Linux Operating System (4331602) - Summer 2024 Solution

Milav Dabgar

June 10, 2024

Question 1(a) [3 marks]

Define Operating System and give its goal.

Solution

Operating System Definition: A program that acts as an interface between computer hardware and users, managing system resources and controlling program execution.
Goals of Operating System:

Table 1. OS Goals

Goal	Description
Resource Management	Efficiently allocate CPU, memory, I/O devices
User Convenience	Provide easy-to-use interface
System Protection	Secure system from unauthorized access

Mnemonic

"RUS: Resource management, User convenience, System protection"

Question 1(b) [4 marks]

Give name Components of Computer System & Explain need of Operating system.

Solution

Computer System Components:

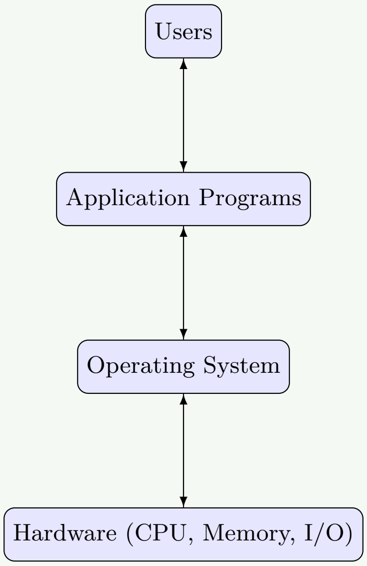

Figure 1. Computer System Hierarchy

Need of Operating System:

- **Resource Manager**: Controls hardware allocation
- **Interface Provider**: Easy communication between user and hardware
- **Security**: Protects system from threats
- **Error Handling**: Manages system errors efficiently

Mnemonic

"RISE: Resource management, Interface, Security, Error handling"

Question 1(c) [7 marks]

Explain below types of Operating system.

Solution

I. Batch Operating System

Table 2. Batch OS

Feature	**Description**
Processing	Jobs processed in batches without user interaction
Efficiency	High throughput, low user interaction
Example	IBM mainframes

II. Multiprogramming Operating System

Table 3. Multiprogramming OS

Feature	**Description**
Concept	Multiple programs in memory simultaneously
CPU Usage	Better CPU utilization
Advantage	Reduced idle time

III. Time Sharing Operating System

Table 4. Time Sharing OS

Feature	**Description**
Time Slices	CPU time divided among users
Response	Quick response time
Example	Unix, Linux

Mnemonic

"BMT: Batch, Multiprogramming, Time-sharing"

Question 1(c) OR [7 marks]

Explain Linux Architecture & characteristics with its components.

Solution

Linux Architecture:

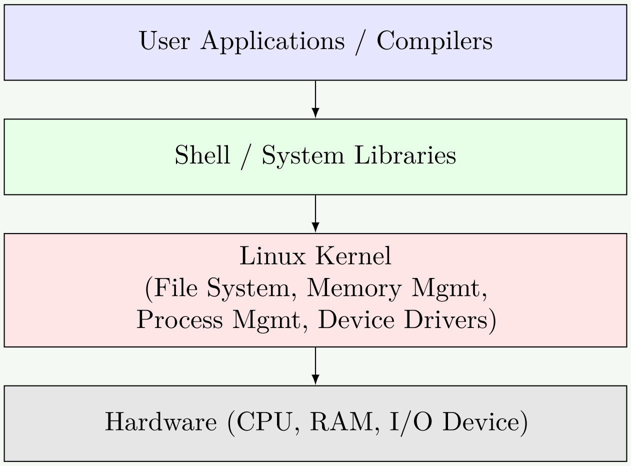
User Applications / Compilers
Shell / System Libraries
Linux Kernel
(File System, Memory Mgmt,
Process Mgmt, Device Drivers)
Hardware (CPU, RAM, I/O Device)

Figure 2. Linux Architecture

Linux Characteristics:

Table 5. Characteristics

Characteristic	**Description**
Open Source	Free and modifiable
Multiuser	Multiple users simultaneously
Multitasking	Multiple processes concurrently
Portable	Runs on various hardware

Components:

- **Kernel**: Core of operating system
- **Shell**: Command interpreter
- **File System**: Organizes data storage

Mnemonic

"COMP: Core, Open source, Multiuser, Portable"

Question 2(a) [3 marks]

Describe Process Control Block. And define (1) PID (2) stack pointer (3) program counter

Solution

Process Control Block (PCB): Data structure containing process information for OS management.
Definitions:

Table 6. PCB Definitions

Term	Definition
PID	Process Identifier - unique number for each process
Stack Pointer	Points to top of process stack
Program Counter	Contains address of next instruction

Mnemonic

"PSP: PID, Stack pointer, Program counter"

Question 2(b) [4 marks]

Describe the Process Model and Process states

Solution

Process Model: Conceptual representation of how processes are managed by OS.
Process States:

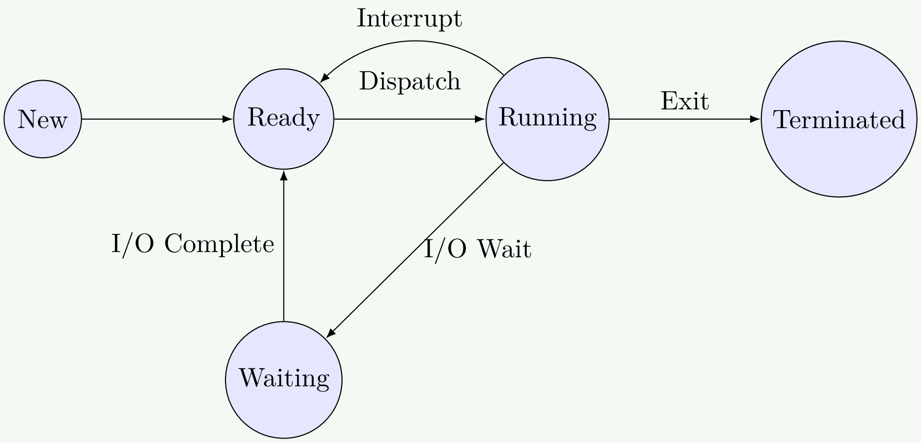

Figure 3. Process State Diagram

Table 7. Process States

State	Description
New	Process being created
Ready	Waiting for CPU
Running	Executing instructions
Waiting	Waiting for I/O
Terminated	Process finished

Mnemonic

"NRRWT: New, Ready, Running, Waiting, Terminated"

Question 2(c) [7 marks]

Demonstrate Scheduling Algorithm:(I) First Come First Serve, (II) Shortest Job First

Solution

I. First Come First Serve (FCFS)

Table 8. FCFS Scheduling

Process	Arrival	Burst	Completion	Turnaround
P1	0	4	4	4
P2	1	3	7	6
P3	2	2	9	7

Average Turnaround Time = (4+6+7)/3 = 5.67
II. Shortest Job First (SJF)

Table 9. SJF Scheduling

Process	Arrival	Burst	Completion	Turnaround
P3	2	2	4	2
P2	1	3	7	6
P1	0	4	11	11

Average Turnaround Time = (2+6+11)/3 = 6.33

Mnemonic

"FS: FCFS (First order), SJF (Shortest first)"

Question 2(a) OR [3 marks]

Define Race condition, Mutual Exclusion

Solution

Table 10. Race vs Mutual Exclusion

Term	**Definition**
Race Condition	Multiple processes access shared data simultaneously causing inconsistent results
Mutual Exclusion	Only one process can access critical section at a time

Example: Two processes updating same bank account balance.

Mnemonic

"RM: Race (simultaneous access), Mutual (one at a time)"

Question 2(b) OR [4 marks]

Define all Throughput, Turnaround Time, Waiting Time, Response Time

Solution

Table 11. Scheduling Metrics

Term	Definition
Throughput	Number of processes completed per unit time
Turnaround Time	Total time from submission to completion
Waiting Time	Time spent waiting in ready queue
Response Time	Time from submission to first response

Formulae:

- **Turnaround Time** = Completion Time - Arrival Time
- **Waiting Time** = Turnaround Time - Burst Time
- **Response Time** = First CPU Time - Arrival Time

Mnemonic

"TTWR: Throughput, Turnaround, Waiting, Response"

Question 2(c) OR [7 marks]

Explain Round Robin Algorithm with example.

Solution

Round Robin: Each process gets equal CPU time slice (quantum).
Example (Time Quantum = 2):

Table 12. RR Example Processes

Process	Burst Time
P1	5
P2	3
P3	4

Execution Timeline:

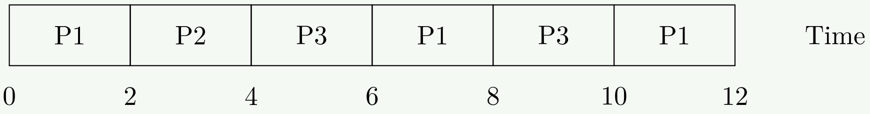

Figure 4. RR Execution Timeline

Table 13. RR Results

Process	Completion Time	Turnaround Time
P1	12	12
P2	6	6
P3	10	10

Average Turnaround Time = (12+6+10)/3 = 9.33
Advantages:

- **Fair**: Equal time to all processes
- **Responsive**: Good for interactive systems

Mnemonic

"RR-FE: Round Robin gives Fair and Equal time"

Question 3(a) [3 marks]

Give File Access Methods type

Solution

Table 14. File Access Methods

Method	Description
Sequential	Read/write in order from beginning
Direct	Access any record directly
Indexed	Use index to locate records

Mnemonic

"SDI: Sequential, Direct, Indexed"

Question 3(b) [4 marks]

Give Deadlock characteristics and Describe: Deadlock Prevention, Deadlock Avoidance

Solution

Deadlock Characteristics:

Table 15. Deadlock Conditions

Condition	Description
Mutual Exclusion	Resources cannot be shared
Hold and Wait	Process holds resource while waiting
No Preemption	Resources cannot be forcibly taken
Circular Wait	Circular chain of waiting processes

Deadlock Prevention: Remove any one of four conditions.
Deadlock Avoidance: Use algorithms like Banker's algorithm to avoid unsafe states.

Mnemonic

"MHNC: Mutual exclusion, Hold and wait, No preemption, Circular wait"

Question 3(c) [7 marks]

Explain the File Allocation Methods Contiguous, linked, indexed

Solution

File Allocation Methods:

Table 16. Allocation Methods Comparison

Method	Description	Advantage	Disadvantage
Contiguous	Sequential blocks	Fast access	External fragmentation
Linked	Scattered blocks with pointers	No fragmentation	Slow random access
Indexed	Index block contains addresses	Fast random access	Extra overhead

I. Contiguous Allocation:

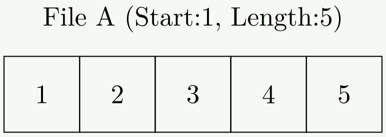

Figure 5. Contiguous Allocation

II. Linked Allocation:

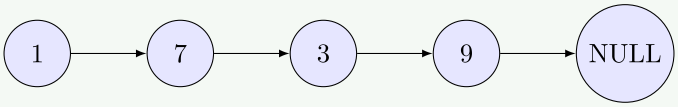

Figure 6. Linked Allocation

III. Indexed Allocation:

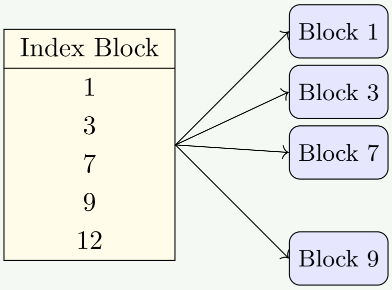

Figure 7. Indexed Allocation

Mnemonic

"CLI: Contiguous, Linked, Indexed"

Question 3(a) OR [3 marks]

Give knowledge Linux File System Structure

Solution

Linux File System Hierarchy:

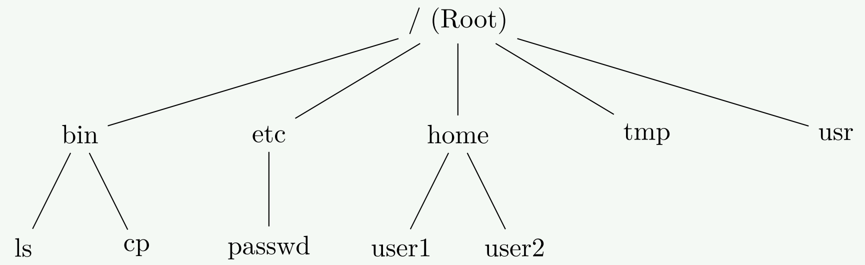

Figure 8. File System Tree

Table 17. Key Directories

Directory	**Purpose**
/bin	Essential system binaries
/etc	System configuration files
/home	User home directories

Mnemonic

"BEH: Bin, Etc, Home"

Question 3(b) OR [4 marks]

Explain Critical Section and Semaphore with example.

Solution

Critical Section: Code segment accessing shared resources.
Critical Section Structure:

- **Entry**: Request permission
- **Critical**: Access shared resource
- **Exit**: Release permission
- **Remainder**: Other code

Semaphore: Synchronization tool using counter variable.
Example:

Listing 1. Binary Semaphore

```
# Binary Semaphore Operations
wait(S):
  while S <= 0 do nothing
  S = S - 1

signal(S):
  S = S + 1
```

Mnemonic

"ECER: Entry, Critical, Exit, Remainder"

Question 3(c) OR [7 marks]

Define and explain Deadlock Avoidance, Deadlock Detection and Recovery

Solution

Deadlock Avoidance:

- Uses **Banker's Algorithm**
- Checks if resource allocation leads to safe state

Deadlock Detection:

- Periodically checks for deadlock using **Wait-for Graph**

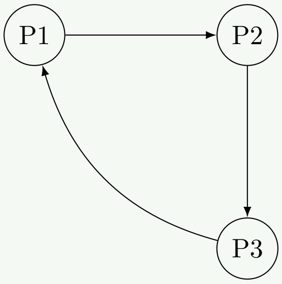

Figure 9. Wait-for Graph (Detection)

Deadlock Recovery Methods:

Table 18. Recovery Methods

Method	Description
Process Termination	Kill deadlocked processes
Resource Preemption	Take resources from processes
Rollback	Return to previous safe state

Mnemonic

"ADR: Avoidance, Detection, Recovery"

Question 4(a) [3 marks]

Why Need of file Protection explain?

Solution

Need for File Protection:

Table 19. Protection Needs

Reason	Description
Privacy	Protect personal data
Security	Prevent unauthorized access
Integrity	Maintain data consistency

Protection Mechanisms:

- Access Control Lists (ACL)
- File Permissions (Read, Write, Execute)
- User Authentication

Mnemonic

"PSI: Privacy, Security, Integrity"

Question 4(b) [4 marks]

Illustrate Program threats, System threats

Solution

Program Threats:

- **Virus**: Self-replicating malicious code
- **Worm**: Network-spreading malware
- **Trojan Horse**: Disguised malicious program

System Threats:

- **Denial of Service**: Overwhelm system resources
- **Port Scanning**: Find vulnerable services
- **Man-in-Middle**: Intercept communications

Mnemonic

"VWT-DPM: Virus, Worm, Trojan; DoS, Port scan, Man-in-middle"

Question 4(c) [7 marks]

Briefly detailing Operating System security policies and procedures

Solution

Security Policies:

Table 20. Security Policies

Policy	**Description**
Access Control	Who can access what resources
Authentication	Verify user identity
Authorization	Determine user permissions
Audit	Monitor and log activities

Security Procedures Flow:

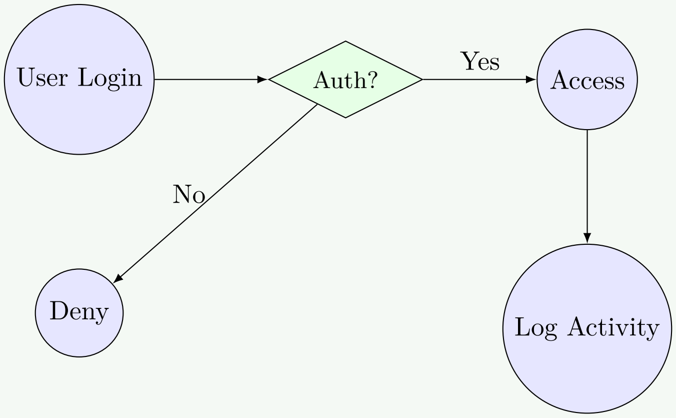

Figure 10. Security Flow

Implementation Steps:

1. User Registration and credential setup
2. Multi-factor Authentication
3. Role-based Access Control
4. Regular Security Audits

Mnemonic

"AAAA: Access control, Authentication, Authorization, Audit"

Question 4(a) OR [3 marks]

Give idea Authentication and Authorization.

Solution

Table 21. Auth vs Authz

Term	**Definition**	**Example**
Authentication	Verify user identity	Username/password
Authorization	Determine access rights	File permissions

Authentication Methods:

- Password-based
- Biometric
- Token-based

Mnemonic

"AA: Authentication (who), Authorization (what)"

Question 4(b) OR [4 marks]

Explain Operating System security policies and procedures

Solution

Security Policies Framework:

Table 22. Security Framework

Component	**Purpose**
User Management	Control user accounts
Data Protection	Secure sensitive information
Network Security	Protect communications
System Monitoring	Detect threats

Mnemonic

"UDNS: User, Data, Network, System"

Question 4(c) OR [7 marks]

Detailing the Security measures in Operating System

Solution

Comprehensive Security Measures:

- **Physical**: Server room access, bio locks
- **Network**: Firewalls, VPN
- **System**: Antivirus, patches
- **Application**: Secure coding
- **Data**: Encryption, backups

Access Control Matrix Example:

Table 23. Access Matrix

User	**File A**	**File B**
Admin	RWX	RWX
User1	RW-	R–
Guest	R–	—

Security Implementation Timeline:

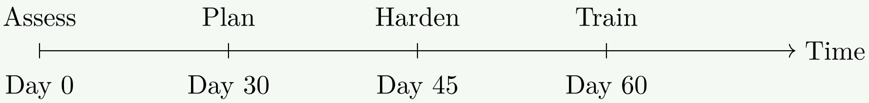

Figure 11. Implementation Timeline

Mnemonic

“PNSAD: Physical, Network, System, Application, Data”

Question 5(a) [3 marks]

Give five Basic commands: calendar, date

Solution

Basic Linux Commands:

Table 24. Basic Commands

Command	**Function**	**Example**
`cal`	Display calendar	`cal 2024`
`date`	Show current date/time	`date +%d/%m/%Y`
`who`	Show logged users	`who`
`pwd`	Print working directory	`pwd`
`clear`	Clear screen	`clear`

Mnemonic

“CDWPC: Cal, Date, Who, Pwd, Clear”

Question 5(b) [4 marks]

Explain Linux File and Directory Commands: ls, cat, mkdir, rmdir, pwd.

Solution

File and Directory Commands:

Table 25. File Commands

Command	Function	Example
ls	List directory contents	ls -la
cat	Display file content	cat file.txt
mkdir	Create directory	mkdir newdir
rmdir	Remove empty directory	rmdir olddir

Usage Examples:

Listing 2. File Commands

```
# List files with details
ls -l /home/user

# Create multiple directories
mkdir -p dir1/dir2/dir3

# Display file with line numbers
cat -n document.txt
```

Mnemonic

"LCMRP: List, Cat, Mkdir, Rmdir, Pwd"

Question 5(c) [7 marks]

Understand and apply control statements Write a shell script to perform given operations: Write a shell script to find maximum number among three numbers.

Solution

Listing 3. Maximum of 3 Numbers

```
#!/bin/bash
# Script to find maximum of three numbers

echo "Enter three numbers:"
read -p "First number: " num1
read -p "Second number: " num2
read -p "Third number: " num3

# Method 1: Using if-elif-else
if [ $num1 -ge $num2 ] && [ $num1 -ge $num3 ]; then
    max=$num1
elif [ $num2 -ge $num1 ] && [ $num2 -ge $num3 ]; then
    max=$num2
else
    max=$num3
fi

echo "Maximum number is: $max"
```

Comparison Operators:

- `-gt`: Greater than
- `-ge`: Greater than or equal to
- `-eq`: Equal to

Mnemonic

"IER: If (condition), Echo (output), Read (input)"

Question 5(a) OR [3 marks]

What is Linux Process commands: top, ps, kill

Solution

Linux Process Commands:

Table 26. Process Commands

Command	Function	Usage
`top`	Display running processes	`top`
`ps`	Show process status	`ps aux`
`kill`	Terminate process	`kill PID`

Details:

- `top`: Real-time CPU/Memory usage
- `ps -aux`: Full detailed process list
- `kill -9 PID`: Force kill a process

Mnemonic

"TPK: Top, Ps, Kill"

Question 5(b) OR [4 marks]

Linux File and Directory Commands: rm, mv,split,diff, grep

Solution

Advanced File Commands:

Table 27. Advanced Commands

Cmd	Function	Example
`rm`	Remove files	`rm -rf folder`
`mv`	Move/rename	`mv old new`
`split`	Split files	`split -l 50 f.txt`
`diff`	Compare files	`diff f1 f2`
`grep`	Search text	`grep "err" log`

Mnemonic

"RMSDG: Remove, Move, Split, Diff, Grep"

Question 5(c) OR [7 marks]

Write a shell script to read five numbers from user and find average of five numbers.

Solution

Listing 4. Average of 5 Numbers

```bash
#!/bin/bash
# Script to calculate average of five numbers

echo "=== Average Calculator ==="
sum=0

echo "Enter 5 numbers:"
for i in {1..5}; do
    read -p "Enter number $i: " num
    sum=$((sum + num))
done

# Calculate average
average=$((sum / 5))

echo "--------------------------------"
echo "Sum: $sum"
echo "Average: $average"
echo "--------------------------------"
```

Key Concepts:

- `$(())`: Arithmetic expansion
- `for`: Loop for iteration
- `read`: User input

Mnemonic

"RSAR: Read, Sum, Average, Result"