Ishika
2301010242
Btech cse (Sec-4)

CLASSTIME Pg. No.
Date / /

Operating System

Assignment - 1

PART A (Short Answer Type):

1. Despite the evolution of hardware, why do modern systems still rely heavily on operating systems?

⇒ Modern systems still need OS bcoz it manages resources like CPU, memory and I/O while providing abstractions such as processes & files. It ensures security, multitasking & portability so applications can run without directly handling complex hardware.

2. You are asked to design an OS for a wearable health device that monitors heart rate. Which type of operating system would be most suitable & Why?

⇒ A real-time embedded OS (RTOS) is most suitable. It offers predictable response to sensor inputs, efficient scheduling and low power consumption. These features are critical for a wearable device that must react quickly to heart-rate data and operate on limited battery.

3. Given the choice to build a new OS kernel for a performance-critical environment, which structure would you avoid (Monolithic, Layered, Microkernel) and Why?

⇒ Avoid microkernel architecture, since its reliance on message passing and frequent context switchess adds overhead. In performance-

critical systems, this reduces efficiency compared to monolithic or layered kernels, which allow faster direct calls.

4. A developer claims that OS structure doesn't matter as long as processes run. Support or refute this claim with reasoning.

⇒ Yes, OS structure matters. It impacts performance, security, modularity & fault isolation. For example, monolithic kernels can be fast but less reliable, while microkernels improve isolation but add overhead. Thus, structure influences efficiency & stability, not just process execution.

5. While debugging, you find a process switching error.

(i) Explain how analysing the Process Control Block (PCB) can point to misinitialized registers or incorrect states

⇒ PCB contains saved registers, PC and state; errors here reveal misinitialized values or incorrect states.

(ii) If a task is moved unexpectedly from running to waiting, what precisely does context switching involve?

⇒ context switching saves the current process state, updates its PCB, loads the next process state, and resumes execution.

(iii) The OS needs to allocate I/O resources mid execution. Which type of system call would you use (e.g. blocking, non-blocking, synchronous, asynchronous) & why?

⇒ Asynchronous non-blocking calls should be used, allowing execution to continue while I/O is handled, ensuring responsiveness.

PART - B (Application / Numerical Based):

6. (a) Total time = Save + Load + Scheduler
= 2ms + 3ms + 1ms
= 6ms

(b) Each context switch adds overhead with no useful work. Too many switches reduce CPU time for processes, lowering throughput & increasing response time.

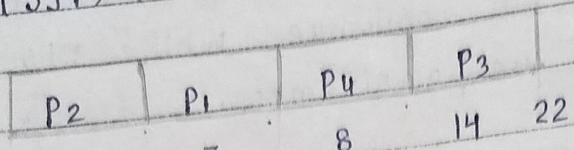
7. Single - thread time = 40 sec
Execution Time ≈ 40/n (With n threads in ideal conditions)
Example (n = 4) = 40/4 = 10 sec.
Multithreading improves performance by running tasks in parallel and overlapping I/O & computation reducing total execution time.

8. (a) FCFS (First - Come, First Served)

P1	P2	P3	P4
0	5	8	16	22

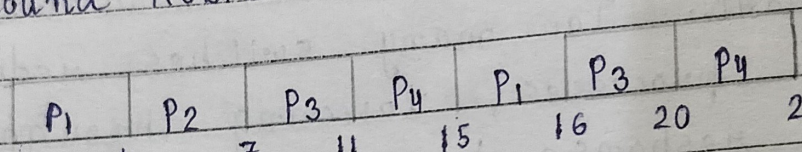
P1 executes from 0 ms to 5 ms (burst time = 5ms)
P2 executes from 5ms to 8ms (burst time = 3ms)
P3 executes from 8ms to 16ms (burst time = 8 ms)
P4 executes from 16ms to 22ms (burst time = 6 ms).

(SJF) (Non - Preemptive Shortest Job First):



The order of execution is based on burst time.
P2 executes from 0ms to 3ms (burst time = 3ms)
P1 executes from 3ms to 8ms (burst time = 5ms)
P4 executes from 8ms to 14ms (burst time = 6ms)
P3 executes from 14ms to 22ms (burst time = 8ms)

Round Robin (Quantum = 4ms)



Processes are executed in cyclic manner with each process getting a time slice of 4ms.

Process P1 is executed for 4ms (remaining burst time = 1ms)
Process P2 is executed for 3ms (completed)
Process P3 is executed for 4ms(remaining burst time = 4ms)
Process P4 is executed for 4ms (remaining burst time = 2ms)
Process P1 is executed for 1ms (completed)
Process P3 is executed for 4ms (completed)
Process P4 is executed for 2ms (completed)

(b) FCFS:
Waiting Times : $P_1 = 0$, $P_2 = 5$, $P_3 = 8$, $P_4 = 16$
Turnaround times : $P_1 = 5$, $P_2 = 8$, $P_3 = 16$, $P_4 = 22$
Average waiting time = $(0+5+8+16)/4 = 7.25$ ms
Average turnaround time = $(5+8+16+22)/4 = 12.75$ ms

SJF :

Waiting times : $P_2 = 0$, $P_1 = 3$, $P_4 = 8$, $P_3 = 14$

Turnaround times : $P_2 = 3$, $P_1 = 8$, $P_4 = 14$, $P_3 = 22$

Average waiting time = $(0+3+8+14)/4 = 6.25$ ms

Average turnaround time = $(3+8+14+22)/4 = 11.75$ ms

Round Robin ($q = 4$) :

Finish times : $P_1 = 16$, $P_2 = 7$, $P_3 = 20$, $P_4 = 22$

Turnaround times : finish - arrival = same as Finish

$P_1 = 16$, $P_2 = 7$, $P_3 = 20$, $P_4 = 22$

Waiting times : turnaround - burst

$P_1 = 11$, $P_2 = 4$, $P_3 = 12$, $P_4 = 16$

Average waiting time = $(11+4+12+16)/4 = 10.75$ ms

Average turnaround time = $(16+7+20+22)/4 = 16.25$ ms

(c) SJF gives lowest waiting and turnaround times so it balances throughput and efficiency best.

9. (i) Migration to Cloud Infrastructure :

(a) A microkernel or layered OS architecture is most suitable because it provides scalability, modularity, and stronger isolation. This helps maintain security and stability when multiple cloud services run simultaneously.

(b) Virtual machines (VMs) support cloud migration by offering isolation (one VM's failure doesn't affect others), management (easy snapshots, migration and deployment) & resource optimization (efficient sharing of CPU, memory & storage across workloads).

(ii) Smart Home IOT System

(a) The OS ensures timely responses by using priority-based scheduling so critical tasks like intrusion detection preempt lower-priority ones. IPC mechanisms (e.g. message queues or signals) allows fast communication between devices and the central controller.

(b) Real-time algorithms such as Earliest Deadline First (EDF) or Rate Monotonic Scheduling (RMS) are suitable because they guarantee deadlines for high-priority tasks while still allowing less critical tasks to execute efficiently.