# Chapter 6: Memory Management

## 1. Why Memory Management Is Important in Linux

### Interview Context

Linux servers often run:

- Multiple applications
- Background services
- Containers
- CI/CD agents

Memory is a **finite resource**.
Linux memory management ensures:

- Fair usage
- High performance
- System stability
- No single process crashes the system

Interview insight:

**Most real production issues are related to memory misuse, not CPU.**

## 2. Physical Memory vs Virtual Memory

### Physical Memory (RAM)

- Actual hardware memory
- Fast access
- Limited size

### Virtual Memory

**Definition**

Virtual memory is an abstraction that allows processes to use **more memory than physically available**.

Linux achieves this using:

- Paging
- Swap
- Address translation

Interview-ready line:

**Virtual memory gives each process an isolated, continuous memory space.**

---

# 3. How Linux Uses Memory (VERY IMPORTANT)

**Linux divides memory into several logical parts:**

- **Used memory**
- **Free memory**
- **Buffers**
- **Cache**

---

## Key Concept (Common Interview Trap)

**Free memory being low is NOT a problem in Linux.**

**Linux uses free memory aggressively for caching to improve performance.**

---

# 4. Free, Used, Buffers, Cache Explained

## Used

- Memory actively used by processes

### Free

- Completely unused memory

### Buffers

- Memory used for block device I/O metadata

### Cache

- Memory used to cache file contents

Interview explanation:

**Linux prefers using RAM for cache rather than leaving it idle.**

---

### Command to View Memory

```
free -h
```

Example output:

```
total   used   free   shared  buff/cache  available
```

Interview note:

**“Available” is the most important column.**

---

# 5. What Is Paging?

### Definition

**Paging is a memory management technique where memory is divided into fixed-size pages.**

Linux:

- Moves inactive pages to swap
- Keeps active pages in RAM

Interview insight:

**Paging allows efficient memory utilization without fragmentation.**

---

# 6. Swap Memory (Revisited, Deeper)

### Definition

**Swap is disk space used to temporarily store inactive memory pages.**

---

### Why Swap Exists

- Prevents sudden application crashes
- Provides memory breathing room
- Handles temporary spikes

---

### Downsides of Swap

- Disk is much slower than RAM
- Excessive swapping causes system slowness

Interview-ready line:

**Swap improves stability but hurts performance if overused.**

---

### Check Swap Usage

```
swapon --show
free -h
```

---

# 7. Memory Overcommit

### Definition

**Linux allows allocating more memory than physically available.**

### Why This Is Allowed

- Most processes don’t use all allocated memory
- Improves performance and flexibility

Interview insight:

**Overcommit increases efficiency but can trigger OOM Killer if misused.**

# 8. What Is OOM Killer? (Detailed)

### Definition

OOM (Out Of Memory) Killer is a kernel mechanism that **terminates processes to recover memory when the system is critically low on RAM**.

### How OOM Killer Decides

- Process memory usage
- Process priority
- System importance

Kernel kills the “least valuable” process.

### How to Detect OOM Events

```
dmesg | grep -i oom
journalctl -k | grep -i oom
```

Interview explanation:

**OOM Killer prevents total system freeze by sacrificing a process.**

# 9. Memory Leak (Very Common Interview Topic)

### Definition

A memory leak occurs when a process:

- Allocates memory
- Does not release it
- Memory usage grows over time

---

### Real-Life Example

- Java application
- Node.js service
- Long-running API server

Symptoms:

- Increasing memory usage
- Frequent OOM kills
- Service crashes

---

### Debugging Approach (Interview Gold)

1. Monitor memory over time
2. Identify growing process
3. Restart service temporarily
4. Fix application-level issue

Interview line:

**Linux exposes the symptom; application causes the leak.**

---

# 10. Checking Memory Usage (Commands)

### System-Level

```
free -h
vmstat
```

### Process-Level

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

### Per-Process Detailed View

```
cat /proc/PID/status
```

Interview note:

**/proc provides real-time kernel data.**

## 11. Page Cache vs Buffer Cache (Advanced but Asked)

### Page Cache

- Caches file contents
- Improves file read performance

### Buffer Cache

- Caches block device metadata

Interview-ready simplification:

**Page cache is for files, buffer cache is for disks.**

## 12. Clearing Cache (Knowledge, Not Recommendation)

Command:

```
sync; echo 3 > /proc/sys/vm/drop_caches
```

Interview caution:

**Clearing cache is rarely needed in production.**

## 13. Real-Life Production Scenarios

### Scenario 1: Server Is Slow but CPU Is Low

- Check memory usage
- Check swap usage
- Look for heavy swapping

### Scenario 2: Application Restart Fixes Issue

- Likely memory leak
- Short-term fix: restart
- Long-term fix: code change

### Scenario 3: Sudden Application Kill

- Check OOM logs
- Increase RAM or tune application

## Chapter 6: Interview Takeaways

After this chapter, you should confidently explain:

- Physical vs virtual memory
- How Linux uses RAM
- Buffers and cache
- Swap behavior
- OOM Killer logic
- Memory leaks and debugging
- Key memory monitoring commands