NAME:__________ STD.:______ DIV.:______ DATE : PAGE :

OPERATING SYSTEM ASSIGNMENT : 5

CAPSTONE ASSIGNMENT

RUCCHIKA KAPOOR

2301010240

PART-A

Q1 Hardware provides raw CPU, memory & devices but lacks safe, convinient coordination. The OS supplies essential abstractions such as:

→ PROCESS MANAGEMENT:
CPU scheduling, context scheduling etc. this allows multitasking and isolation

→ MEMORY MANAGEMENT:
virtual memory, protection, paging and swapping etc. this provides process isolation and efficient use of physical RAM.

→ I/O MANAGEMENT:
device drivers, buffering, uniform APIs etc this hides device complexity. Together these services enable portable, secure and better resource utilization.

Q2 MONOLITHIC:
Most services in kernel. High performance but lower modularity and ~~hardware~~ harder to maintain.

NAME:________ STD.:______ DIV.:______ DATE: PAGE:

LAYERED:

OS split into layers with well defined interfaces but can limit flexibility and performance.

MICROKERNEL:

minimal kernel, ~~os~~ services as user process. Slight overhead from IPC. It's the best among all for web application because of its maintainability & reliability.

Q3 Threads are more efficient than processes in the following ways:

→ Threads share address space & many resources, so creating & switching threads is cheaper.

→ Processes have seperate PCB & require heavier context scheduling overhead & kernel involvement.

→ However threads risk safety and need synchronization on multi-core, kernel level threads may still require kernel structures.

Q4 processes require → 12MB, 18MB, 6 MB

blocks → 20MB, 10MB, 15MB

First fit:

20
$\boxed{12}$
Block 1

leftover = 8MB

13
$\boxed{15}$
Block 2

NAME:________________ STD.:________ DIV.:________

DATE :
PAGE :

Anywhere as blocks are very small (8, 10, 15). So first fit fails because the memory gets fragmented.

BEST FIT:

20	18	Block 1
10	6	B2
15	12	B3

All processes fit successfully, so best-fit works.

Q5.

	BT	AT	CT	TAT	W
P_1	5	0	5	5	0
P_2	3	1	8	7	4
P_3	8	2	16	14	6
P_4	6	3	22	19	13
				35	23

FCFS:

P_1	P_2	P_3	P_4

0 5 8 16 22

avg. waiting ⇒ 23/4 = 5.75
avg. TAT ⇒ 35/4 = 11.25

SJF:

	BT	AT	CT	TAT	WT
P_1	5	0	5	5	0
P_2	3	1	8	7	4
P_3	8	2	14	11	5
P_4	6	3	22	20	12

P_1	P_2	P_3	P_4

0 5 8 14 22

avg. WT = 5.25
avg. TAT = 10.75

Round Robin

	BT	AT	CT	TAT	WT
P_1	5	0	16	16	11
P_2	3	1	7	6	3
P_3	8	2	20	18	10
P_4	6	3	22	19	13

avg. WT = 9.25
avg. TAT = 14.75

P_1	P_2	P_3	P_4	P_1	P_3	P_4

0 4 7 11 15 16 20 22

SJF gives the best avg TAT and slightly better waiting man FCFS.

Q6 sequence = 2, 1, 4, 2, 3, 4
frames = 3

FIFO:

2	1	4	2	3	4
2	2	2	2	3	3
	1	1	1	1	1
		4	4	4	4
✓	✓	✓	X	✓	X

fault = 4

DATE :

PAGE :

NAME: 2 1 4 2 3 STD.: 3 DIV.: 4

LRU :

2	1	4	2	3	2
2	2	2	2	2	2
	1	1	1	3	3
		4	4	4	4
✓	✓	✓	x	✓	x

fault = 4

Q7

```
with open ('/tmp/os_demo.txt','w') as f:
    f.write ('os demo\n')
with open ('/tmp/os_demo.txt','r') as f:
    print (f.read()).
```

This shows how system calls mediate user/kernel boundary & how OS handles file ~~di~~ descriptions, caching and access control.

Q8 Two critical issues are:

→ Consistency, performance keeping replicas consist across sites while providing low latency access.

→ Fault tolerance & concurrency control: handling partial failures, network partitions etc.

ARCHITECTURAL APPROACHES THAT CAN BE USED are:

→ uses replication w/ leases & consistency levels.
→ Implement a distributed ~~mediate~~ metadata service & data servers for chunk storage.
This splits metadata from bulkdata, enabling scalability.

Satinder
21/11/25