Assignment-1
Operating System
(BTech CSE) (Sec L)

Gurleen Kaur
2201010207

Que1. Modern systems still rely heavily on operating system because Hardware, no matter how advanced, cannot function efficiently without software to manage & coordinate its use. The OS acts as a bridge between hardware & users programs, ensuring usability, security & performance.

Key Reasons:

1. Resource Management :- The OS allocates CPU, memory, storage, and I/O devices so multiple applications can run smoothly without conflict.
2. Abstraction :- It hides hardware complexibility and provides a simple interface.
3. Multitasking & Scheduling :- OS ensures fair & efficient execution of multiple processes.
4. Security & Protection :- It controls access, prevents unauthorized tools, and safeguards data.
5. Portability & Compatability :- Applications can run on diff. hardware without rewriting code, thanks to the OS interface.

Que 2. A wearable health device requires constant monitoring of heart signals and immediate processing of the data. Delays or missed readings could lead to inaccurate results. A Real-Time operating system provides:-

a) Deterministic Response
b) Low power consumption
c) Multitasking
d) Reliability & Safety.

Que 3. A microkernel keeps only minimal services (like IPC, Sechedulling) in the kernel, while device drivers, file system, etc. run in User Space.
- This design improves modularity, Security, fault isolation, but it requires frequent context switching & inter process communication b/w kernel & user space.
- These overheads significantly reduce performance especially in real-time or high speed system.
- In Contrast, Monolithic kernels & Layered kernels are more suitable for raw performance needs.

Que 4. Route-structure does matter. Running process is necessary but not sufficient; the OS architecture strongly affects how well those processes run in real world.
- Kernel design changes ~~Small~~ Syscall & Content-Switch Costs.
- Clear isolation reduces attack surface & limit damage from bugs.
- Fault isolation prevents one component Crash from taking down the whole system.
- RTOS structure are required for-time Critical tasks.

Que 5.
i) The PCB stores process state, registers, Progr counter, stack pointer, and I/O status. If registers on state PC in the PCB are wrong: it indicates misinitialized registers or improper state saving during switching.

ii) A Context Switch stores the content process CPU state into the PCB, updates its state to waiting then loads the saved state of the next scheduled process from its PCB into the CPU.

iii) For mid-execution I/O allocation, use a blocking synchronous system call because the process must wait until the OS allocates & confirms I/O resources to avoid conflicts or inconsistent execution.

Que 6. Given
- Save State Time = 2ms
- Load State Time = 3ms
- Schedular Overhead = 1ms

a) Total Context Switching Time

Total Time = Save State + Schedular Overhead + Load State

= 2ms + 1ms + 3ms = **6ms**

b) Impact on Multitasking Performance.
- Context Switch Time is pure overhead
- Higher switching time reduces CPU efficiency since the CPU spends time managing tasks instead of executing them.
- With frequent Switches, system responsiveness improves, but may drop.

Que 7

i) $T_{multi} = \frac{T_{single}}{n}$

So

$T_{multi} = 40/n$ Seconds

For example: with 2 Threads →
40 ÷ 2 = 20 sec
4 Threads →
40 ÷ 4 = 10 sec
8 Threads →
40 ÷ 8 = 5 sec

2. How multithreading improves Performance
- Parallelism: Tasks are divided among multiple threads, running concurently on multiple cores.
- Resource Utilization: While one thread waits I/O, another can keep the CPU busy.
- Reduce responsive time: Application fast faster since multiple operations.

Que 8: a) FCFS ($P_1 \rightarrow P_2 \rightarrow P_3 \rightarrow P_4$)
Time access 0 — 5 — 8 — 16 — 22
Gantt: | P1 (0-5) | P2 (5-8) | P3 (8-16) | P4 (16-22)

Non-Preemptive SJF (Shortest Job first)
Order by burst: P2(3), P1(5), P4(6), P3(8)
Time axis — 0 — 3 — 8 — 14 — 22.
Gant: | P2 (0-3) | P1 (3-8) | P4 (8-14) | P3 (14-22)

Round Robin (quantum = 4ms)
Simulation (all arrive at 0):

0-4: P1	11-15: P4
4-7: P2	15-16: P1
7-11: P3	16-20: P2
	20-22: P3

(b) Waiting Time & Turnaround Time:

Defination: TAT = Completion time - Arrival time (=0)

waiting = TAT - Burst

(FCFS) = Completion :- P1 = 5, P2 = 9, P3 = 16, P4 = 22.

TATs :- P1 = 5; P2 = 8, P3 = 16, P4 = 22.

waitings :- P1 = 0, P2 = 5, P3 = 8, P4 = 16.

Averages :- • Avg waiting = (0+5+8+16)/4
= 29/4 = 7.25ms

• Avg Turnaround = (5+8+16+22)/4
= 51/4 = 12.75 ms

Non-preemptive SJF =

- Completion : P2 = 3, P1 = 8, P4 = 14, P3 = 22
- TATs : P2 = 3, P1 = 8, P4 = 14, P3 = 22
- waiting : P2 = 0, P1 = 3, P4 = 8, P3 = 14

(c) All three finish the last Job at time 22 ms so throughput is the same for this dataset : 4 Jobs / 22ms ≈ 0.1818 Jobs / ms.

For turnaround & average waiting SJF is best here:

- SJF : Avg TAT = 11.75 ms. Avg waiting = 6.25 ms
- FCFS : Avg TAT = 12.75 ms Avg waiting = 7.25 ms

Que 9. i) cloud ~~Arisa~~ Migration.

a) Microkernel architecture → Best for Scalability & Security since only core services run in kernel space while drivers and services run in user space → reduces attack surface, easier scaling in distributed / virtualized setup.

b) Virtual Machine → Provide isolation, ~~resource~~ management & resource optimization.

ii) Smart Home IOT

a) OS uses process scheduling to assign CPU time based on priority. eg → intrusion detection vs lighting. IPC ensures device / controllers share data quickly & safely.

b) Suitable Scheduling is ensures urgent tasks are executed effective. for Periodic sensor data.