- PRATEEK KUMAR PRASAD
- 2301420018
- B.Tech CSE (DS)

Page No.
Date

Operating System
Assignment - 1

1. Modern system still rely heavily on OS despite the rapid evolution of hardware because OS acts as bridge b/w the hardware and the user / application software. Even though hardware has become faster and more powerful, it cannot function efficiently on its own w/o an OS to manage and coordinate its resources. OS remains essential : Resource management, Abstraction of hardware, complexity, multi-tasking, concurrency, security and protection.

2. For designing an OS for a wearable health device that monitors heart rates, the most suitable type of OS would be a real time OS.

why: need for real time processing small size and efficiency continuous sensors monitoring reliability and softly networking and communication.

3. To build a new OS kernal for a performance critical environment, the structure I would avoid is the micro kernal architecture.

why: high context switching overhead increased latency.

4 - Saying "OS structure doesn't matter as long as process run" ~~understand~~ understates how much the kernel design affects performance, reliability, security, real-time behaviour, maintainability and even what kinds of applications the system can support.

Key reasons :- Performance
Predictability and real-time guarantees
Security and attack surface

5 - i) The Process Control Block (PCB) is a data structure maintained by the OS that stores all the information about a process

ii) When a process moves unexpectedly from running to waiting, the OS performs a context switch, which involves saving the current process state and loading the next process state

iii) When the OS needs to allocate, 2/IO resources while process is still executing, the correct type of system call depends on however you want the process to behave

Given data : Time to save state = 2ms
Time to load state = 3ms
Schedule overload = 1ms

a) Total Context Switching Time = 2 + 1 + 3
= 6 ms

b) Impact on Multitasking performance
- Increased CPU overhead
- Reduced throughput
- High latency for processes
- Multitasking trade-off

Given : $T_1 = 40$ s
Threads per process = 2

a) Execution Time (Estimated)

$$T_2 = \frac{T_1}{2} = \frac{40}{2} = 20s$$

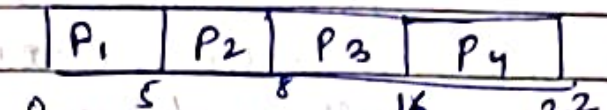
b) Multithreading improves performance
- Parallelism
- Better CPU utilization
- Latency hidden
- Lower overhead than processes
- Responsiveness

8

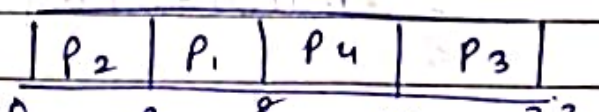
Process	Time (ms)
P_1	5
P_2	3
P_3	8
P_4	6

a) Gantt Chart

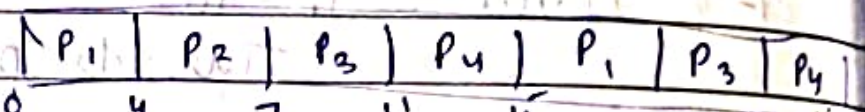
FCFS →



SJF →



Round Robin →



b)

FCFS	SJF	Round Robin
$P_1 = 5-5=0$	$P_1 = 3-3=0$	$P_1 = 16-5=11$
$P_2 = 8-3=5$	$P_2 = 8-5=3$	$P_2 = 7-3=4$
$P_3 = 16-8=8$	$P_3 = 14-6=8$	$P_3 = 20-8=12$
$P_4 = 22-6=16$	$P_4 = 22-8=14$	$P_4 = 22-6=16$
Avg wait Time	Avg wait Time	Avg Wait Time
$= \frac{0+5+8+16}{4}$	$\frac{0+3+8+14}{4}$	$\frac{11+4+12+16}{4}$
$= 7.2$ ms	$= 6.25$ ms	$= 10.75$ ms

Avg Turn Around Time

$\frac{5+8+11+22}{4}$ $\frac{3+8+14+22}{4}$ $\frac{16+7+20+22}{4}$

$= 12.75$ ms $= 11.75$ ms $= 16.25$ ms

c) Throughput here is essentially the no of process completed per unit time. Since the makespan is 22 ms for all three scheduling schemes in this scenario, throughput is the same for all.

SJF gives the best average turn around time and the best average waiting time among the three for this scenario. While throughput remains equal.

Therefore SJF best balances through and turn around for these jobs and arrivals

a- i) Cloud Migration and OS architecture
a) Best choice OS architecture - Microkernel architecture
b) Role of virtual machines in Cloud migration
↳ Isolation
↳ Efficient resources management
↳ Easy migration and scalability

ii) Smart Home System with IOT Devices
a) Role of Scheduling and IPC
↳ Process Scheduling
↳ Inter-process communication

b) Suitable scheduling algorithm:

→ Priority Scheduling: Ensures urgent tasks like intrusion detection interrupt lower-priority tasks immediately.

→ Earliest Deadline First: Ideal for real-time tasks where deadline (EDF) must be met.

→ Round Robin (RR): Can be for periodic tasks like thermostat update while still prioritizing urgent tasks.