

# Linux Operating System (4331602) - Winter 2024 Solution

Milav Dabgar

December 05, 2024

## પ્રશ્ન 1(a) [3 ગુણ]

મલ્ટિપ્રોગ્રામિંગ ઓપરેટિંગ સિસ્ટમ સમજાવો અને તેના ફાયદા જણાવો.

જવાબ

જવાબ:

Multiprogramming Operating System એકસાથે ઘણા પ્રોગ્રામને મેમરીમાં રાખીને CPU નો સમય અસરકારક રીતે વહેંચીને કામ કરે છે.

કોષ્ટક 1. Multiprogramming System Features

લક્ષણ	વર્ણન
Memory Management	મેમરીમાં અનેક પ્રોગ્રામ લોડ કરવા
CPU Scheduling	CPU પ્રોગ્રામ વચ્ચે બદલાય છે
Resource Sharing	સિસ્ટમ રિસોર્સનો કુશળ ઉપયોગ

- વધારો CPU ઉપયોગ: CPU પ્રોગ્રામ વચ્ચે બદલાતું રહે છે
- સારો throughput: એકમ સમયમાં વધુ પ્રોગ્રામ પૂર્ણ થાય છે
- ઓછો response time: પેરેલલ પ્રોસેસિંગથી પ્રોગ્રામ ઝડપથી ચાલે છે

મેમરી ટ્રીક

"MCP - Memory sharing, CPU utilization, Parallel execution"

## પ્રશ્ન 1(b) [4 ગુણ]

લિનક્સ ઓપરેટિંગ સિસ્ટમની લાક્ષણિકતાઓ સમજાવો.

જવાબ

જવાબ:

કોષ્ટક 2. Linux Operating System Characteristics

લાક્ષણિકતા	વર્ણન
Open Source	સોર્સ કોડ ફ્રીમાં ઉપલબ્ધ અને સુધારી શકાય
Multi-user	અનેક યુઝર એકસાથે સિસ્ટમ એક્સેસ કરી શકે
Multi-tasking	અનેક પ્રોસેસ એકસાથે ચાલી શકે
Portable	વિવિધ હાર્ડવેર પ્લેટફોર્મ પર ચાલે છે
Security	મજબૂત પરમિશન સિસ્ટમ અને એક્સેસ કંટ્રોલ
Stability	વિશ્વસનીય અને મજબૂત સિસ્ટમ પર્ફોર્મન્સ

- કેસ સેન્સિટિવ: અપરકેસ અને લોઅરકેસ વચ્ચે તફાવત કરે છે
- Command line interface: સિસ્ટમ ઓપરેશન માટે શક્તિશાળી શેલ
- File system hierarchy: રૂટ (/) થી શરૂ થતું વ્યવસ્થિત ડિરેક્ટરી સ્ટ્રક્ચર

### મેમરી ટ્રીક

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

## પ્રશ્ન 1(c) [7 ગુણ]

FCFS શેડ્યુલિંગ અલ્ગોરિધમ તેના ફાયદા અને ગેરફાયદા સાથે સમજાવો. નીચેના ડેટા માટે ગેન્ટ ચાર્ટ સાથે FCFS અલ્ગોરિધમ માટે સરેરાશ waiting time અને સરેરાશ turnaround time ની ગણતરી કરો.

### જવાબ

જવાબ:

First Come First Serve (FCFS) એ નોન-પ્રીએમ્પ્ટિવ શેડ્યુલિંગ અલ્ગોરિધમ છે જ્યાં પ્રોસેસ તેના આવવાના ક્રમમાં એક્ઝિક્યુટ થાય છે.

કોષ્ટક 3. FCFS Algorithm Analysis

પાસાં	વર્ણન
Policy	પહેલા આવેલ પ્રોસેસને પહેલા CPU મળે
Type	નોન-પ્રીએમ્પ્ટિવ
Implementation	સાદી ક્યુ (FIFO)

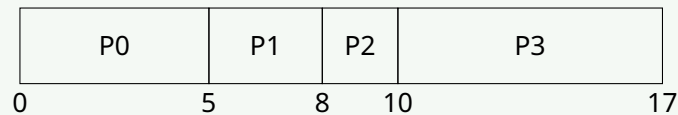
ફાયદા:

- સરળ અમલીકરણ: સમજવામાં અને કોડ કરવામાં સહેલું
- ન્યાયિક શેડ્યુલિંગ: કોઈ starvation થતું નથી

ગેરફાયદા:

- કોન્વીય ઇફેક્ટ: નાના પ્રોસેસ મોટા પ્રોસેસની રાહ જુએ છે
- ખરાબ સરેરાશ waiting time: સિસ્ટમ પર્ફોર્મન્સ માટે શ્રેષ્ઠ નથી

Gantt Chart Calculation:



આકૃતિ 1. Gantt Chart (FCFS)

કોષ્ટક 4. Process Execution Analysis

પ્રોસેસ	આવવાનો સમય	બર્સ્ટ	શરૂઆત	સમાપ્તિ	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

સરેરાશ Waiting Time =  $(0 + 2 + 3 + 4)/4 = 2.25 \text{ ms}$

સરેરાશ Turnaround Time =  $(5 + 5 + 5 + 11)/4 = 6.5 \text{ ms}$

### મેમરી ટ્રીક

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

## પ્રશ્ન 1(c) OR [7 ગુણ]

રાઉન્ડ રોબિન અલ્ગોરિધમ તેના ફાયદા અને ગેરફાયદા સાથે સમજાવો. નીચેના ડેટા માટે ગેન્ટ ચાર્ટ સાથે રાઉન્ડ રોબિન અલ્ગોરિધમ માટે સરેરાશ waiting time અને સરેરાશ turnaround time ની ગણતરી કરો. (ટાઇમ ક્વોન્ટમ = 2 ms)

**જવાબ**

**જવાબ:**

Round Robin એ પ્રીએમ્પ્ટિવ શેડ્યુલિંગ અલ્ગોરિધમ છે જ્યાં દરેક પ્રોસેસને સમાન CPU ટાઇમ સ્લાઇસ (ક્વોન્ટમ) મળે છે.

**કોષ્ટક 5. Round Robin Features**

લક્ષણ	વર્ણન
Time Quantum	દરેક પ્રોસેસ માટે નિશ્ચિત ટાઇમ સ્લાઇસ
Preemption	ક્વોન્ટમ પૂરું થયા પછી પ્રોસેસ અટકાવાય છે
Queue Type	વર્તુળાકાર રેડી ક્યુ

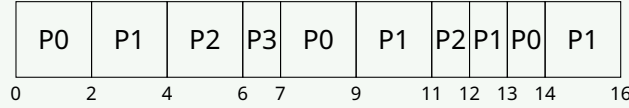
**ફાયદા:**

- ન્યાયિક વિતરણ: દરેક પ્રોસેસને સમાન CPU ટાઇમ મળે છે
- કોઈ starvation નથી: બધા પ્રોસેસને આખરે CPU મળે છે

**ગેરફાયદા:**

- Context switching overhead: વારંવાર પ્રોસેસ બદલાવાનું
- Performance depends on quantum: ખૂબ નાનું કે મોટું હોવાથી અસર થાય છે

Gantt Chart (Quantum = 2ms):



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

**કોષ્ટક 6. Round Robin Execution**

પ્રોસેસ	આવવાનો સમય	બર્સ્ટ	પૂર્ણતા	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

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

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

**મેમરી ટ્રીક**

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

## પ્રશ્ન 2(a) [3 ગુણ]

રિયલ ટાઇમ ઓપરેશન સિસ્ટમ સમજાવો.

**જવાબ**

**જવાબ:**

Real Time Operating System (RTOS) ડેટાને પ્રોસેસ કરે છે અને કડક સમય મર્યાદામાં ઇવેન્ટ્સની જવાબ આપે છે.

## કોષ્ટક 7. RTOS Types

પ્રકાર	રિસ્પોન્સ ટાઇમ	ઉદાહરણ
Hard Real-time	ગેરેન્ટીડ ડેડલાઇન	મિસાઇલ ગાઇડન્સ
Soft Real-time	લવચીક ડેડલાઇન	વિડિઓ સ્ટ્રીમિંગ

- **Deterministic behavior:** અનુમાનિત રિસ્પોન્સ ટાઇમ
- **Priority-based scheduling:** મહત્વપૂર્ણ ટાસ્ક્ને વધુ પ્રાયોરિટી
- **Minimal latency:** ઝડપી ઇન્ટરપ્ટ હેન્ડલિંગ અને કોન્ટેક્સ્ટ સ્વિચિંગ

## મેમરી ટ્રીક

“RTD - Real Time is Deterministic”

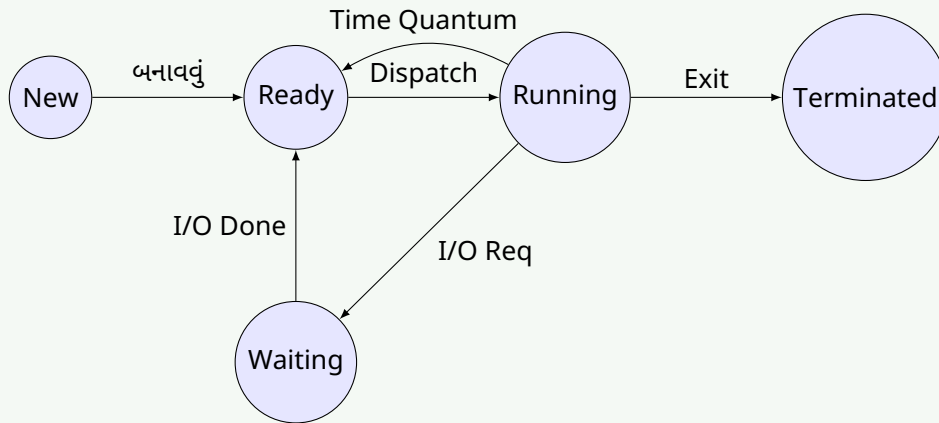
## પ્રશ્ન 2(b) [4 ગુણ]

ડાયાગ્રામ સાથે પ્રોસેસ લાઇફ સાઇકલ સમજાવો.

## જવાબ

જવાબ:

Process Life Cycle એક પ્રોસેસ એક્ઝિક્યુશન દરમિયાન પસાર થતા વિવિધ સ્ટેટ્સ દર્શાવે છે.



આકૃતિ 3. Process State Transition

## કોષ્ટક 8. Process States

સ્ટેટ	વર્ણન
New	પ્રોસેસ બનાવવામાં આવી રહ્યું છે
Ready	CPU એસાઇનમેન્ટ માટે રાહ જોઈ રહ્યું છે
Running	ઇન્સ્ટ્રક્શન્સ એક્ઝિક્યુટ થઈ રહ્યા છે
Waiting	I/O પૂર્ણતા માટે રાહ જોઈ રહ્યું છે
Terminated	પ્રોસેસે એક્ઝિક્યુશન પૂર્ણ કર્યું છે

## મેમરી ટ્રીક

“NRRWT - New Ready Running Waiting Terminated”

પ્રશ્ન 2(c) [7 ગુણ]

લિનક્સમાં વિવિધ ફાઇલ અને ડિરેક્ટરી સંબંધિત કમાન્ડ્સ સમજાવો.

જવાબ

જવાબ:

કોષ્ટક 9. File Commands

કમાન્ડ	કાર્ય	ઉદાહરણ
ls	ડિરેક્ટરી કન્ટેન્ટ્સ લિસ્ટ કરો	ls -la
cat	ફાઇલ કન્ટેન્ટ દર્શાવો	cat file.txt
cp	ફાઇલ કોપી કરો	cp source dest
mv	ફાઇલ મૂવ/રિનેમ કરો	mv old new
rm	ફાઇલ ડિલીટ કરો	rm file.txt

કોષ્ટક 10. Directory Commands

કમાન્ડ	કાર્ય	ઉદાહરણ
mkdir	ડિરેક્ટરી બનાવો	mkdir mydir
rmdir	ખાલી ડિરેક્ટરી ડિલીટ કરો	rmdir mydir
cd	ડિરેક્ટરી બદલો	cd /home
pwd	વર્કિંગ ડિરેક્ટરી પ્રિન્ટ કરો	pwd

- **File permissions:** એક્સેસ રાઇટ્સ સુધારવા માટે chmod વાપરો
- **File ownership:** ફાઇલ ઓનર બદલવા માટે chown વાપરો
- **File information:** વિગતવાર ફાઇલ ઇન્ફો માટે stat વાપરો

મેમરી ટ્રીક

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

પ્રશ્ન 2(a) OR [3 ગુણ]

ઓપરેટિંગ સિસ્ટમ સર્વિસિસનું વિગતવાર વર્ણન કરો.

જવાબ

જવાબ:

Operating System Services યુઝર એપ્લિકેશન્સ અને હાર્ડવેર રિસોર્સિસ વચ્ચે ઇન્ટરફેસ પ્રદાન કરે છે.

કોષ્ટક 11. OS Services Categories

કેટેગરી	સર્વિસિસ
User Interface	GUI, કમાન્ડ લાઇન, બેચ
Program Execution	લોડિંગ, રનિંગ, ટર્મિનેટિંગ
I/O Operations	ફાઇલ ઓપરેશન્સ, ડિવાઇસ કમ્યુનિકેશન
File System	ફાઇલ સિસ્ટમ, ડિલીશન, મેનિપ્યુલેશન
Communication	પ્રોસેસ કમ્યુનિકેશન, નેટવર્ક
Error Detection	હાર્ડવેર/સોફ્ટવેર એરર ડેટેક્ટિંગ

- **Resource allocation:** CPU, મેમરી અને ડિવાઇસ મેનેજમેન્ટ

- **Accounting:** રિસોર્સ ઉપયોગ અને પર્ફોર્મન્સ ટ્રેક કરવું
- **Protection and security:** એક્સેસ કંટ્રોલ અને ઓથેન્ટિકેશન

### મેમરી ટ્રીક

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

## પ્રશ્ન 2(b) OR [4 ગુણ]

પ્રોસેસ કંટ્રોલ બ્લોક સમજાવો.

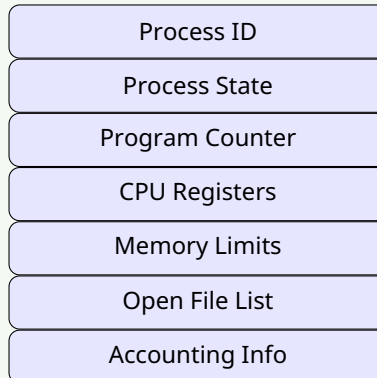
### જવાબ

જવાબ:

Process Control Block (PCB) એ ડેટા સ્ટ્રક્ચર છે જેમાં પ્રોસેસ વિશેની બધી માહિતી હોય છે.

કોષ્ટક 12. PCB Components

કમ્પોનન્ટ	સ્ટોર કરેલી માહિતી
Process ID	અનન્ય પ્રોસેસ આઈડેન્ટિફાયર
Process State	વર્તમાન સ્ટેટ (ready, running, waiting)
CPU Registers	પ્રોગ્રામ કાઉન્ટર, સ્ટેક પોઇન્ટર, રજિસ્ટર્સ
Memory Management	બેઝ/લિમિટ રજિસ્ટર્સ, પેજ ટેબલ્સ
I/O Status	ઓપન ફાઇલ્સ, એલોકેટેડ ડિવાઇસિસ
Accounting	CPU ઉપયોગ, ટાઇમ લિમિટ્સ



આકૃતિ 4. PCB Structure

### મેમરી ટ્રીક

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

## પ્રશ્ન 2(c) OR [7 ગુણ]

લિનક્સના ઇન્સ્ટોલેશન સ્ટેપ્સ સમજાવો.

જવાબ

જવાબ:  
Linux Installation સિસ્ટમ તૈયાર કરવા અને બૂટેબલ મીડિયાથી ઓપરેટિંગ સિસ્ટમ ઇન્સ્ટોલ કરવાનું છે.

કોષ્ટક 13. Installation Steps

સ્ટેપ	વર્ણન
1. Download ISO	લિનક્સ ડિસ્ટ્રિબ્યુશન ઇમેજ ફાઇલ લો
2. Create Bootable Media	ઇન્સ્ટોલેશન મીડિયા બનાવવા USB/DVD વાપરો
3. Boot from Media	BIOS/UEFI બૂટ ઓર્ડર બદલો
4. Select Language	ઇન્સ્ટોલેશન ભાષા પસંદ કરો
5. Partition Disk	રૂટ, સ્વેપ, હોમ પાર્ટિશન બનાવો
6. Configure Network	IP, DNS, હોસ્ટનેમ સેટ કરો
7. Create User Account	યુઝરનેમ, પાસવર્ડ સેટ કરો
8. Install Bootloader	બૂટિંગ માટે GRUB કોન્ફિગર કરો
9. Complete Installation	મીડિયા કાઢો અને રીબૂટ કરો

Partitioning Scheme:

- **Root (/):** સિસ્ટમ ફાઇલ્સ માટે ઓછામાં ઓછું 20GB
- **Swap:** વર્ચ્યુઅલ મેમરી માટે RAM નો 2x સાઇઝ
- **Home (/home):** યુઝર ડેટા માટે બાકીની જગ્યા

Post-installation:

- **Update system:** sudo apt update && sudo apt upgrade
- **Install drivers:** ગ્રાફિક્સ, નેટવર્ક, ઓડિયો ડ્રાઇવર્સ
- **Configure security:** ફાયરવોલ, યુઝર પરમિશન્સ

મેમરી ટ્રીક

``DCBSLNCIU - Download, Create media, Boot, Select language, Layout disk, Network, Create user..."

પ્રશ્ન 3(a) [3 ગુણ]

વ્યાખ્યાયિત કરો: પ્રક્રિયા, પ્રોગ્રામ, સ્વેપિંગ

જવાબ

જવાબ:

કોષ્ટક 14. Basic Definitions

શબ્દ	વ્યાખ્યા
Process	એલોકેટેડ રિસોર્સિસ સાથે એક્ઝિક્યુશનમાં રહેલ પ્રોગ્રામ
Program	ડિસ્ક પર સ્ટોર કરેલ ઇન્સ્ટ્રક્શન્સનો સેટ
Swapping	મેમરી અને ડિસ્ક વચ્ચે પ્રોસેસને મૂવ કરવું

- **Process:** પ્રોસેસ ID, મેમરી સ્પેસ અને એક્ઝિક્યુશન સ્ટેટ સાથેની સક્રિય એન્ટિટી
- **Program:** સેકન્ડરી સ્ટોરેજમાં સ્ટોર કરેલી નિષ્ક્રિય એન્ટિટી, એક્ઝિક્યુટેબલ ફાઇલ
- **Swapping:** ફિઝિકલ મેમરી કરતાં વધુ પ્રોસેસ હેન્ડલ કરવાની મેમરી મેનેજમેન્ટ ટેકનિક

મેમરી ટ્રીક

``PAP-MDS - Process is Active Program; Program is instructions; Swapping is Memory-Disk transfer"

### પ્રશ્ન 3(b) [4 ગુણ]

વિવિધ ફાઇલ ઓપરેશન્સની યાદી બનાવો અને તેમાંના દરેકનું વર્ણન કરો.

**જવાબ**

**જવાબ:**

**કોષ્ટક 15. File Operations**

ઓપરેશન	વર્ણન	સિસ્ટમ કોલ
<b>Create</b>	નિર્દિષ્ટ નામ સાથે નવી ફાઇલ બનાવો	creat()
<b>Open</b>	રીડિંગ/રાઇટિંગ માટે ફાઇલ તૈયાર કરો	open()
<b>Read</b>	ફાઇલમાંથી ડેટા મેળવો	read()
<b>Write</b>	ફાઇલમાં ડેટા સ્ટોર કરો	write()
<b>Close</b>	ફાઇલ એક્સેસ પૂર્ણ કરો, રિસોર્સિસ રીલીઝ કરો	close()
<b>Delete</b>	ફાઇલ સિસ્ટમમાંથી ફાઇલ કાઢો	unlink()
<b>Seek</b>	ફાઇલ પોઇન્ટરને સ્પેસિફિક પોઝિશન પર મૂવ કરો	lseek()

- **File attributes:** એક્સેસ પરમિશન્સ, ટાઇમસ્ટેમ્પ્સ, સાઇઝ ઇન્ફોર્મેશન
- **File locking:** કોન્કરન્ટ એક્સેસ કોન્ફ્લિક્ટ અટકાવવું
- **Buffer management:** કેશિંગ દ્વારા I/O પર્ફોર્મન્સ ઓપ્ટિમાઇઝ કરવું

**મેમરી ટ્રીક**

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

### પ્રશ્ન 3(c) [7 ગુણ]

ફિબોનાકી શ્રેણી બનાવવા અને પ્રિન્ટ કરવા માટે શેલ સ્ક્રિપ્ટ લખો.

**જવાબ**

**Fibonacci Series** એવી સંખ્યાઓ બનાવે છે જ્યાં દરેક સંખ્યા તેની પહેલાની બે સંખ્યાઓનો સરવાળો હોય છે.

**Listing 1. Fibonacci Script**

```

1  #!/bin/bash
2  # Fibonacci series generator
3
4  echo "કેટલા ટર્મ્સ દાખલ કરો:"
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

કોષ્ટક 16. Script Components

કમ્પોનન્ટ	હેતુ
#!/bin/bash	ઇન્ટરપ્રેટર સ્પેસિફાઇ કરતી શેલ્લો લાઇન
read n	ટર્મિનલ સંખ્યા માટે યુઝર ઇનપુટ સ્વીકારો
for loop	સિક્વન્સ જનરેટ કરવા માટે પુનરાવર્તન કરો
Arithmetic	શ્રેણીમાં આગળની સંખ્યા ગણો

મેમરી ટ્રીક

``FLAB - Fibonacci uses Loop with Addition of Both previous numbers"

## પ્રશ્ન 3(a) OR [3 ગુણ]

શેડ્યુલરના પ્રકારોની યાદી બનાવો અને તેમાંથી કોઈપણ એક સમજાવો.

જવાબ

જવાબ:

કોષ્ટક 17. Types of Schedulers

શેડ્યુલર પ્રકાર	કાર્ય
Long-term	જોબ પૂલમાંથી રેડી ક્યુમાં પ્રોસેસ પસંદ કરે છે
Short-term	રેડી ક્યુમાંથી CPU માટે પ્રોસેસ પસંદ કરે છે
Medium-term	મેમરી અને ડિસ્ક વચ્ચે સ્વેપિંગ હેન્ડલ કરે છે

Short-term Scheduler (CPU Scheduler):

- **Frequency:** ખૂબ જ વારંવાર એક્ઝિક્યુટ થાય છે (મિલિસેકન્ડ્સ)
- **Function:** નક્કી કરે છે કે આગળ કયો પ્રોસેસ CPU મેળવશે
- **Algorithms:** FCFS, SJF, રાઉન્ડ રોબિન, પ્રાયોરિટી
- **Goal:** CPU ઉપયોગ અને throughput મેક્સિમાઇઝ કરવું

મેમરી ટ્રીક

``LSM-JRC - Long-term (Job), Short-term (Ready), Medium-term (swap Control)"

## પ્રશ્ન 3(b) OR [4 ગુણ]

વિવિધ ફાઇલ એટ્રિબ્યુટ્સની યાદી બનાવો અને તેમાંથી દરેકનું વર્ણન કરો.

જવાબ

જવાબ:

કોષ્ટક 18. File Attributes

એટ્રિબ્યુટ	વર્ણન
Name	ફાઇલનું માનવ-વાંચી શકાય તેવું આઇડેન્ટિફાયર
Type	ફાઇલ ફોર્મેટ (ટેક્સ્ટ, બાઇનરી, એક્ઝિક્યુટેબલ)
Size	બાઇટ્સમાં વર્તમાન ફાઇલ સાઇઝ
Location	સ્ટોરેજ ડિવાઇસ પર ફિઝિકલ એડ્રેસ
Protection	એક્સેસ પરમિશન્સ (રીડ, રાઇટ, એક્ઝિક્યુટ)
Time stamps	ક્રિએશન, મોડિફિકેશન, એક્સેસ ટાઇમ્સ
Owner	ફાઇલ બનાવનાર યુઝર

#### Permission Structure:

- **User (u):** ઓનર પરમિશન્સ
- **Group (g):** ગ્રુપ મેમ્બર પરમિશન્સ
- **Other (o):** બાકીના બધા યુઝર્સની પરમિશન્સ

#### મેમરી ટ્રીક

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

## પ્રશ્ન 3(c) OR [7 ગુણ]

વ્હાઇલ લૂપનો ઉપયોગ કરીને 1 થી 10 ના સરવાળા માટે શેલ સ્ક્રિપ્ટ લખો.

#### જવાબ

**While Loop** નિર્દિષ્ટ કંડિશન સાચી રહે ત્યાં સુધી એક્ઝિક્યુશન ચાલુ રાખે છે.

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"
```

કોષ્ટક 19. Script Logic

કમ્પોનન્ટ	હેતુ
i=1	કાઉન્ટર વેરિએબલ ઇનિશિયલાઇઝ કરો
sum=0	એક્ક્યુમ્યુલેટર ઇનિશિયલાઇઝ કરો
while [ \$i -le 10 ]	$i \leq 10$ સુધી ચાલુ રાખો
sum=\$((sum + i))	વર્તમાન સંખ્યા સરવાળામાં ઉમેરો
i=\$((i + 1))	કાઉન્ટર વધારો

## મેમરી ટ્રીક

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

## પ્રશ્ન 4(a) [3 ગુણ]

ડેડલોક થવાની કંડિશનની યાદી બનાવો અને સમજાવો.

## જવાબ

જવાબ:

**Deadlock** ત્યારે થાય છે જ્યારે પ્રોસેસિસ એકબીજા પાસે રહેલા રિસોર્સિસ માટે અનિશ્ચિત સમય સુધી રાહ જુએ છે.

## કોષ્ટક 20. Deadlock Conditions (Coffman Conditions)

કંડિશન	વર્ણન
Mutual Exclusion	એક સમયે માત્ર એક પ્રોસેસ રિસોર્સ વાપરી શકે
Hold and Wait	પ્રોસેસ રિસોર્સ રાખીને બીજાની રાહ જુએ છે
No Preemption	રિસોર્સિસ બળજબરીથી છીનવી શકાતા નથી
Circular Wait	રિસોર્સિસ માટે રાહ જોતા પ્રોસેસિસની સર્ક્યુલર ચેઇન

ડેડલોક માટે ચારેય કંડિશન એકસાથે સાચી હોવી જરૂરી છે.

## મેમરી ટ્રીક

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

## પ્રશ્ન 4(b) [4 ગુણ]

ફાઇલ એક્સેસ મેથડ્સની ટ્રીચિ બનાવો. કોઈપણ એક સમજાવો.

## જવાબ

જવાબ:

## કોષ્ટક 21. File Access Methods

મેથડ	વર્ણન
Sequential Access	શરૂઆતથી અંત સુધી ફાઇલ વાંચો
Direct Access	કોઈપણ રેકૉર્ડ પર સીધું જમ્પ કરો
Index Sequential	સિક્વેન્શિયલ અને ઇન્ડેક્સ એક્સેસનું કોમ્બિનેશન

## Sequential Access Method:

- **Process:** રેકૉર્ડ્સને ક્રમમાં એક પછી એક વાંચો
- **Advantages:** સરળ અમલીકરણ, બેય પ્રોસેસિંગ માટે કુશળ
- **Disadvantages:** સ્પેસિફિક રેકૉર્ડ એક્સેસ માટે ધીમું

## મેમરી ટ્રીક

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

પ્રશ્ન 4(c) [7 ગુણ]

ઓપરેટિંગ સિસ્ટમમાં સુરક્ષા પગલાંનું વર્ણન કરો.

જવાબ

જવાબ:

Operating System Security અનધિકૃત એક્સેસ અને ખતરાઓથી સિસ્ટમ રિસોર્સિસને સુરક્ષિત રાખે છે.

કોષ્ટક 22. Security Mechanisms

મેકેનિઝમ	વર્ણન
Authentication	યુઝર આઇડેન્ટિટી વેરિફાઇ કરવું (પાસવર્ડ્સ, બાયોમેટ્રિક્સ)
Authorization	રિસોર્સ એક્સેસ પરમિશન્સ કંટ્રોલ કરવું
Access Control Lists	કોણ કયા રિસોર્સિસ એક્સેસ કરી શકે તે ડિક્રાઇન કરવું
Encryption	ડેટા ગુપ્તતા સુરક્ષિત રાખવી
Audit Logs	સિસ્ટમ પ્રવૃત્તિઓ અને એક્સેસ ટ્રેક કરવી
Firewalls	નેટવર્ક ટ્રાફિક કંટ્રોલ કરવું

Security Levels:

- Physical security: હાર્ડવેર અને સુવિધાઓને સુરક્ષિત રાખવી
- User authentication: લોગિન ક્રેડેન્શિયલ્સ અને બાયોમેટ્રિક્સ
- File permissions: રીડ, રાઇટ, એક્ઝિક્યુટ કંટ્રોલ્સ
- Network security: સિક્યોર કમ્યુનિકેશન પ્રોટોકોલ્સ

મેમરી ટ્રીક

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

પ્રશ્ન 4(a) OR [3 ગુણ]

ડેડલોકનો સામનો કરવાની રીતોની યાદી બનાવો. ડેડલોક ડિટેક્શન અને રિકવરી સમજાવો.

જવાબ

જવાબ:

કોષ્ટક 23. Deadlock Handling Methods

મેથડ	અભિગમ
Prevention	ઓછામાં ઓછી એક કોફ્મેન કંડિશન રોકવી
Avoidance	રિસોર્સ એલોકેશન સ્ટેટને ડાયનેમિકલી તપાસવું
Detection & Recovery	ડેડલોકને મંજૂરી આપો, પછી ડિટેક્ટ કરો અને રિકવર કરો
Ignore	ડેડલોક ક્યારેય નથી થતું તેવું માનવું (ઓસ્ટ્રિય અલ્ગોરિધમ)

Deadlock Detection:

- Wait-for graph: પ્રોસેસ ડિપેન્ડન્સીઝનો ગ્રાફ મેઇન્ટેઇન કરવો
- Detection algorithm: ગ્રાફમાં સાયકલ્સ માટે નિયમિત ચેક કરવું

Deadlock Recovery:

- Process termination: એક કે વધુ ડેડલોકડ પ્રોસેસિસને કિલ કરવા
- Resource preemption: પ્રોસેસિસ પાસેથી રિસોર્સિસ લેવા
- Rollback: ચેકપોઇન્ટ્સ વાપરીને પ્રોસેસિસને સેફ સ્ટેટમાં પાછા લાવવા

## મેમરી ટ્રીક

“PADI - Prevention, Avoidance, Detection, Ignore”

## પ્રશ્ન 4(b) OR [4 ગુણ]

ફાઇલ એલોકેશન મેથડ્સની યાદી બનાવો. કોઈપણ એક સમજાવો.

## જવાબ

જવાબ:

કોષ્ટક 24. File Allocation Methods

મેથડ	વર્ણન
Contiguous	સતત ડિસ્ક બ્લોકસ એલોકેટ કરવા
Linked	છૂટાછવાયા બ્લોકસને લિંક કરવા માટે પોઇન્ટર્સ વાપરવા
Indexed	બ્લોક એન્ડ્રેસિસ સ્ટોર કરવા માટે ઇન્ડેક્સ બ્લોક વાપરવો

## Contiguous Allocation:

- **Structure:** ફાઇલ ડિસ્ક પર સતત બ્લોકસ કબજે કરે છે
- **Advantages:** ઝડપી એક્સેસ, સરળ અમલીકરણ, સિક્વન્શિયલ એક્સેસ માટે સારું
- **Disadvantages:** એક્સટર્નલ ફ્રેગમેન્ટેશન, ફાઇલ વધારવી મુશ્કેલ
- **Directory entry:** શરૂઆતનું એન્ડ્રેસ અને લેન્થ સમાવે છે

## મેમરી ટ્રીક

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

## પ્રશ્ન 4(c) OR [7 ગુણ]

પ્રોગ્રામ થ્રેટ્સ અને સિસ્ટમ થ્રેટ્સ સમજાવો.

## જવાબ

જવાબ:

**Program Threats** એવા મેલિશિયસ સોફ્ટવેર છે જે સિસ્ટમ કે ડેટાને નુકસાન પહોંચાડી શકે છે.

કોષ્ટક 25. Program Threats

થ્રેટ પ્રકાર	વર્ણન
Virus	અન્ય પ્રોગ્રામ્સને ચેપ લગાડતો સ્વ-પ્રતિકૃતિ કરતો કોડ
Worm	નેટવર્ક પર ફેલાતો સ્ટેન્ડઅલોન મેલવેર
Trojan Horse	કાયદેસર સોફ્ટવેરના વેશમાં છુપાયેલો મેલિશિયસ કોડ
Logic Bomb	સ્પેસિફિક ઇવેન્ટ પર મેલિશિયસ એક્શન ટ્રિગર કરતો કોડ

**System Threats** ઓપરેટિંગ સિસ્ટમ અને સિસ્ટમ રિસોર્સિસને ટાર્ગેટ કરે છે.

કોષ્ટક 26. System Threats

ગ્રેટ પ્રકાર	વર્ણન
Buffer Overflow	મેલિશિયસ કોડ એક્ઝિક્યુટ કરવા ઇનપુટ બફર્સ ઓવરફ્લો કરવા
Denial of Service	સર્વિસ અનઉપલબ્ધ બનાવવા સિસ્ટમ રિસોર્સિસને ઓવરલોડ કરવા
Privilege Escalation	અધિકૃત કરતાં વધુ એક્સેસ પ્રિવિલેજ મેળવવા
Man-in-the-Middle	બે પક્ષો વચ્ચેની કમ્યુનિકેશન ઇન્ટરસેપ્ટ કરવી

### મેમરી ટ્રીક

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

## પ્રશ્ન 5(a) [3 ગુણ]

ઇન્ટર પ્રોસેસ કમ્યુનિકેશન સમજાવો.

### જવાબ

જવાબ:

Inter Process Communication (IPC) પ્રોસેસિસને ડેટા એક્સચેન્જ કરવા અને પ્રવૃત્તિઓ સિંક્રોનાઇઝ કરવા સક્ષમ બનાવે છે.

કોષ્ટક 27. IPC Mechanisms

મેકેનિઝમ	વર્ણન
Pipes	એકદિશીય કમ્યુનિકેશન ચેનલ
Message Queues	સ્ટ્રક્ચર્ડ મેસેજ પાસિંગ
Shared Memory	મલ્ટિપલ પ્રોસેસિસ માટે કોમન મેમરી એરિયા
Semaphores	કાઉન્ટર્સ વાપરીને સિંક્રોનાઇઝેશન
Signals	નોટિફિકેશન માટે સોફ્ટવેર ઇન્ટરપ્ટ્સ

### મેમરી ટ્રીક

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

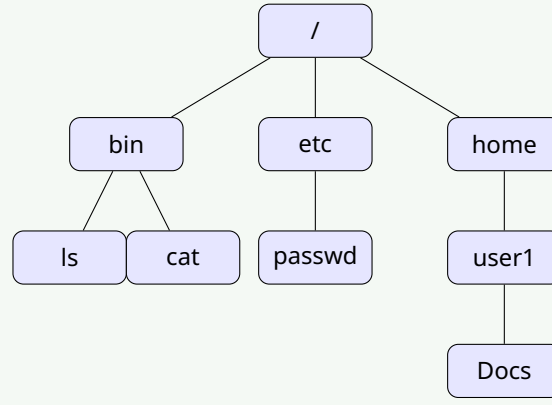
## પ્રશ્ન 5(b) [4 ગુણ]

લિનક્સ દ્વારા વપરાતું ફાઇલ સ્ટ્રક્ચર સમજાવો.

### જવાબ

જવાબ:

Linux File System ૩૮ ડિરેક્ટરીથી શરૂ થતું હાયરાર્કિકલ ડિરેક્ટરી સ્ટ્રક્ચર અનુસરે છે.



આકૃતિ 5. Linux File System Hierarchy

## કોષ્ટક 28. Important Directories

ડિરેક્ટરી	હેતુ
/	રૂટ ડિરેક્ટરી, હાયરાર્કીની ટોચ
/bin	આવશ્યક યુઝર કમાન્ડ્સ
/etc	સિસ્ટમ કોન્ફિગરેશન ફાઇલ્સ
/home	યુઝર હોમ ડિરેક્ટરીઓ
/var	વેરિએબલ ડેટા (લોગ્સ, મેઇલ)
/usr	યુઝર પ્રોગ્રામ્સ અને યુટિલિટીઝ
/tmp	ટેમ્પરરી ફાઇલ્સ

## મેમરી ટ્રીક

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

## પ્રશ્ન 5(c) [7 ગુણ]

ઓપરેટિંગ સિસ્ટમ સિક્યોરિટી નીતિઓ અને પ્રક્રિયાઓ સમજાવો.

## જવાબ

જવાબ:

**Security Policies** સિસ્ટમ રિસોર્સિસ અને ડેટાને સુરક્ષિત રાખવા માટેના નિયમો અને માર્ગદર્શિકા ડિફાઇન કરે છે.

## કોષ્ટક 29. Security Policy Components

કમ્પોનન્ટ	વર્ણન
Access Control Policy	કોણ કયા રિસોર્સિસ એક્સેસ કરી શકે
Password Policy	મજબૂત પાસવર્ડ્સ માટેની આવશ્યકતાઓ
Audit Policy	કઈ પ્રવૃત્તિઓ મોનિટર અને લોગ કરવી
Backup Policy	ડેટા બેકઅપ અને રિકવરી પ્રક્રિયાઓ
Incident Response	સિક્યોરિટી બ્રીચ હેન્ડલ કરવાના સ્ટેપ્સ

## Security Procedures:

- **Authentication:** મલ્ટિ-ફેક્ટર ઓથેન્ટિકેશન, પાસવર્ડ જટિલતા
- **Authorization:** લીસ્ટ પ્રિવિલેજનો સિદ્ધાંત, રોલ-બેઝ્ડ એક્સેસ
- **Monitoring:** લોગ એનાલિસિસ, ઇન્ટ્રુઝન ડિટેક્શન

## મેમરી ટ્રીક

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

## પ્રશ્ન 5(a) OR [3 ગુણ]

ક્રિટિકલ સેક્શન સમજાવો.

## જવાબ

જવાબ:

**Critical Section** એ કોડ સેગમેન્ટ છે જ્યાં પ્રોસેસ શેડ રિસોર્સિસ એક્સેસ કરે છે જે એક્સાથે એક્સેસ થવા જોઈએ નહીં.

કોષ્ટક 30. Critical Section Properties

પ્રોપર્ટી	વર્ણન
<b>Mutual Exclusion</b>	એક સમયે માત્ર એક પ્રોસેસ ક્રિટિકલ સેક્શનમાં
<b>Progress</b>	આગળા પ્રોસેસની પસંદગી અનિશ્ચિત સમય માટે મોકૂફ ન મૂકવી
<b>Bounded Waiting</b>	અન્ય પ્રોસેસિસ ક્રિટિકલ સેક્શનમાં એન્ટર કરવાની સંખ્યા પર મર્યાદા

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);

```

## મેમરી ટ્રીક

“MPB - Mutual exclusion, Progress, Bounded waiting”

## પ્રશ્ન 5(b) OR [4 ગુણ]

લિનક્સ ફાઇલ સિસ્ટમના પ્રકારો સમજાવો.

## જવાબ

જવાબ:

**Linux File Systems** ઓર્ગેનાઇઝ અને ડેટા સ્ટોરેજ મેનેજ કરે છે.

કોષ્ટક 31. Linux File System Types

ફાઇલ સિસ્ટમ	વર્ણન
<b>ext4</b>	ચોથું એક્સટેન્ડેડ ફાઇલ સિસ્ટમ, સૌથી સામાન્ય
<b>XFS</b>	ઉચ્ચ પર્ફોર્મન્સ જર્નલિંગ ફાઇલ સિસ્ટમ
<b>Btrfs</b>	એડવાન્સ્ડ ફીચર્સ સાથે B-ટ્રી ફાઇલ સિસ્ટમ
<b>ZFS</b>	બિલ્ટ-ઇન RAID સાથે ઝેટાબાઇટ ફાઇલ સિસ્ટમ

**ext4 Features:**

- **Journaling:** સિસ્ટમ ક્રેશ પછી ઝડપી રિકવરી
- **Large file support:** 16TB સુધીની ફાઇલ્સ



- **Backwards compatibility:** ext2/ext3 પાર્ટિશન-સ માઉન્ટ કરી શકે છે

### મેમરી ટ્રીક

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

## પ્રશ્ન 5(c) OR [7 ગુણ]

પ્રોટેક્શન મેકેનિઝમની જરૂરિયાત અને વિવિધ પ્રોટેક્શન ડોમેઇન સમજાવો.

### જવાબ

**જવાબ:**

**Protection Mechanism** પ્રોસેસિસને એકબીજા અને સિસ્ટમ રિસોર્સિસ સાથે દખલગીરી કરવાથી અટકાવે છે.

**Need for Protection:**

- **Resource sharing:** મલ્ટિપલ યુઝર્સ/પ્રોસેસિસ સમાન રિસોર્સિસ એક્સેસ કરે છે
- **Error containment:** બગસને સંપૂર્ણ સિસ્ટમને અસર કરતા અટકાવવા
- **Security enforcement:** એક્સેસ કંટ્રોલ નીતિઓ લાગુ કરવી

### કોષ્ટક 32. Protection Domains

ડોમેઇન પ્રકાર	વર્ણન
User Domain	યુઝર પ્રોસેસિસ માટે મર્યાદિત એક્સેસ રાઇટ્સ
Kernel Domain	સિસ્ટમ રિસોર્સિસ પર સંપૂર્ણ એક્સેસ
System Domain	સિસ્ટમ સર્વિસિસ માટે મધ્યમ પ્રિવિલેજિસ

### કોષ્ટક 33. Access Rights

Right	વર્ણન
Read	રિસોર્સનું કન્ટેન્ટ જોવું
Write	રિસોર્સ કન્ટેન્ટ સુધારવું
Execute	પ્રોગ્રામ ચલાવવું કે ડિરેક્ટરીમાં પ્રવેશ

### મેમરી ટ્રીક

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