Linux Operating System (4331602) - Summer 2025 Solution

Milav Dabgar

May 13, 2025

Question 1(a) [3 marks]

Define Operating System and explain the need of OS.

Solution

Answer:

Operating System is a system software that acts as an intermediary between computer hardware and application software. It manages hardware resources and provides services to user programs.

Need of Operating System:

- **Resource Management**: Manages CPU, memory, storage, and I/O devices efficiently
- **User Interface**: Provides command-line and graphical interfaces for user interaction
- **Program Execution**: Loads and executes user programs safely

Mnemonic

"RUP - Resource, User, Program management"

Question 1(b) [4 marks]

Write a short note on Process Control Block (PCB).

Solution

Answer:

Process Control Block (PCB) is a data structure maintained by the operating system for each running process.

Table 1. PCB Components

PCB Component	Description
Process ID	Unique identifier for the process
Process State	Current state (ready, running, waiting)
Program Counter	Address of next instruction to execute
CPU Registers	Values of CPU registers when process is suspended
Memory Management	Base and limit registers, page tables
I/O Status	List of open files and I/O devices

Key Functions:

- **Process Identification**: Stores unique process ID and parent process ID
- **State Information**: Maintains current execution state and context
- **Resource Allocation**: Tracks allocated resources and memory usage

Mnemonic

"PIS - Process ID, Information, State tracking"

Question 1(c) [7 marks]

List different types of Operating systems. Explain the working of batch operating systems with a suitable example.

Solution

Answer:
Types of Operating Systems:

Table 2. Types of Operating Systems

Type	Description
Batch OS	Groups similar jobs and executes them together
Time-sharing OS	Multiple users share system simultaneously
Real-time OS	Provides guaranteed response time
Distributed OS	Manages multiple interconnected computers
Network OS	Provides network services and resource sharing
Mobile OS	Designed for mobile devices

Batch Operating System Working:

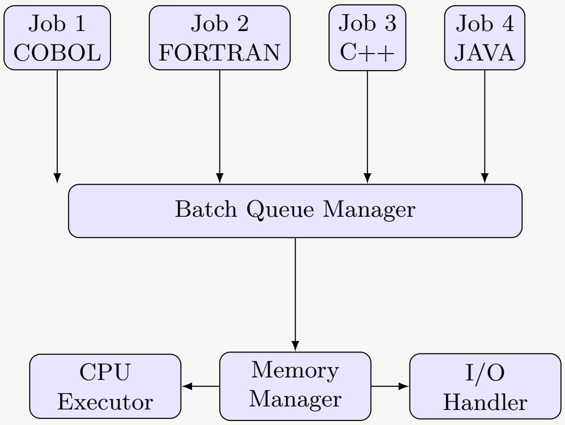


Figure 1. Batch Operating System

Example: Bank transaction processing where all day's transactions are collected and processed together at night for efficiency.

Key Features:

- **Job Grouping**: Similar jobs executed together for efficiency
- **No User Interaction**: Jobs run without user intervention once submitted
- **High Throughput**: Maximizes system utilization

Mnemonic

"JNH - Jobs grouped, No interaction, High throughput"

Question 1(c) OR [7 marks]

List different types of Operating systems. Explain the real time operating systems in detail.

Solution

Answer:

Types of Operating Systems: (Same table as above)

Real-Time Operating System (RTOS): Real-time OS provides guaranteed response within specified time constraints for critical applications.

Types of RTOS:

Table 3. Types of RTOS

Type	Deadline	Example
Hard Real-time	Must meet deadline	Air traffic control, pacemaker
Soft Real-time	Can tolerate some delay	Video streaming, online gaming
Firm Real-time	Occasional deadline miss acceptable	Live audio processing

Characteristics:

- **Deterministic**: Predictable response time for all operations
- **Priority-based Scheduling**: High-priority tasks get immediate attention
- **Minimal Interrupt Latency**: Fast context switching capabilities
- **Memory Management**: Real-time memory allocation without delays

Applications:

- Medical devices, automotive systems, industrial automation, aerospace control systems

Mnemonic

"DPMA - Deterministic, Priority-based, Minimal latency, Applications critical"

Question 2(a) [3 marks]

Differentiate between program and process.

Solution

Answer:

Table 4. Program vs Process

Aspect	Program	Process
Definition	Static code stored on disk	Program in execution
State	Passive entity	Active entity
Memory	No memory allocation	Allocated memory space
Lifetime	Permanent until deleted	Temporary during execution
Resources	No resource consumption	Consumes CPU, memory, I/O

Key Differences:

- **Static vs Dynamic**: Program is static file, process is dynamic execution
- **Resource Usage**: Process consumes system resources, program doesn't
- **Multiple Instances**: One program can create multiple processes

Mnemonic

“SDR - Static vs Dynamic, Resource usage, Multiple instances”

Question 2(b) [4 marks]

Explain the different states of a process with the help of a process state diagram.

Solution

Answer:

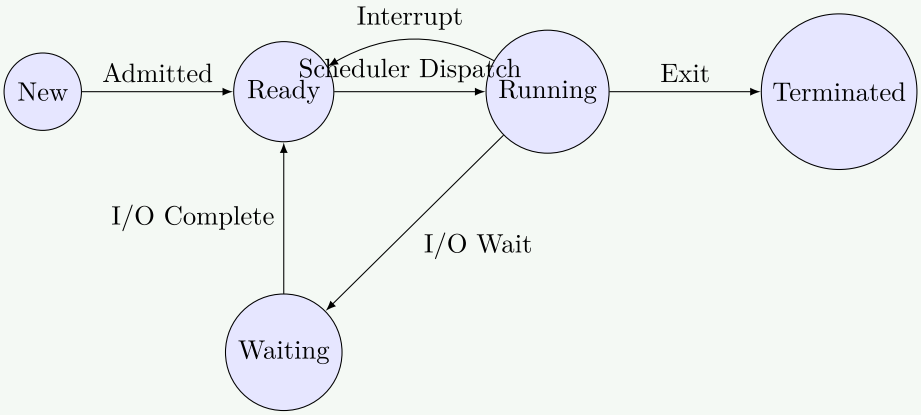


Figure 2. Process State Diagram

Process States:

Table 5. Process States

State	**Description**
New	Process being created
Ready	Waiting for CPU assignment
Running	Currently executing on CPU
Waiting	Blocked for I/O or event
Terminated	Process execution completed

State Transitions:

- **Ready to Running**: Process scheduler assigns CPU
- **Running to Ready**: Time slice expires or higher priority process arrives
- **Running to Waiting**: Process requests I/O operation
- **Waiting to Ready**: I/O operation completes

Mnemonic

“NRWRT - New, Ready, Waiting, Running, Terminated states”

Question 2(c) [7 marks]

Describe the Round Robin algorithm. Calculate the average waiting time & average turnaround time along with Gantt chart for the given data. Consider context switch = 01 ms and quantum time = 04 ms.

Solution

Answer:
Round Robin Algorithm: Round Robin is a preemptive scheduling algorithm where each process gets equal CPU time (quantum) in circular manner.
Given Data:

- Quantum Time = 4 ms
- Context Switch = 1 ms

Table 6. Process Data

Process	Arrival Time	Burst Time
P1	0	8
P2	3	3
P3	1	10
P4	4	5

Gantt Chart:

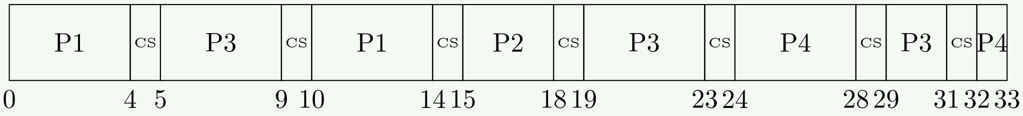


Figure 3. Gantt Chart (Round Robin)

Note: The exact timeline may vary slightly depending on interpretation of CS handling, assuming CS happens after process switch.
Calculations:

Table 7. Calculations

Process	Completion	Turnaround	Waiting
P1	14	14 - 0 = 14	14 - 8 = 6
P2	18	18 - 3 = 15	15 - 3 = 12
P3	31	31 - 1 = 30	30 - 10 = 20
P4	33	33 - 4 = 29	29 - 5 = 24

Average Waiting Time = $(6 + 12 + 20 + 24)/4 = 15.5$ ms
Average Turnaround Time = $(14 + 15 + 30 + 29)/4 = 22$ ms
Key Features:

- **Fair Scheduling**: Each process gets equal CPU time
- **Preemptive**: Running process is interrupted after quantum expires
- **Context Switching**: Overhead included in calculations

Mnemonic

"FPC - Fair, Preemptive, Context switching overhead"

Question 2(a) OR [3 marks]

Differentiate: CPU bound process v/s I/O bound process.

Solution

Answer:

Table 8. CPU Bound vs I/O Bound

Aspect	**CPU Bound Process**	**I/O Bound Process**
Primary Activity	Intensive calculations	Frequent I/O operations
CPU Usage	High CPU utilization	Low CPU utilization
Burst Time	Long CPU bursts	Short CPU bursts
Waiting Time	Less I/O waiting	More I/O waiting
Examples	Mathematical calculations	File operations, database queries

Key Differences:

- **Resource Consumption**: CPU-bound uses more processor, I/O-bound uses more input/output
- **Performance Impact**: CPU-bound affected by processor speed, I/O-bound by storage speed
- **Scheduling Priority**: Different algorithms favor each type differently

Mnemonic

"CIR - CPU intensive, I/O intensive, Resource usage differs"

Question 2(b) OR [4 marks]

What is a deadlock? Explain the necessary conditions for a deadlock to occur.

Solution

Answer:
Deadlock is a situation where two or more processes are permanently blocked, each waiting for resources held by others.
Necessary Conditions (Coffman Conditions):

Table 9. Coffman Conditions

Condition	**Description**
Mutual Exclusion	Resources cannot be shared simultaneously
Hold and Wait	Process holds resources while waiting for others
No Preemption	Resources cannot be forcibly taken from processes
Circular Wait	Circular chain of processes waiting for resources

Example Scenario:

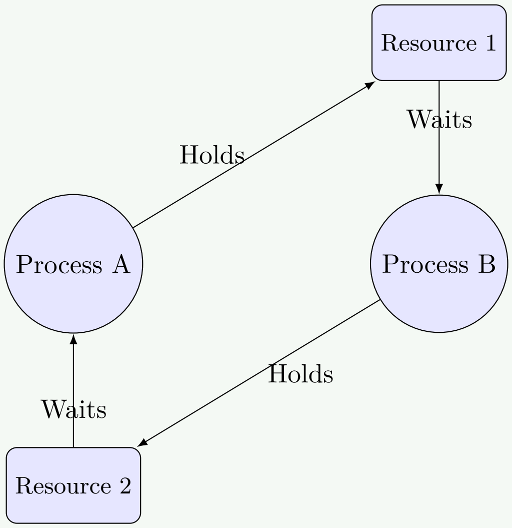


Figure 4. Deadlock Scenario

Mnemonic

"MHNC - Mutual exclusion, Hold-wait, No preemption, Circular wait"

Question 2(c) OR [7 marks]

Describe the FCFS algorithm. Calculate the average waiting time and average turn-around time along with Gantt chart for the given data.

Solution

Answer:
First Come First Serve (FCFS) Algorithm: FCFS is a non-preemptive scheduling algorithm where processes are executed in arrival order.
Given Data:

Table 10. Process Data

Process	Arrival Time	Burst Time
P1	0	7
P2	3	6
P3	5	9
P4	6	4

Gantt Chart:

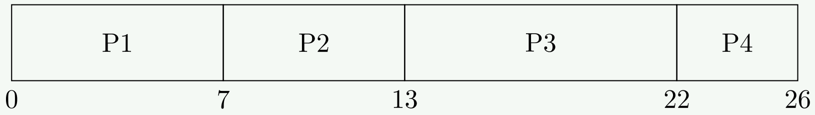


Figure 5. Gantt Chart (FCFS)

Calculations:

Table 11. FCFS Calculations

Process	Turnaround (CT-AT)	Waiting (TAT-BT)
P1	$7 - 0 = 7$	$7 - 7 = 0$
P2	$13 - 3 = 10$	$10 - 6 = 4$
P3	$22 - 5 = 17$	$17 - 9 = 8$
P4	$26 - 6 = 20$	$20 - 4 = 16$

Avg Waiting Time $= (0 + 4 + 8 + 16)/4 = 7$ ms
Avg Turnaround Time $= (7 + 10 + 17 + 20)/4 = 13.5$ ms
Characteristics:

- **Simple Implementation**: Easy to understand and implement
- **Non-preemptive**: Once started, process runs to completion
- **Convoy Effect**: Short processes wait for long processes

Mnemonic

"SNC - Simple, Non-preemptive, Convoy effect possible"

Question 3(a) [3 marks]

Explain single-level directory structure.

Solution

Answer:
Single-level directory structure is the simplest file organization where all files are stored in one directory.

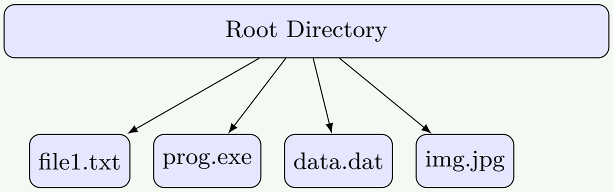


Figure 6. Single-level Directory

Characteristics:

- **Simple Structure**: All files in one location
- **Unique Names**: Each file must have unique name
- **No Organization**: No grouping or categorization possible

Limitations:

- Name collision when multiple users create files with same names
- Difficult to organize large number of files
- No privacy or access control between users

Mnemonic

"SUN - Simple, Unique names, No organization"

Question 3(b) [4 marks]

Explain the different file attributes.

Solution

Answer:
File attributes are metadata that provide information about files stored in the file system.

Table 12. File Attributes

Attribute	Description
Name	Human-readable file identifier
Type	File format (executable, text, image)
Size	Current file size in bytes
Location	Physical address on storage device
Protection	Access permissions (read, write, execute)
Time stamps	Creation, modification, access times
Owner	User who created the file

Common File Attributes:

- **Identifier**: Unique number for file system reference
- **Access Control**: User permissions and group access rights

Storage Location: File attributes are typically stored in directory entries or file allocation tables.

Mnemonic

"NTSLPTO - Name, Type, Size, Location, Protection, Time, Owner"

Question 3(c) [7 marks]

List the different file allocation methods and explain contiguous allocation with necessary diagram.

Solution

Answer:

File Allocation Methods:

- **Contiguous**: Files stored in consecutive blocks
- **Linked**: Files stored using linked list of blocks
- **Indexed**: Uses index block to point to data blocks

Contiguous Allocation: In contiguous allocation, each file occupies a set of contiguous blocks on the disk.

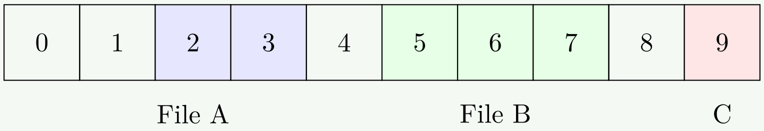


Table 13. Directory Table

Filename	Start	Length
File A	2	2
File B	5	3
File C	9	1

Advantages:

- **Fast Access**: Direct calculation of block addresses
- **Minimal Seek Time**: Consecutive blocks reduce head movement

Disadvantages:

- **External Fragmentation**: Unused spaces between files
- **File Size Limitation**: Difficult to extend files

Mnemonic

"FMS vs EFC - Fast access, Minimal seek, Simple vs External fragmentation, File size limits, Compaction needed"

Question 3(a) OR [3 marks]

Explain the different types of Linux file systems in brief.

Solution

Answer:

Table 14. Linux File Systems

File System	Description
ext2	Second extended filesystem, no journaling
ext3	Third extended filesystem with journaling
ext4	Fourth extended filesystem, improved performance
XFS	High-performance 64-bit journaling filesystem
Btrfs	B-tree filesystem with advanced features
ZFS	Copy-on-write filesystem with data integrity

Key Features:

- **Journaling**: ext3, ext4, XFS provide crash recovery
- **Performance**: ext4, XFS optimized for large files
- **Advanced Features**: Btrfs, ZFS offer snapshots and compression

Mnemonic

"EEXBZ - ext2/3/4, XFS, Btrfs, ZFS options"

Question 3(b) OR [4 marks]

Explain the different file operations.

Solution

Answer:

Table 15. File Operations

Operation	Description
Create	Make new file with specified name and attributes
Open	Prepare file for reading/writing operations
Read	Retrieve data from file at current position
Write	Store data to file at current position
Seek	Move file pointer to specific position
Close	Release file resources and update metadata
Delete	Remove file and deallocate storage space

File Operation Sequence:

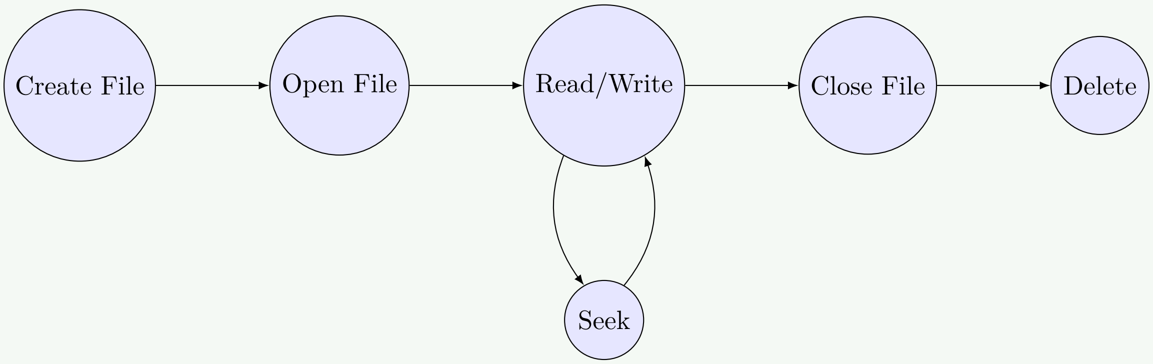


Figure 7. File Operation Sequence

Mnemonic

"CORWSCD - Create, Open, Read, Write, Seek, Close, Delete"

Question 3(c) OR [7 marks]

List the different file allocation methods and explain indexed allocation with necessary diagram.

Solution

Answer:
Indexed Allocation: In indexed allocation, each file has an index block containing pointers to data blocks.

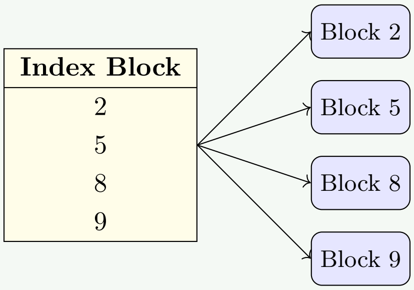


Figure 8. Indexed Allocation

Advantages:

- **No External Fragmentation**: Blocks can be anywhere on disk
- **Dynamic File Size**: Easy to extend files
- **Fast Random Access**: Direct access to any block

Disadvantages:

- **Index Block Overhead**: Extra space for pointers
- **Multiple Disk Access**: Two accesses needed (index + data)

Mnemonic

"NDF vs IMI - No fragmentation, Dynamic size, Fast access vs Index overhead, Multiple access"

Question 4(a) [3 marks]

Define System threats and explain its types.

Solution

Answer:
System Threats are malicious attempts to disrupt or damage computer system operations, steal information, or gain unauthorized access.

Table 16. System Threats

Threat Type	Description
Worms	Self-replicating programs that spread across networks
Viruses	Malicious code that attaches to other programs
Trojan Horses	Legitimate-looking programs with hidden malicious functions
Denial of Service	Attacks that overwhelm system resources
Port Scanning	Unauthorized probing of network services

Impact: System threats can lead to data loss, system downtime, privacy breaches, and financial damage.

Mnemonic

"WVTDP - Worms, Viruses, Trojans, DoS, Port scanning"

Question 4(b) [4 marks]

Differentiate: User Authentication v/s User Authorization.

Solution

Answer:

Table 17. Authentication vs Authorization

Aspect	User Authentication	User Authorization
Purpose	Verify user identity	Determine user permissions
When	Before system access	After authentication
Methods	Passwords, biometrics	Access control lists, roles
Question	”Who are you?”	”What can you do?”
Process	One-time at login	Continuous during session

Relationship: Authentication must occur before authorization.

Mnemonic

"WHO vs WHAT - Authentication asks WHO, Authorization determines WHAT"

Question 4(c) [7 marks]

Discuss various operating system security policies and procedures.

Solution

Answer:

Security Policies:

- **Access Control**: Defines who can access what resources
- **Password Policy**: Rules for password creation and management
- **Audit Policy**: Logging and monitoring of system activities
- **Update Policy**: Regular security patches and updates
- **Data Classification**: Categorizing data by sensitivity levels

System Monitoring Flow:

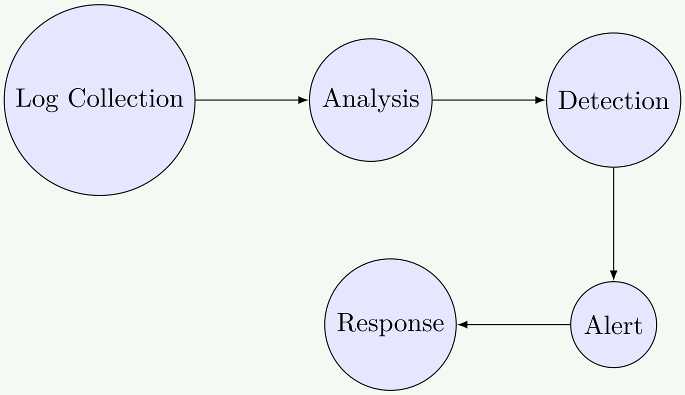


Figure 9. Security Monitoring

Security Procedures:

- **User Account Management**: Review accounts, revoke access
- **Incident Response**: Detection, containment, recovery
- **Backup and Recovery**: Regular backups, disaster planning

Mnemonic

"AAPUD + UMSIR - Policies + Procedures"

Question 4(a) OR [3 marks]

Define Program threats and explain its types.

Solution

Answer:
Program Threats are malicious software designed to disrupt, damage, or gain unauthorized access to computer programs and data.

Table 18. Program Threats

Threat Type	Description
Malware	Malicious software including viruses, worms
Spyware	Programs that secretly monitor user activities
Adware	Unwanted advertising software
Ransomware	Encrypts data and demands payment
Rootkits	Hide malicious activities from detection

Mnemonic

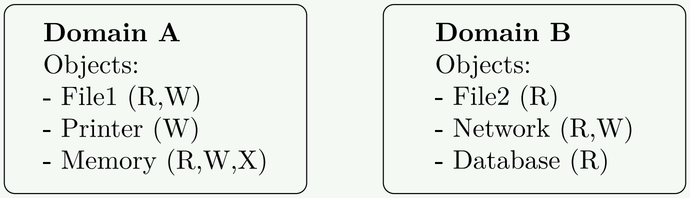
"MSARR - Malware, Spyware, Adware, Ransomware, Rootkits"

Question 4(b) OR [4 marks]

Explain the protection domain with a suitable example.

Solution

Answer:
Protection Domain is a set of objects and access rights that define what resources a process can access.
Domain Structure Example:



Domain A
Objects:
- File1 (R,W)
- Printer (W)
- Memory (R,W,X)

Domain B
Objects:
- File2 (R)
- Network (R,W)
- Database (R)

Figure 10. Protection Domains

Benefits:

- **Isolation**: Prevents unauthorized access
- **Flexibility**: Controlled resource sharing
- **Security**: Least privilege principle

Mnemonic

"OAS - Objects, Access rights, Subjects define domains"

Question 4(c) OR [7 marks]

Explain Access Control List in detail.

Solution

Answer:
Access Control List (ACL) specifies which users or processes are granted access to objects.
ACL Implementation:

File: /home/project/report.txt	
User	**Permissions**
alice	read, write
bob	read
admin	read, write, delete
group:dev	read, write

Table 19. ACL Example

Advantages:

- **Granular Control**: Fine-grained permission management
- **Audit Trail**: Clear record of who has access

Disadvantages:

- **Performance Overhead**: Must check ACL for each access
- **Complexity**: Difficult to manage for many users/objects

Mnemonic

"SOA + GDSC - Subject, Object, Access + Granular, Distributed, Centralized"

Question 5(a) [3 marks]

Explain the following commands: (i) man (ii) cd (iii) ls

Solution

Answer:

Table 20. Basic Commands

Command	**Purpose**	**Syntax**
man	Display manual pages	`man [cmd]`
cd	Change directory	`cd [dir]`
ls	List directory contents	`ls [opts] [dir]`

Details:

- **man**: Shows documentation (e.g., `man ls`)
- **cd**: Navigates filesystem (e.g., `cd /home`, `cd ..`)
- **ls**: Lists files (e.g., `ls -la` for hidden files)

Mnemonic

"MCD - Manual pages, Change directory, Directory listing"

Question 5(b) [4 marks]

Write a shell script to find maximum number among three numbers.

Solution

Listing 1. Maximum of 3 Numbers

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

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

if [ $num1 -gt $num2 ]; then
    if [ $num1 -gt $num3 ]; then
        max=$num1
    else
        max=$num3
    fi
else
    if [ $num2 -gt $num3 ]; then
        max=$num2
    else
        max=$num3
    fi
fi

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

Mnemonic

"ICD - Input, Compare, Display result"

Question 5(c) [7 marks]

Write a shell script to find the sum of all the individual digits in a given 5-digit number.

Solution

Listing 2. Sum of Digits

```
#!/bin/bash
# Script to find sum of digits in a 5-digit number

echo "Enter a 5-digit number:"
read number

# Validate input
if [ ${#number} -ne 5 ] || ! [[ $number =~ ^[0-9]+$ ]]; then
    echo "Error: Please enter exactly 5 digits"
    exit 1
fi

sum=0
temp=$number

# Extract and sum each digit
while [ $temp -gt 0 ]; do
    digit=$((temp % 10))    # Get last digit
    sum=$((sum + digit))    # Add to sum
    temp=$((temp / 10))     # Remove last digit
done

echo "Number: $number"
echo "Sum of digits: $sum"
```

Mnemonic

"VEDS - Validate, Extract, Display, Sum digits"

Question 5(a) OR [3 marks]

Explain the following commands: (i) date (ii) top (iii) cmp

Solution

Answer:

Table 21. More Commands

Cmd	Purpose	Example
date	Display/set system date	`date +%F`
top	Real-time process view	`top`
cmp	Byte-by-byte file compare	`cmp f1 f2`

Mnemonic

"DTC - Date/time, Task monitor, Compare files"

Question 5(b) OR [4 marks]

Explain the installation steps of Linux.

Solution

Answer:
Installation Process:

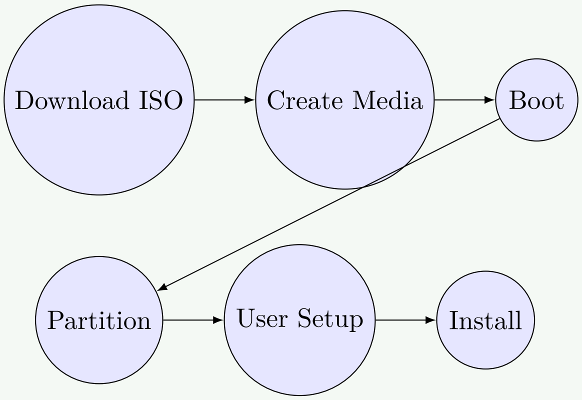


Figure 11. Installation Flow

Key Steps:

1. **Partitioning**: Configure disk space (root, home, swap)
2. **Configuration**: Set timezone, keyboard, user accounts
3. **Installation**: Copy system files and install bootloader

Mnemonic

"DCBCPUPI - Download, Create, Boot, Choose, Partition, User, Package, Install"

Question 5(c) OR [7 marks]

Write a shell script to find sum and average of N numbers.

Solution

Listing 3. Sum and Average of N Numbers

```bash
#!/bin/bash
# Script to find sum and average of N numbers

echo "How many numbers do you want to enter?"
read n

# Validate input
if ! [[ $n =~ ^[0-9]+$ ]] || [ $n -le 0 ]; then
    echo "Error: Please enter a positive integer"
    exit 1
fi
```

```bash

sum=0
echo "Enter $n numbers:"

# Read N numbers
for ((i=1; i<=n; i++)); do
    echo -n "Enter number $i: "
    read number
    # Simple accumulation
    sum=$(echo "$sum + $number" | bc -l)
done

# Calculate average
average=$(echo "scale=2; $sum / $n" | bc -l)

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

Mnemonic

"VLAD - Validate, Loop, Arithmetic, Display"