containing: - Binaries - Libraries - Config files - Metadata

Examples: - .deb (Debian/Ubuntu) - .rpm (RHEL/CentOS)

21. Package Management Systems

Debian-Based Systems

- Ubuntu
- Debian

Tool: - apt

Red Hat-Based Systems

- RHEL
- CentOS
- Rocky Linux

Tools: - yum - dnf

22. apt Package Manager (Debian/Ubuntu)

Update repository info:

```
sudo apt update
```

Install package:

```
sudo apt install nginx
```

Remove package:

```
sudo apt remove nginx
```

Upgrade system:

```
sudo apt upgrade
```

23. yum / dnf Package Manager (RHEL/CentOS)

Install package:

```
sudo yum install httpd
```

Remove package:

```
sudo yum remove httpd
```

Update system:

```
sudo yum update
```

24. Package Queries

List installed packages:

```
apt list --installed  
yum list installed
```

Search package:

```
apt search vim  
yum search vim
```

25. Package Installation from Local File

Debian

```
sudo dpkg -i file.deb  
sudo apt -f install
```

RPM

```
sudo rpm -ivh file.rpm
```

26. Dependency Management

- apt and yum automatically resolve dependencies
 - rpm and dpkg do **not** resolve dependencies
-

27. Repositories

Repositories are servers that store packages.

Config locations:

```
/etc/apt/sources.list  
/etc/yum.repos.d/
```

28. Best Practices

- Always update repo before install
- Avoid force killing processes

- Use nice carefully
 - Prefer official repositories
-

29. Interview Quick Notes

- Process = program in execution
 - PID uniquely identifies process
 - CFS ensures fairness
 - Nice controls priority
 - apt/yum manage software
-

30. Hands-on Practice Tasks

1. Start a background process
 2. Change its nice value
 3. Kill it gracefully
 4. Install and remove a package
-