

# Linux Operating System (4331602) - Winter 2024 Solution

Milav Dabgar

December 05, 2024

## Question 1(a) [3 marks]

Explain Multiprogramming Operating System and give its advantages.

### Solution

Answer:

**Multiprogramming Operating System** allows multiple programs to reside in memory simultaneously and execute concurrently by sharing CPU time efficiently.

Table 1. Multiprogramming System Features

Feature	Description
Memory Management	Multiple programs loaded in memory
CPU Scheduling	CPU switches between programs
Resource Sharing	Efficient utilization of system resources

- **Increased CPU utilization:** CPU remains busy switching between programs
- **Better throughput:** More programs completed per unit time
- **Reduced response time:** Programs execute faster due to parallel processing

### Mnemonic

“MCP - Memory sharing, CPU utilization, Parallel execution”

## Question 1(b) [4 marks]

Explain Characteristics of Linux operating system.

### Solution

Answer:

Table 2. Linux Operating System Characteristics

Characteristic	Description
Open Source	Source code freely available and modifiable
Multi-user	Multiple users can access system simultaneously
Multi-tasking	Multiple processes run concurrently
Portable	Runs on various hardware platforms
Security	Strong permission system and access controls
Stability	Robust and reliable system performance

- **Case sensitive:** Distinguishes between uppercase and lowercase
- **Command line interface:** Powerful shell for system operations
- **File system hierarchy:** Organized directory structure starting from root (/)

### Mnemonic

“LAMPS - Linux is Accessible, Multi-user, Portable, Secure”

## Question 1(c) [7 marks]

Explain FCFS scheduling algorithm with its advantages and disadvantages. Calculate Average waiting time and average turnaround time for FCFS algorithm with gantt chart for following data.

### Solution

**Answer:**

**First Come First Serve (FCFS)** is a non-preemptive scheduling algorithm where processes are executed in order of their arrival.

**Table 3.** FCFS Algorithm Analysis

Aspect	Description
Policy	First arrived process gets CPU first
Type	Non-preemptive
Implementation	Simple queue (FIFO)

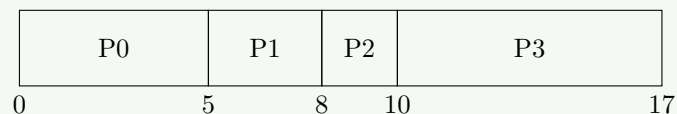
**Advantages:**

- **Simple implementation:** Easy to understand and code
- **Fair scheduling:** No starvation occurs

**Disadvantages:**

- **Convoy effect:** Short processes wait for long processes
- **Poor average waiting time:** Not optimal for system performance

**Gantt Chart Calculation:**



**Figure 1.** Gantt Chart (FCFS)

**Table 4.** Process Execution Analysis

Process	Arrival	Burst	Start	Finish	Waiting	Turnaround
P0	0	5	0	5	0	5
P1	3	3	5	8	2	5
P2	5	2	8	10	3	5
P3	6	7	10	17	4	11

Average Waiting Time =  $(0 + 2 + 3 + 4)/4 = 2.25$  ms

Average Turnaround Time =  $(5 + 5 + 5 + 11)/4 = 6.5$  ms

### Mnemonic

“FCFS-SiNo - First Come First Serve is Simple but Not optimal”

## Question 1(c) OR [7 marks]

Explain Round Robin algorithm with its advantages and disadvantages. Calculate Average waiting time and average turnaround time for Round Robin algorithm with gantt chart for following data. (Time Quantum = 2 ms)

### Solution

**Answer:**

**Round Robin** is a preemptive scheduling algorithm where each process gets equal CPU time slice (quantum).

**Table 5.** Round Robin Features

Feature	Description
<b>Time Quantum</b>	Fixed time slice for each process
<b>Preemption</b>	Process interrupted after quantum expires
<b>Queue Type</b>	Circular ready queue

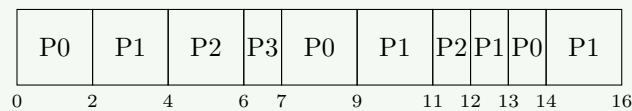
**Advantages:**

- **Fair allocation:** Each process gets equal CPU time
- **No starvation:** All processes eventually get CPU

**Disadvantages:**

- **Context switching overhead:** Frequent process switching
- **Performance depends on quantum:** Too small or large affects efficiency

**Gantt Chart (Quantum = 2ms):**



**Figure 2.** Gantt Chart (Round Robin)

**Table 6.** Round Robin Execution

Process	Arrival	Burst	Completion	Waiting	Turnaround
P0	0	4	14	10	14
P1	1	5	16	10	15
P2	2	3	12	7	10
P3	3	1	7	3	4

**Average Waiting Time** =  $(10 + 10 + 7 + 3)/4 = 7.5$  ms

**Average Turnaround Time** =  $(14 + 15 + 10 + 4)/4 = 10.75$  ms

### Mnemonic

“RR-TEQ - Round Robin uses Time Equal Quantum”

## Question 2(a) [3 marks]

Explain Real Time Operation System.

### Solution

**Answer:**

**Real Time Operating System (RTOS)** processes data and responds to events within strict time constraints.

Table 7. RTOS Types

Type	Response Time	Example
Hard Real-time	Guaranteed deadline	Missile guidance
Soft Real-time	Flexible deadline	Video streaming

- **Deterministic behavior:** Predictable response times
- **Priority-based scheduling:** Critical tasks get higher priority
- **Minimal latency:** Fast interrupt handling and context switching

## Mnemonic

“RTD - Real Time is Deterministic”

## Question 2(b) [4 marks]

Explain Process Life Cycle with diagram.

## Solution

**Answer:**

**Process Life Cycle** shows different states a process goes through during execution.

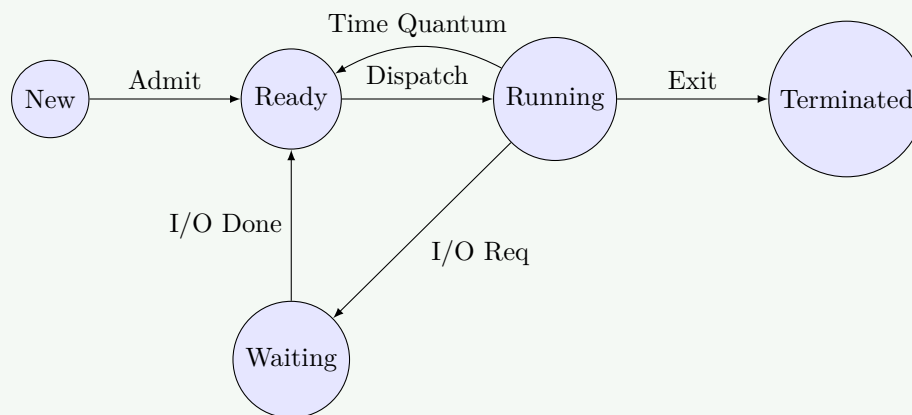


Figure 3. Process State Transition

Table 8. Process States

State	Description
New	Process being created
Ready	Waiting for CPU assignment
Running	Instructions being executed
Waiting	Waiting for I/O completion
Terminated	Process finished execution

## Mnemonic

“NRRWT - New Ready Running Waiting Terminated”

## Question 2(c) [7 marks]

Explain Various file and directory related commands in Linux.

### Solution

Answer:

Table 9. File Commands

Command	Function	Example
<b>ls</b>	List directory contents	<code>ls -la</code>
<b>cat</b>	Display file content	<code>cat file.txt</code>
<b>cp</b>	Copy files	<code>cp source dest</code>
<b>mv</b>	Move/rename files	<code>mv old new</code>
<b>rm</b>	Remove files	<code>rm file.txt</code>

Table 10. Directory Commands

Command	Function	Example
<b>mkdir</b>	Create directory	<code>mkdir mydir</code>
<b>rmdir</b>	Remove empty directory	<code>rmdir mydir</code>
<b>cd</b>	Change directory	<code>cd /home</code>
<b>pwd</b>	Print working directory	<code>pwd</code>

- **File permissions:** Use `chmod` to modify access rights
- **File ownership:** Use `chown` to change file owner
- **File information:** Use `stat` for detailed file information

### Mnemonic

“LCCMR-MRCP - List, Cat, Copy, Move, Remove for files; Make ...”

## Question 2(a) OR [3 marks]

Describe operating system services in detail.

### Solution

Answer:

Operating System Services provide interface between user applications and hardware resources.

Table 11. OS Services Categories

Category	Services
<b>User Interface</b>	GUI, Command Line, Batch
<b>Program Execution</b>	Loading, Running, Terminating
<b>I/O Operations</b>	File operations, Device communication
<b>File System</b>	Creation, Deletion, Manipulation
<b>Communication</b>	Process communication, Network
<b>Error Detection</b>	Hardware/Software error handling

- **Resource allocation:** CPU, memory, and device management
- **Accounting:** Track resource usage and performance

- **Protection and security:** Access control and authentication

### Mnemonic

“UPIFCE - User interface, Program execution, I/O, File system, Communication, Error detection”

## Question 2(b) OR [4 marks]

Explain Process Control Block.

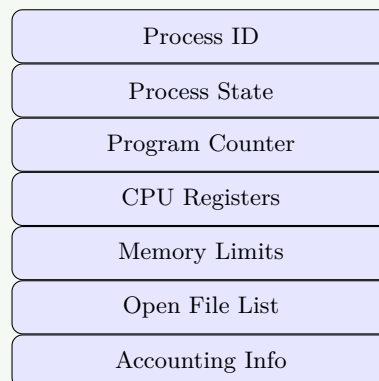
### Solution

**Answer:**

**Process Control Block (PCB)** is a data structure containing all information about a process.

**Table 12.** PCB Components

Component	Information Stored
<b>Process ID</b>	Unique process identifier
<b>Process State</b>	Current state (ready, running, waiting)
<b>CPU Registers</b>	Program counter, stack pointer, registers
<b>Memory Management</b>	Base/limit registers, page tables
<b>I/O Status</b>	Open files, allocated devices
<b>Accounting</b>	CPU usage, time limits



**Figure 4.** PCB Structure

### Mnemonic

“PPCMIA - Process ID, Process state, Program Counter, CPU registers, Memory, I/O, Accounting”

## Question 2(c) OR [7 marks]

Explain installation steps of Linux.

### Solution

**Answer:**

**Linux Installation** involves preparing system and installing operating system from bootable media.

**Table 13.** Installation Steps

Step	Description
1. Download ISO	Get Linux distribution image file
2. Create Bootable Media	Use USB/DVD to create installation media
3. Boot from Media	Change BIOS/UEFI boot order
4. Select Language	Choose installation language
5. Partition Disk	Create root, swap, home partitions
6. Configure Network	Set IP, DNS, hostname
7. Create User Account	Set username, password
8. Install Bootloader	Configure GRUB for booting
9. Complete Installation	Remove media and reboot

**Partitioning Scheme:**

- **Root (/):** 20GB minimum for system files
- **Swap:** 2x RAM size for virtual memory
- **Home (/home):** Remaining space for user data

**Post-installation:**

- **Update system:** `sudo apt update && sudo apt upgrade`
- **Install drivers:** Graphics, network, audio drivers
- **Configure security:** Firewall, user permissions

**Mnemonic**

“DCBSLNCIU - Download, Create media, Boot, Select language, Layout disk, Network, Create user...”

**Question 3(a) [3 marks]**

**Define: Process, Program, Swapping**

**Solution**

**Answer:**

**Table 14.** Basic Definitions

Term	Definition
<b>Process</b>	Program in execution with allocated resources
<b>Program</b>	Set of instructions stored on disk
<b>Swapping</b>	Moving processes between memory and disk

- **Process:** Active entity with process ID, memory space, and execution state
- **Program:** Passive entity, executable file stored in secondary storage
- **Swapping:** Memory management technique to handle more processes than physical memory

**Mnemonic**

“PAP-MDS - Process is Active Program; Program is instructions; Swapping is Memory-Disk transfer”

**Question 3(b) [4 marks]**

**List out various file operations and describe each of them.**

## Solution

Answer:

Table 15. File Operations

Operation	Description	System Call
Create	Make new file with specified name	creat()
Open	Prepare file for reading/writing	open()
Read	Retrieve data from file	read()
Write	Store data to file	write()
Close	Finish file access, release resources	close()
Delete	Remove file from file system	unlink()
Seek	Move file pointer to specific position	lseek()

- **File attributes:** Access permissions, timestamps, size information
- **File locking:** Prevent concurrent access conflicts
- **Buffer management:** Optimize I/O performance through caching

## Mnemonic

“CORWCDS - Create, Open, Read, Write, Close, Delete, Seek”

## Question 3(c) [7 marks]

Write a shell script to generate and print Fibonacci series.

## Solution

**Fibonacci Series** generates numbers where each number is sum of two preceding numbers.

Listing 1. Fibonacci Script

```

1  #!/bin/bash
2  # Fibonacci series generator
3
4  echo "Enter number of terms:"
5  read n
6
7  a=0
8  b=1
9
10 echo "Fibonacci Series:"
11 echo -n "$a $b "
12
13 for((i=2; i<n; i++))
14 do
15     c=$((a + b))
16     echo -n "$c "
17     a=$b
18     b=$c
19 done
20 echo

```

Table 16. Script Components



Component	Purpose
<b>#!/bin/bash</b>	Shebang line specifying interpreter
<b>read n</b>	Accept user input for number of terms
<b>for loop</b>	Iterate to generate sequence
<b>Arithmetic</b>	Calculate next number in series

**Mnemonic**

“FLAB - Fibonacci uses Loop with Addition of Both previous numbers”

**Question 3(a) OR [3 marks]**

List out types of scheduler and explain any one of them.

**Solution**

**Answer:**

**Table 17.** Types of Schedulers

Scheduler Type	Function
<b>Long-term</b>	Selects processes from job pool to ready queue
<b>Short-term</b>	Selects process from ready queue for CPU
<b>Medium-term</b>	Handles swapping between memory and disk

**Short-term Scheduler (CPU Scheduler):**

- **Frequency:** Executes very frequently (milliseconds)
- **Function:** Decides which process gets CPU next
- **Algorithms:** FCFS, SJF, Round Robin, Priority
- **Goal:** Maximize CPU utilization and throughput

**Mnemonic**

“LSM-JRC - Long-term (Job), Short-term (Ready), Medium-term (swap Control)”

**Question 3(b) OR [4 marks]**

List out various file attributes and describe each of them.

**Solution**

**Answer:**

**Table 18.** File Attributes

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

**Permission Structure:**

- **User (u):** Owner permissions
- **Group (g):** Group member permissions
- **Other (o):** All other users permissions

**Mnemonic**

“NTSLPTO - Name, Type, Size, Location, Protection, Time, Owner”

**Question 3(c) OR [7 marks]**

Write a shell script to sum of 1 to 10 using while loop.

**Solution**

**While Loop** continues execution as long as specified condition remains true.

**Listing 2.** Sum 1 to 10

```

1  #!/bin/bash
2  # Sum of numbers 1 to 10 using while loop
3
4  echo "Calculating sum of 1 to 10:"
5
6  i=1
7  sum=0
8
9  while [ $i -le 10 ]
10 do
11     sum=$((sum + i))
12     echo "Adding $i, current sum: $sum"
13     i=$((i + 1))
14 done
15
16 echo "Final sum of 1 to 10 is: $sum"
```

**Table 19.** Script Logic

Component	Purpose
<b>i=1</b>	Initialize counter variable
<b>sum=0</b>	Initialize accumulator
<b>while [ \$i -le 10 ]</b>	Continue while $i \leq 10$
<b>sum=\$((sum + i))</b>	Add current number to sum
<b>i=\$((i + 1))</b>	Increment counter

**Mnemonic**

“WICS - While loop needs Initialize, Condition, Sum calculation”

**Question 4(a) [3 marks]**

List out and explain condition for Deadlock to occur.

**Solution**

**Answer:**

**Deadlock** occurs when processes wait indefinitely for resources held by each other.

**Table 20.** Deadlock Conditions (Coffman Conditions)

Condition	Description
<b>Mutual Exclusion</b>	Only one process can use resource at a time
<b>Hold and Wait</b>	Process holds resources while waiting for others
<b>No Preemption</b>	Resources cannot be forcibly taken away
<b>Circular Wait</b>	Circular chain of processes waiting for resources

All four conditions must be true simultaneously for deadlock to occur.

**Mnemonic**

“MHNC - Mutual exclusion, Hold and wait, No preemption, Circular wait”

**Question 4(b) [4 marks]**

List out File access methods. Explain any one.

**Solution**

**Answer:**

**Table 21.** File Access Methods

Method	Description
<b>Sequential Access</b>	Read file from beginning to end
<b>Direct Access</b>	Jump to any record directly
<b>Index Sequential</b>	Combination of sequential and indexed access

**Sequential Access Method:**

- **Process:** Read records one by one in order
- **Advantages:** Simple implementation, efficient for batch processing
- **Disadvantages:** Slow for specific record access

**Mnemonic**

“SDI - Sequential (start to end), Direct (jump anywhere), Index (combined approach)”

## Question 4(c) [7 marks]

Describe Security measures in operating system.

### Solution

**Answer:**

**Operating System Security** protects system resources from unauthorized access and threats.

**Table 22.** Security Mechanisms

Mechanism	Description
<b>Authentication</b>	Verify user identity (passwords, biometrics)
<b>Authorization</b>	Control resource access permissions
<b>Access Control Lists</b>	Define who can access specific resources
<b>Encryption</b>	Protect data confidentiality
<b>Audit Logs</b>	Track system activities and access
<b>Firewalls</b>	Control network traffic

**Security Levels:**

- **Physical security:** Protect hardware and facilities
- **User authentication:** Login credentials and biometrics
- **File permissions:** Read, write, execute controls
- **Network security:** Secure communication protocols

### Mnemonic

“AAAEAF - Authentication, Authorization, Access control, Encryption, Audit, Firewall”

## Question 4(a) OR [3 marks]

List out ways to deal with deadlock. Explain deadlock detection and recovery.

### Solution

**Answer:**

**Table 23.** Deadlock Handling Methods

Method	Approach
<b>Prevention</b>	Ensure at least one Coffman condition cannot hold
<b>Avoidance</b>	Dynamically examine resource allocation state
<b>Detection &amp; Recovery</b>	Allow deadlock, then detect and recover
<b>Ignore</b>	Assume deadlock never occurs (Ostrich algorithm)

**Deadlock Detection:**

- **Wait-for graph:** Maintain graph of process dependencies
- **Detection algorithm:** Periodically check for cycles in graph

**Deadlock Recovery:**

- **Process termination:** Kill one or more deadlocked processes
- **Resource preemption:** Take resources from processes
- **Rollback:** Return processes to safe state using checkpoints

**Mnemonic**

“PADI - Prevention, Avoidance, Detection, Ignore”

**Question 4(b) OR [4 marks]**

List out File allocation methods. Explain any one.

**Solution**

**Answer:**

**Table 24.** File Allocation Methods

Method	Description
<b>Contiguous</b>	Allocate consecutive disk blocks
<b>Linked</b>	Use pointers to link scattered blocks
<b>Indexed</b>	Use index block to store block addresses

**Contiguous Allocation:**

- **Structure:** File occupies consecutive blocks on disk
- **Advantages:** Fast access, simple implementation, good for sequential access
- **Disadvantages:** External fragmentation, difficult to grow files
- **Directory entry:** Contains starting address and length

**Mnemonic**

“CLI - Contiguous (consecutive), Linked (pointers), Indexed (table)”

**Question 4(c) OR [7 marks]**

Describe program threats and system threats.

**Solution**

**Answer:**

**Program Threats** are malicious software that can harm system or data.

**Table 25.** Program Threats

Threat Type	Description
<b>Virus</b>	Self-replicating code that infects other programs
<b>Worm</b>	Standalone malware that spreads across networks
<b>Trojan Horse</b>	Malicious code disguised as legitimate software
<b>Logic Bomb</b>	Code that triggers malicious action on specific event

**System Threats** target operating system and system resources.

**Table 26.** System Threats

Threat Type	Description
Buffer Overflow	Overflow input buffers to execute malicious code
Denial of Service	Overwhelm system resources to make service unavailable
Privilege Escalation	Gain higher access privileges than authorized
Man-in-the-Middle	Intercept communication between two parties

**Mnemonic**

“VWTLB-BPDM - Virus, Worm, Trojan, Logic bomb... Buffer overflow...”

**Question 5(a) [3 marks]**

**Explain Inter Process Communication.**

**Solution**

**Answer:**

**Inter Process Communication (IPC)** enables processes to exchange data and synchronize activities.

**Table 27.** IPC Mechanisms

Mechanism	Description
Pipes	Unidirectional communication channel
Message Queues	Structured message passing
Shared Memory	Common memory area for multiple processes
Semaphores	Synchronization using counters
Signals	Software interrupts for notification

**Mnemonic**

“PMSSS - Pipes, Message queues, Shared memory, Semaphores, Signals”

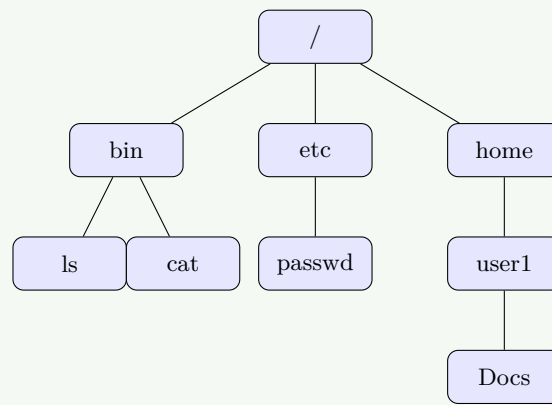
**Question 5(b) [4 marks]**

**Explain File structure used by Linux.**

**Solution**

**Answer:**

**Linux File System** follows hierarchical directory structure starting from root directory.



**Figure 5.** Linux File System Hierarchy

**Table 28.** Important Directories

Directory	Purpose
/	Root directory, top of hierarchy
/bin	Essential user commands
/etc	System configuration files
/home	User home directories
/var	Variable data (logs, mail)
/usr	User programs and utilities
/tmp	Temporary files

### Mnemonic

“BEHVUT - Bin, Etc, Home, Var, Usr, Tmp”

## Question 5(c) [7 marks]

Explain operating system security policies and procedures.

### Solution

**Answer:**

**Security Policies** define rules and guidelines for protecting system resources and data.

**Table 29.** Security Policy Components

Component	Description
Access Control Policy	Who can access what resources
Password Policy	Requirements for strong passwords
Audit Policy	What activities to monitor and log
Backup Policy	Data backup and recovery procedures
Incident Response	Steps to handle security breaches

**Security Procedures:**

- **Authentication:** Multi-factor authentication, Password complexity
- **Authorization:** Principle of least privilege, Role-based access
- **Monitoring:** Log analysis, Intrusion detection

**Mnemonic**

“APABI - Access control, Password, Audit, Backup, Incident response”

**Question 5(a) OR [3 marks]**

Explain Critical section.

**Solution**

**Answer:**

**Critical Section** is code segment where process accesses shared resources that must not be accessed concurrently.

**Table 30.** Critical Section Properties

Property	Description
<b>Mutual Exclusion</b>	Only one process in critical section at a time
<b>Progress</b>	Selection of next process cannot be postponed indefinitely
<b>Bounded Waiting</b>	Limit on number of times other processes enter critical section

**Structure:**

```

1 do {
2     entry_section();    // Request permission
3     critical_section(); // Access shared resource
4     exit_section();     // Release permission
5     remainder_section(); // Other work
6 } while(true);

```

**Mnemonic**

“MPB - Mutual exclusion, Progress, Bounded waiting”

**Question 5(b) OR [4 marks]**

Explain types of Linux file system.

**Solution**

**Answer:**

**Linux File Systems** organize and manage data storage on disk devices.

**Table 31.** Linux File System Types

File System	Description
<b>ext4</b>	Fourth extended file system, most common
<b>XFS</b>	High-performance journaling file system
<b>Btrfs</b>	B-tree file system with advanced features
<b>ZFS</b>	Zettabyte file system with built-in RAID

**ext4 Features:**

- **Journaling:** Faster recovery after system crash
- **Large file support:** Files up to 16TB
- **Backwards compatibility:** Can mount ext2/ext3 partitions



**Mnemonic**

“EXBZNF - Ext4, XFS, Btrfs, ZFS, NTFS, FAT32”

**Question 5(c) OR [7 marks]**

Explain need of protection mechanism and various protection domain.

**Solution**

**Answer:**

**Protection Mechanism** prevents processes from interfering with each other and system resources.

**Need for Protection:**

- **Resource sharing:** Multiple users/processes access same resources
- **Error containment:** Prevent bugs from affecting entire system
- **Security enforcement:** Implement access control policies

**Table 32.** Protection Domains

Domain Type	Description
User Domain	Limited access rights for user processes
Kernel Domain	Full access to system resources
System Domain	Intermediate privileges for system services

**Table 33.** Access Rights

Right	Description
Read	View content of resource
Write	Modify resource content
Execute	Run program or enter directory

**Mnemonic**

“RECES-UKS - Resource sharing, Error containment, Security; User domain, Kernel domain...”