Aditya Singh
230101022g

Date / /

Page No.

Assignment -5

1 Necessity of operating system and essentical abstractions

→ modern computer system still heavily on os despite advantages in hardware because os serves two crucial, non-redundant rules:-

- Resources management - It manages and detects & allocates system resources (cpu memory, I/o device) efficienty.

- Abstraction layers:- It provides a ~~sys~~ simple clean and consistent interface to underlying complex, raw hardware making it easier for programme to write application

2 compharison of os Structure:-

→ monolithic:-
All os service are combined into single binary running in kernal space and has fast execution due to minimal over head

Layered:-
Services are organized hierarchically in layer with each layer relying only on the function of the layors below is

microlcernal:-

The lcernal is minimaals only handliny core servrices other service run as users level servers slowest be cause most service call requise context switching and inter process

3 Analysis of Thead & Process Efficiency

-> Process :-

slow requires allocating new virtual address space a new process central block, and copying data structure

Threads :-

fast creation only requires creating a new thread central block.

efficiency:-

Hence, Threads are most efficient.

4 memory Allocation Simulation (first-fit Best-fit, worst-fit):

Process	Requirement
P_1	12 MB
P_2	18 MB
P_3	6 MB

Algorithm	Allocation steps
first-fit	Total unused space · 8 MB + 4 MB + 5 MB = 17 MB P_2 is waiting.
Best fit	Total unused space : 8 MB + 4 MB + 5 MB = 17 Mb P_2 is waiting
worst-fit	Total unused space : 8 MB + 4 MB + 5 MB = 17 MB P_2 is waiting

5 FCFS (first-come-first-served)

Processes are executed in the order of their arrival time.

- P_1 arrives at 0, execetes for 5ms (0-5)
- P_2 arrives at 1, but wait unit P_1 fineshed (5-8)
- P3 arives at 2, wait until P_2 finished (8-16)
- P4 arrives at 3, wait P3 finished for execution 3 ms (16-19).

P_1	P_2	P_3	P_4
0	5	8	16-19

2 Shortest - Job first (SJF)

→ Process are selected boredom the shortest Burst time among these that have arrived

- t = 0, only P1 has arrived,
  P1 starts (BT = 5)

t = 5 ; P1 finishes ; P2 (AT = 3, BT = 5) how all arrived

Selection: P2 and P4 both have (BT = 3)

- choose P2 arrives first

P I executes :-

t = 8 , P2 finishes P3 (BT = 8) and P4 (BT = 3) remain

- selection: P4 has the shortest (BT = 3). P4 executes (B = 11)
- t = 11, P4 executes only P3 (BT = 8), P3 executes
  11 - 19

P1 P2 P4 P3.
0 5 8 11-19

3 Round - Rowd Quantum = 4 ms

Process are given a maximum of 4 ms CPU time, if not finished they are preempted and moved to the baek to the back of the ready quue.

Time (ms)	Execution Process	Remaining BT	Ready queue
0 - 4	P_1 (BT = 5)	1	P_2 (a+1), P_3(a+2), P_4(a+3)
4 - 7	P_2 (BT = 3)	0	P_3, P_4, P_1 (Remain BT=1)
7 - 11	P_3 (BT = 8)	4	P_4, P_1 P_3 (Remain BT=5
11 - 14	P_4 (BT = 3)	0	P_1, P_3
14 - 15	P_1 (BT = 1)	0	P3
15 - 19	P_3 (BT = 4)	0	∅

P_1	P_2	P_3	P_4	P_1	P_3
0	4	7	11	14	15-19

9 Distributed time management

→ (a) Issue in Distributed OS Design for file sharing and resource management

- Global clock Inconsistency -

Distributed system lack a single, sychronized clock. This makes it challenging to establish a consistent order of event which is vertical for file consistency and lock ordering

2 Network latency :- The network introduced delay test for every file access, severy degrade performance A network partion can spliting the system to conflicity update

3 Authentication securing resource is complex issue control list most be consistenty managed all synchronized across multiple independent serve

10 Synchronous checkpoing and Recovary

1 Initiation :- A check point intiates P, begins the process by recarding its local state and sending a checkpoint request message to all other process.

2 co-ordination :- upon recieving the request all processes (Pi) in immediately block their normal execution and stop sending application message

3 local state Recording :- Each blocked process records Its correct local state and the state its communication channels.

(1) IOT Smart Home system

Process scheduling stratey

- Algorithm solution : preemptive
  - priority scheduling with interrupt drive activation
  - Justification ~~R~~ Stratege

1 Priority Assignment :- Assign priority to the security device interruption All other receives low priority

2 Preemption :- use ~~preeption~~ preemption a high - priority security interrupt access while a low schedules will imm respond to power - priority task an

dispatch the critical interrupt handles

3. Justification :- This ensures minimal latency for life safety and security critical task, fulfilling the primary requirement of priority interrupts. The low-priority can tolerate delay but securing alerts can not

Satinder
21/11/25