Alya Kumari
2301010223

Operating systems Assignment no 1

PART - A

Ans 1. Modern systems rely on operating systems because they provide a crucial layer of abstraction between complex hardware and the software that runs on it. The OS manages resources like the CPU, memory, and input/output devices, making them easier for applications to use. It also provides essential services such as user interface, multitasking, capabilities and security features that are not inherent to hardware itself.

Ans 2 A Real time operating system is the most suitable choice for a wearable health device. These devices require immediate and predictable processing of data. An RTOS is designed to handle data as it is generated, without any delays which is critical for a heart rate monitor to provide accurate, real time feedback and alerts for any health abnormalities.

Ans 3 for a performance - critical environment a layered Kernel structure is generally avoided. The multiple layers of abstraction in this architecture can create significant overhead, slowing down communication between different parts of system and reducing overall performance compared to a monolithic or even a microkernel architecture.

Ans 4 This claim is incorrect. The structure of an OS has a significant impact on its performance, stability, security and maintainence. for instance, in a monolithic kernal, a single faculty drivers can cause the entire system to crash. In contrast, a microkernel architecture isolates services so the failures of one component is less likely to affect the entire system.

Ans 5(i) The process control block (PCB) is a data structure that stores all the information the OS needs about a process. When a process switching error occurs, analyzing the PCB can reveal several issues.

(ii) Context switching :- This involves saving the context of the current process to its PCB, updating its state to waiting, moving it to a wait queue, selecting a new process from the ready queue, and loading its context to the CPU

(iii) Allocating Input / Output Resources :- Use a blocking, synchronous system call. This is the safest and simplest approach, as it makes the process wait until the Input / Output operation is complete preventing data inconsistencies.

PART - B

(6.) Context switching Time.

Given: Save state = 2ms, Load state = 3ms, Scheduler overhead = .1ms

(a) Total Context switching Time: 2 + 3 + 1 = 6ms

b) Impact on performance: High context switching time is pure overhead, reducing system throughput and increasing latency

(7) Thread Efficiency

- Every single threaded time = 40sec, 2 threads

Estimated Execution time: 40/2 = 20sec

- Through parallelism on multicore CPUs, improved responsiveness in applications, and efficient resource sharing between threads.

(8) Process scheduling.

- Process and Burst times: P1 = 5, P2 = 3, P3 = 8, P4 = 6

FCFS

Process	P1	P2	P3	P4
Time	0-5	5-8	8-16	16-22
Duration	5ms	3ms	8ms	6ms

- SJF (Shortest Job first)

Process	P2	P1	P4	P3
Time	0-3	3-8	8-14	14-22
Duration	3ms	5ms	6ms	8ms

Round Robin (RR)

Process	P1	P2	P3	P4	P5	P3	P4
Time	0-4	4-7	7-11	11-15	15-16	16-20	20-22
Duration	4ms	3ms	4ms	4ms	1ms	4ms	2ms

(b) Average waiting and Turnaround Times:

Algorithm	Avg Wait time	Avg Turnaround time
FCFS	7.25	12.75
SJF	6.25	11.75
Round Robin	10.75	16.25

Best algorithm :- SJF is the best algorithm for balancing throughput and Turnaround time.

Ans 4. Virtualization and Smart homes.

(i) Cloud Migration

(a) OS Architecture :- A microKernel is best for scalability and security in a virtualised cloud environment due to its modularity and strong isolation of services.

(b) Role of VMs :- Virtual machines aid in Migration by providing isolation for security, easy management and resource optimization through efficient hardware sharing.

(ii) Smart home system

(a) OS Role :- The OS uses priority based preemptive scheduling to ensure high priority tasks are processed immediately, while using inter process communication (IPC) for devices to communicate efficiently.

(b) Suitable algorithm :- A priority Based preemptive algorithm that can be combined with round robin for task of same priority

—— X —— X —— X ——

Assignment no:- 2.

(1) Address Translation Process

The memory management unit (MMU) translates a logical address into a physical address (frame number, offset) by looking up the page number in the process's page table. This process is accelerated by a hardware cache called the TLB (Translation lookaside buffer)

(2) Fragmentation

- Internal fragmentation :- Unused memory within an allocated block. occurs in fixed size allocation schemes.
- External fragmentation :- Total free memory is sufficient, but its not contiguous. occurs in dynamic allocation schemes.
- Mitigation :- The primary technique to eliminate external fragmentation is paging which allows a process to be stored in non contiguous frames.

(3) Paging Based Memory Allocation Trade offs

- Memory Overhead : Requires a page table for each process
- Speed - Address translation requires an extra memory access
- Fragmentation - Solves external fragmentation but can still have minor internal fragmentation in a final page of process

(4) OS and hardware interaction in Virtual Memory.

- Hardware (MMU & TLB): Performs the fast, common case address translation. Traps to the OS on a miss or fau
- Operating system:- Handles page faults by loading pages from the disk into memory, updating page tables

Ans 5. Virtual address and Page table calculation.

Given:- 16 bit virtual address, 1KB Page size.
No. of Virtual pages:- $2^{16}/2^{10} = 2^6 = 64$ Pages
Page table size:- 64 entries * 2 bytes/entry = 128 byt

Ans 6. Memory Allocation Simulation.
for the given processes and initial 1000 KB block, first fit, Best-fit, and Worst fit all result in the same allocation pattern.

Result: Proces P1, P2 and P3 are allocated. P4 (426 KB) cannot be allocated
Total unused memory: 259 KB.

Ans 7. Page Replacement Algorithms.
Given: 3 frames and reference string.
7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2
Page faults summary:
FIFO: 7 page faults.
LRU: 6 page faults.
Optimal: 5 Page faults.
optimal is more efficient.

Ans 8. Dirty Page Overhead.

Given 10ms disk write, 30% of 1000 replaced pages are dirty.

- Optimization: Use a dirty bit in the page table. If the bit is 0, the OS can overwrite it

Ans 9. Autonomous Vehicle Case study.

- Critical Processes (Object detection, Route Planning) These requires real-time responsiveness. the OS should use the working set model to ensure the necessary pages are always in memory, combined with a priority based page replacement policy to prevent them from being swapped out

- Memory Allocation :- A priority based memory allocation strategy is best. Critical tasks get them from a reserved, non paged pool for guaranteed performance, while non critical tasks use the standard demand paged memory pool for efficiency.