

Linux Operating System (4331602) - Winter 2023 Solution

Milav Dabgar

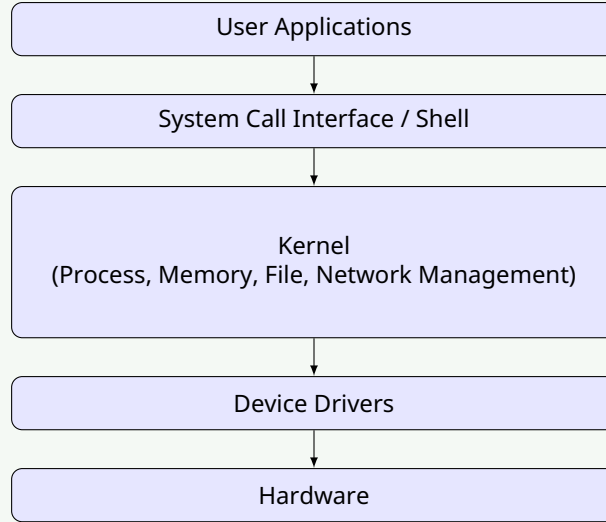
January 16, 2024

પ્રશ્ન 1(a) [3 ગુણ]

Linux ના આર્કિટેક્ચર દોરો અને સંક્ષિપ્તમાં વિવિધ સ્તરો સમજાવો.

જવાબ

જવાબ:
Linux Architecture:



આકૃતિ 1. Linux Architecture

Layers:

- **User Space:** વપરાશકર્તા applications અને system utilities સમાવે છે.
- **System Call Interface:** user programs અને kernel વચ્ચે interface પ્રદાન કરે છે.
- **Kernel Space:** મૂળ operating system જે process, memory, વગેરેનું management કરે છે.
- **Hardware:** Computer system ના physical components.

મેમરી ટ્રીક

“USKDH - Users System Kernel Drives Hardware”

પ્રશ્ન 1(b) [4 ગુણ]

રેસની સ્થિતિ શું છે? યોગ્ય ઉદાહરણ સાથે સમજાવો.

જવાબ

જવાબ:

કોષ્ટક 1. Race Condition

પાસું	વિવરણ
વ્યાખ્યા	અનેક processes એકસાથે shared resource ને access કરે છે.
સમસ્યા	timing dependency ને કારણે અનિશ્ચિત પરિણામો.
ઉદાહરણ	બે transactions દ્વારા બેંક account balance ને update કરવું.

Example Scenario:

- 1. **Process A:** balance = 1000 વાંચે છે, 100 ઉમેરે છે (1100 લખવા તૈયાર).
- 2. **Process B:** balance = 1000 વાંચે છે, 50 બાદ કરે છે (950 લખવા તૈયાર).
- 3. **પરિણામ:** અંતિમ balance 1050 ને બદલે 1100 અથવા 950 હોઈ શકે.

મેમરી ટ્રીક

“RRRR - Race Results Random Resources”

પ્રશ્ન 1(c) [7 ગુણ]

વિવિધ પ્રકારની ઓપરેટિંગ સિસ્ટમોની યાદી બનાવો. મલ્ટિપ્રોગ્રામિંગ ઓપરેટિંગ સિસ્ટમના કાર્યને યોગ્ય ઉદાહરણ સાથે સમજાવો.

જવાબ

જવાબ:

Types of Operating Systems:

કોષ્ટક 2. Types of Operating Systems

Type	Characteristics
Batch	Jobs user interaction વિના groups માં process થાય છે.
Time-sharing	અનેક વપરાશકર્તાઓ એકસાથે system share કરે છે.
Real-time	Operations માટે સખત time constraints હોય છે.
Distributed	Networked processors વચ્ચે computations વહેંચાયેલ હોય છે.
Multiprogramming	CPU utilization માટે memory માં અનેક programs રાખવામાં આવે છે.

Multiprogramming Working:

- **Memory Management:** Main memory માં એકસાથે અનેક jobs load થાય છે.
- **CPU Utilization:** જ્યારે એક job I/O માટે રાહ જુએ, ત્યારે CPU બીજી job પર switch કરે છે.
- **Context Switching:** OS હાલની job નું state save કરે છે અને બીજી job load કરે છે.

ઉદાહરણ: એક user વેબ browser, music player, અને word processor એકસાથે ચલાવે છે. જ્યારે browser network data ની રાહ જુએ, ત્યારે CPU music player ચલાવે છે.

મેમરી ટ્રીક

“MPMP - Multiple Programs Maximize Performance”

પ્રશ્ન 1(c) OR [7 ગુણ]

વિવિધ પ્રકારની ઓપરેટિંગ સિસ્ટમોની યાદી બનાવો. બેચ ઓપરેટિંગ સિસ્ટમ્સ વિગતવાર સમજાવો.

જવાબ

જવાબ:

Batch Operating System:

- **Job Collection:** Jobs (program + data) offline collect થાય છે.
- **Batching:** Operator સમાન jobs ને batches માં group કરે છે.
- **Sequential Execution:** Batch ની દરેક job એક પછી એક execute થાય છે.
- **No Interaction:** Execution દરમિયાન user interact કરી શકતો નથી.

મેમરી ટ્રીક

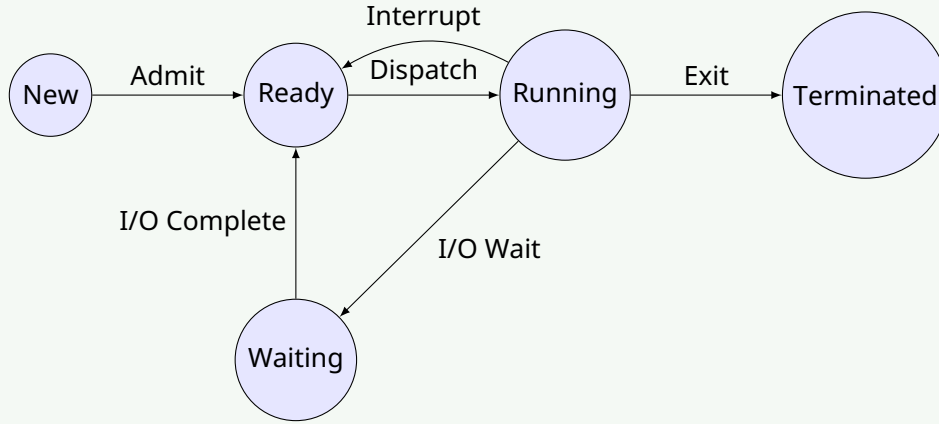
``BBBB - Batch Brings Better Business"

પ્રશ્ન 2(a) [3 ગુણ]

પ્રક્રિયા જીવન ચક્ર દોરો અને સમજાવો.

જવાબ

જવાબ:



આકૃતિ 2. Process State Diagram

States:

- **New:** Process બની રહી છે.
- **Ready:** Process processor મળવાની રાહમાં છે.
- **Running:** Instructions execute થઈ રહી છે.
- **Waiting:** Process કોઈ event (I/O) ની રાહમાં છે.
- **Terminated:** Process execution પૂરું કર્યું છે.

મેમરી ટ્રીક

``NRWRT - New Ready Waiting Running Terminated"

પ્રશ્ન 2(b) [4 ગુણ]

ડેડલોકને વ્યાખ્યાયિત કરો અને ડેડલોક થવા માટે જરૂરી શરતોની ચર્ચા કરો.

જવાબ

જવાબ:

Deadlock: એવી પરિસ્થિતિ જ્યાં processes blocked હોય છે કારણ કે દરેક process resource hold કરે છે અને બીજા resource ની રાહ જુએ છે.

શરતો (Coffman Conditions):

1. **Mutual Exclusion:** Resource share કરી શકાતો નથી.
2. **Hold and Wait:** Process resource hold કરી બીજાની રાહ જુએ છે.
3. **No Preemption:** Resource જબરદસ્તીથી લઈ શકાતો નથી.
4. **Circular Wait:** Processes circular chain માં એકબીજાની રાહ જુએ છે.

મેમરી ટ્રીક

“MHNC - My Hold Never Circles”

પ્રશ્ન 2(c) [7 ગુણ]

રાઉન્ડ રોબિન અલ્ગોરિધમનું વર્ણન કરો. આપેલ ડેટા માટે ગેન્ટ ચાર્ટ સાથે સરેરાશ રાહ જોવાનો સમય અને સરેરાશ ટર્ન-અરાઉન્ડ સમયની ગણતરી કરો. સંદર્ભ સ્વિચ = 01 ms અને ક્વોન્ટમ સમય = 05 ms ધ્યાનમાં લો.

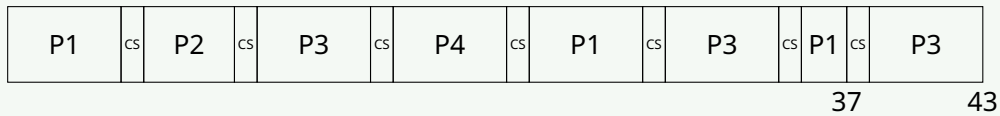
જવાબ

જવાબ:

Round Robin: Preemptive scheduling જેમાં દરેક process ને fixed time slice (quantum) મળે છે.

Given Data: Context Switch = 1ms, Quantum = 5ms.

Gantt Chart:



આકૃતિ 3. Gantt Chart (RR)

ગણતરી:

- Average Waiting Time = 16.5 ms
- Average Turnaround Time = 25.5 ms

મેમરી ટ્રીક

“RRRR - Round Robin Rotates Regularly”

પ્રશ્ન 2(a) OR [3 ગુણ]

તફાવત: CPU બાઉન્ડ પ્રક્રિયા v/s I/O બાઉન્ડ પ્રક્રિયા.

જવાબ

જવાબ:

કોષ્ટક 3. CPU vs I/O Bound

પાસું	CPU Bound	I/O Bound
Activity	High CPU computations	Frequent I/O operations
Burst Time	Long CPU bursts	Short CPU bursts
Wait States	ઓછી વાર	I/O માટે વારંવાર રાહ જુએ છે
Examples	Scientific calculation	File copy, Data processing

મેમરી ટ્રીક

“CIC - CPU Computes I/O Interacts”

પ્રશ્ન 2(b) OR [4 ગુણ]

ક્રિટિકલ સેક્શનને વ્યાખ્યાયિત કરો અને ક્રિટિકલ સેક્શન સોલ્યુશનની સામાન્ય રચનાની ચર્ચા કરો.

જવાબ

જવાબ:

Critical Section (CS): Code segment જ્યાં shared resources access થાય છે.

General Structure:

```

1 do {
2   entry section // Request permission
3   critical section
4   exit section // Release permission
5   remainder section
6 } while (true);

```

જરૂરિયાતો:

- **Mutual Exclusion:** CS માં ફક્ત એક જ process.
- **Progress:** જો CS ખાલી હોય, તો next process selection અટકવી ન જોઈએ.
- **Bounded Waiting:** Entry માટે રાહ જોવાનો સમય મર્યાદિત હોવો જોઈએ.

મેમરી ટ્રીક

“ECER - Entry Critical Exit Remainder”

પ્રશ્ન 2(c) OR [7 ગુણ]

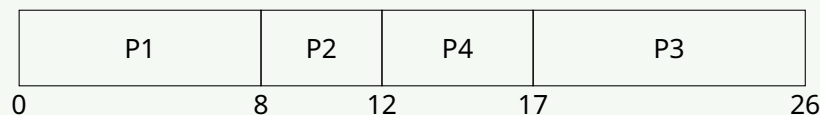
SJF અલ્ગોરિથમનું વર્ણન કરો. કોષ્ટકમાં આપેલ ડેટા માટે ગેન્ટ ચાર્ટ સાથે સરેરાશ રાહ જોવાનો સમય અને સરેરાશ ટર્ન-અરાઉન્ડ સમયની ગણતરી કરો.

જવાબ

જવાબ:

Shortest Job First (SJF): Non-preemptive algorithm જ્યાં સૌથી નાના burst time વાળી process પહેલા schedule થાય છે.

Gantt Chart:



આકૃતિ 4. Gantt Chart (SJF)

ગણતરી:

- Avg Wait = 5.75 ms
- Avg TAT = 12.25 ms

મેમરી ટ્રીક

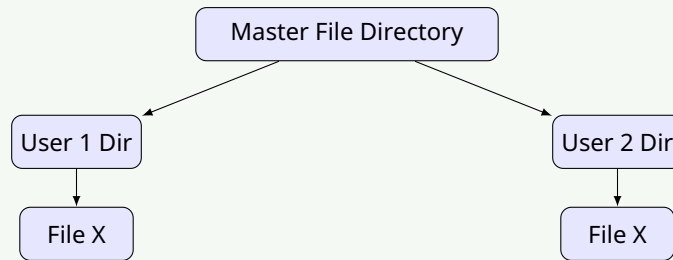
“SJSS - Shortest Jobs Start Soon”

પ્રશ્ન 3(a) [3 ગુણ]

બે-સ્તરની ડિરેક્ટરી રચના સમજાવો.

જવાબ

જવાબ:



આકૃતિ 5. Two-level Directory

વિશેષતા:

- દરેક user માટે અલગ directory (UFD).
- Name collision problem ઉકેલે છે.
- Users વચ્ચે isolation પૂરું પાડે છે.

મેમરી ટ્રીક

“TTTT - Two Tiers Tackle Troubles”

પ્રશ્ન 3(b) [4 ગુણ]

વિવિધ ફાઇલ કામગીરી સમજાવો.

જવાબ

જવાબ:

કોષ્ટક 4. File Operations

Operation	Description
Create	Space allocate કરે અને directory entry બનાવે.
Open	Metadata memory માં load કરે access માટે.
Read	Current position થી data વાંચે.
Write	Current position પર data લખે.
Delete	Space release કરે અને entry remove કરે.
Close	Internal resources free કરે.

મેમરી ટ્રીક

``CORWCD - Create Open Read Write Close Delete"

પ્રશ્ન 3(c) [7 ગુણ]

વિવિધ ફાઇલ ફાઇલવણી પદ્ધતિઓની યાદી બનાવો અને જરૂરી રેખાકૃતિ સાથે સંલગ્ન ફાઇલવણી સમજાવો.

જવાબ

જવાબ:

Methods: Contiguous, Linked, Indexed.**Contiguous Allocation:** File disk પર સતત (consecutive) blocks રોકે છે.

File A (Start:1, Len:3)

આકૃતિ 6. Contiguous Allocation

ફાયદા: સરળ (start, length), ઝડપી access. ગેરફાયદા: External fragmentation, file expand કરવામાં મુશ્કેલી.

મેમરી ટ્રીક

``CCCC - Contiguous Creates Continuous Clusters"

પ્રશ્ન 3(a) OR [3 ગુણ]

ફાઇલ સ્ટ્રક્ચરના પ્રકારોનું વર્ણન કરો.

જવાબ

જવાબ:

- **Sequential:** Records ક્રમમાં store થાય છે. સરળ પણ search ધીમું.
- **Direct/Random:** Key દ્વારા direct access. ઝડપી access.
- **Indexed:** અલગ index file data records ને point કરે છે.

મેમરી ટ્રીક

``SDI - Sequential Direct Indexed"

પ્રશ્ન 3(b) OR [4 ગુણ]

વિવિધ ફાઇલ લક્ષણો સમજાવો.

જવાબ

જવાબ:

કોષ્ટક 5. File Attributes

Attribute	Description
Name	Human-readable identifier
Type	File format (.txt, .exe)
Size	File size
Location	Device પર file location નો pointer
Protection	Access control info (R/W/X)
Time/Date	Creation, modification info

મેમરી ટ્રીક

“NTSLPT - Name Type Size Location Permissions Time”

પ્રશ્ન 3(c) OR [7 ગુણ]

વિવિધ ફાઈલ ફાળવણી પદ્ધતિઓની યાદી બનાવો અને જરૂરી રેખાકૃતિ સાથે લિંક કરેલ ફાળવણી સમજાવો.

જવાબ

જવાબ:

Linked Allocation: Files non-contiguous blocks માં store થાય છે. દરેક block next block નું pointer ધરાવે છે.



આકૃતિ 7. Linked Allocation

ફાયદા: External fragmentation નથી, file સરળતાથી વધારી શકાય. **ગેરફાયદા:** Random access ધીમું છે, pointer overhead.

મેમરી ટ્રીક

“LLLL - Links Lead Logical Locations”

પ્રશ્ન 4(a) [3 ગુણ]

પ્રોગ્રામ ધમકીઓ વ્યાખ્યાયિત કરો અને તેના પ્રકારો સમજાવો.

જવાબ

જવાબ:

Program Threats: Malicious code જે program માં embed હોય છે.

- **Trojan Horse:** ઉપયોગી દેખાય છે પણ નુકસાન કરે છે.
- **Trap Door:** Designer દ્વારા છોડવામાં આવેલ secret entry point.
- **Logic Bomb:** Code જે ચોક્કસ શરતો હેઠળ execute થાય (explode) છે.
- **Virus:** Code જે અન્ય programs માં પોતાને embed કરે છે.

મેમરી ટ્રીક

“TTLV - Trojan Trap Logic Virus”

પ્રશ્ન 4(b) [4 ગુણ]

સિસ્ટમ ઓથેન્ટિકેશન સમજાવો.

જવાબ

જવાબ:

Authentication: User identity ની verification.

પદ્ધતિઓ:

1. **Passwords:** Secret string.
2. **Biometrics:** Fingerprint, retina scan.
3. **Smart Cards:** Chip વાળા physical token.
4. **Two-Factor:** બે પદ્ધતિઓનું combination (e.g., Password + OTP).

મેમરી ટ્રીક

“PBST - Passwords Biometrics Smartcards Two-factor”

પ્રશ્ન 4(c) [7 ગુણ]

એક્સેસ કંટ્રોલ લિસ્ટને વિગતવાર સમજાવો.

જવાબ

જવાબ:

Access Control List (ACL): દરેક object સાથે જોડાયેલ list જે define કરે છે કે કયા users તેને access કરી શકે.

Structure: File X: (User A, Read), (User B, Read/Write)

ફાયદા:

- Individual objects પર ચોક્કસ control.
- Specific users માટે permission revoke કરવી સરળ.

ગેરફાયદા:

- ACLs search કરવું ધીમું હોઈ શકે.
- ઘણી files માટે ACLs manage કરવું જટિલ છે.

મેમરી ટ્રીક

“ACLU - Access Controls Limit Users”

પ્રશ્ન 4(a) OR [3 ગુણ]

સિસ્ટમ ધમકીઓ વ્યાખ્યાયિત કરો અને તેના પ્રકારો સમજાવો.

જવાબ

જવાબ:

System Threats: OS અથવા environment ને target કરે છે.

- **Worm:** Independent program જે network પર ફેલાય છે અને resources વાપરે છે.
- **Port Scanning:** Vulnerabilities શોધવા open ports detect કરવા.
- **Denial of Service:** System ને overwhelm કરી legitimate use અટકાવવું.

મેમરી ટ્રીક

“WPD - Worm Port DoS”

પ્રશ્ન 4(c) OR [7 ગુણ]

વિવિધ ઓપરેટિંગ સિસ્ટમ સુરક્ષા નીતિઓ અને પ્રક્રિયાઓની ચર્ચા કરો.

જવાબ

જવાબ:

Policies:

- **User Policy:** Strong passwords.
- **Access Policy:** Least privilege principle.
- **Data Policy:** Sensitive data નું encryption.

Procedures (પ્રક્રિયાઓ):

- **Auditing:** Logs monitor કરવા.
- **Backups:** Regular data backup.
- **Updates:** OS patching.

મેમરી ટ્રીક

“APPI - Access Password Policy Incident”

પ્રશ્ન 5(a) [3 ગુણ]

નીચેના આદેશો સમજાવો: (i) pwd (ii) cd (iii) comm

જવાબ

જવાબ:

કોષ્ટક 6. Commands

Command	Purpose
pwd	Print Working Directory.
cd	Change Directory. Folder બદલવા માટે.
comm	બે sorted files ને line-by-line compare કરે.

મેમરી ટ્રીક

“PCC - Pwd Cd Comm”

પ્રશ્ન 5(b) [4 ગુણ]

ત્રીજી ફાઇલમાં બે ફાઇલોના સમાવિષ્ટોને જોડવા માટે શેલ સ્ક્રિપ્ટ લખો.

જવાબ

Listing 1. Concatenate Files

```

1  #!/bin/bash
2  # Script to concatenate two files
3
4  echo "Enter first filename:"
5  read f1
6  echo "Enter second filename:"

```

```

7 read f2
8 echo "Enter output filename:"
9 read f3
10
11 if [ -f "$f1" ] && [ -f "$f2" ]; then
12     cat "$f1" "$f2" > "$f3"
13     echo "Files merged into $f3"
14 else
15     echo "Files not found"
16 fi

```

મેમરી ટ્રીક

``CCCC - Cat Combines Content Correctly"

પ્રશ્ન 5(c) [7 ગુણ]

આપેલ 5 અંકની સંખ્યામાં તમામ વ્યક્તિગત અંકોનો સરવાળો શોધવા માટે શેલ સ્ક્રિપ્ટ લખો.

જવાબ

Listing 2. Sum of Digits

```

1 #!/bin/bash
2 # Sum of 5 digits
3
4 echo "Enter 5 digit number:"
5 read n
6
7 if [ ${#n} -ne 5 ]; then
8     echo "Please enter 5 digits"
9     exit 1
10 fi
11
12 sum=0
13 while [ $n -gt 0 ]
14 do
15     rem=$((n % 10))
16     sum=$((sum + rem))
17     n=$((n / 10))
18 done
19
20 echo "Sum of digits: $sum"

```

મેમરી ટ્રીક

``SSSS - Sum Separates Single Symbols"

પ્રશ્ન 5(a) OR [3 ગુણ]

નીચેના આદેશો સમજાવો: (i) man (ii) mkdir (iii) grep

જવાબ

જવાબ:

કોષ્ટક 7. More Commands

Cmd	Purpose
man	Manual. Help display કરે છે.
mkdir	Make Directory. નવું folder બનાવે છે.
grep	Global Regular Expression Print. Files માં text search કરે છે.

મેમરી ટ્રીક

``MMG - Manual Make Grep``

પ્રશ્ન 5(b) OR [4 ગુણ]

ફિબોનાચી શ્રેણી જનરેટ કરવા અને પ્રદર્શિત કરવા માટે શેલ સ્ક્રિપ્ટ લખો.

જવાબ

Listing 3. Fibonacci Series

```

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

```

મેમરી ટ્રીક

``FFFF - Fibonacci Follows Forward Formula``

પ્રશ્ન 5(c) OR [7 ગુણ]

આપેલ string palindrome છે કે કેમ તે નિર્ધારિત કરવા માટે શેલ સ્ક્રિપ્ટ લખો.

જવાબ

Listing 4. Palindrome Check

```
1  #!/bin/bash
2  # Palindrome Check
3
4  echo "Enter string:"
5  read str
6  len=${#str}
7  rev=""
8
9  for (( i=len-1; i>=0; i-- ))
10 do
11     rev="$rev${str:$i:1}"
12 done
13
14 if [ "$str" == "$rev" ]; then
15     echo "Palindrome"
16 else
17     echo "Not Palindrome"
18 fi
```

મેમરી ટ્રીક

``PPPP - Palindromes Proceed Perfectly Parallel``