Kashik Khaskwal
2K17/MC/55

ASSIGNMENT-1

OS

Unit 1 & Unit 2

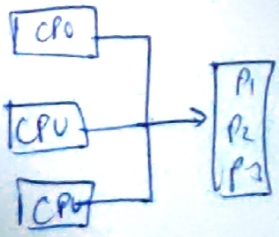
A1

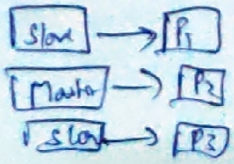
a) **System Calls** are the calls made by the applications or the processes for a particular execution of code block also known as interrupt in computer. You can call the CPU to execute your program with high priority & execute other commands later.

System Programs are the programs that are used & required to run the system - machine, I/O devices & other connected peripherals. They are also known as system software.

b)

Symmetric	Asymmetric
- Each processor run the task in OS	- Only master processor run task in OS.
- Processor take processes from Ready Queue	- Master processor assign processes to slave processors.
- All processor have same architecture	- May have same or diff architecture.





c) Multiprogramming

It is the ability of OS to execute more than one program on a single processor machine. More than one job can reside in MM at any given time. eg Firefox & Excell running simultaneously

Multitasking

It is the ability of os to execute more than one task simultaneously on single processor machine. Actually CPU switches from one task to another so quickly that appears they are being executed at same time.

d) Hard real time systems guarantees the critical task complete on time. This goal requires that all delays in the system to be bounded from the retrieval of stored data to the time that it takes the os to finish any req made of it

Soft real time systems where a critical real time task gets priority over other task & retains the priority until it completes.

e)

Microkernel	Monolithic Kernel
- User Service & Kernel service are kept in separate address space.	- Both are kept in same address space
- Smaller in size	- Comparatively larger in size
- Slow Execution	- Fast Execution
- Easily Extendible	- More to extend

Q2 A2

A distributed system is a system in which components are located on diff networked comp, which can communicate by passing messages. They interact with one another to achieve a common goal. Features & Challenges are:-

- **Resource Sharing**
  Existing resources can be accessed ~~acc~~ multiple comp in system. Data is shared for consistency & Exchange of information.

- **Heterogeneity**
  In distributed system, components can have variety & difference in network, Comp hardware, OS, etc.

- **Openness**
  It is concerned with extension & improvement of Distributed systems. Must be open in terms of Hardware & Software.

- **Concurrency**
  It property of system representing the fact that multiple activities are executed at the same time. It reduces the latency & increases throughput.

- **Scalability**
  Concerned about how Distributed system handles growth as no. of users increases

- **Fault tolerance**
  Software, network, anything can fail. System must be designed to tolerate this.

- **Transparency**
  Should be perceived by users as a whole rather than a collection of cooperating components

A3

(Privileged) Kernel mode

- Set value of timer
- Clear memory
- Turn off interrupts
- Modify entries in device status table
- Access I/O devices

User mode

- Read Clock
- Issue a trap instruction
- Switch from User to Kernel

A4

Features

- Have room on the board for four or more processors
- Efficient in handling high speed mathematical computation
- Can handle large database of business transactions

Requirements

- Must not be reliant on previous outcome
- Does not need to be executed in particular order
- Do not return anything that would need to be accessed later in the code.
- SMP Env require that each CPU in system have access to same physical memory using same system Bus
- Custom hardware implementation as typically on circuit boards

A5 By establishing a set of priviliged instructions that can be executed only when in the monitor mode, the OS is assured of controlling the entire system at all times.

A6 An Interrupt is a hardware generated change of flow within the system. A trap is software generated interrupt. Yes, they can & can be used to call OS routines or to catch arithmetic errors.

A7

a) Install faster CPU – No

b) Increase # processes – No

c) Decrease # processors – Yes

d) Install more Main mem – Yes, as more pages can remain resident & not require paging to or from the disk.

e) Increase the Page Size – ↑ page size will result in fewer page faults if data is being accessed sequentially. If data access is more or less random, more paging action could ensue because fewer pages can be kept in memory & more data is transferred per page fault. So its as likely to decrease utilization as its to increase it.

A8

The main advantage is it allows good main... where we can make changes without affe... layers interfaces.

A9

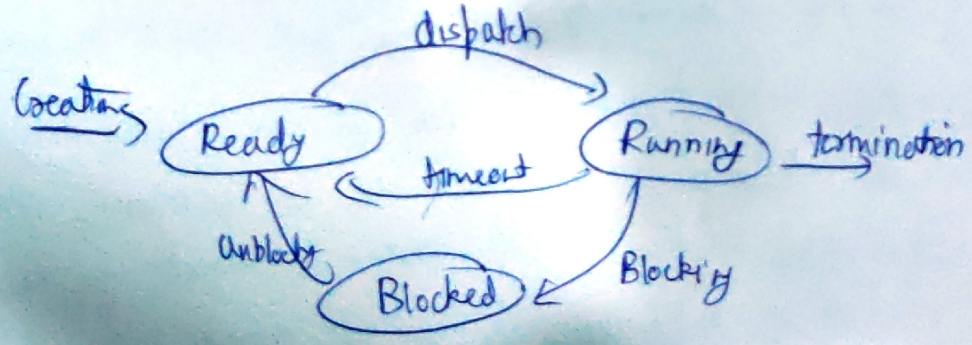
- A web server that services each request in a seperate thread.
- A parallelized application such as Matrix multiplication where diff parts of matrix can be looked on in pa...

A10

n! schedules are possible.

($n! = n*(n-1)*(n-2)*\ldots*3*2*1$)

A11 A process is an instance of program execution. The lifecycle of process can be described by a state diagram which has the states representing the execution status of the process at various times & transitions that represent changes in execution status.



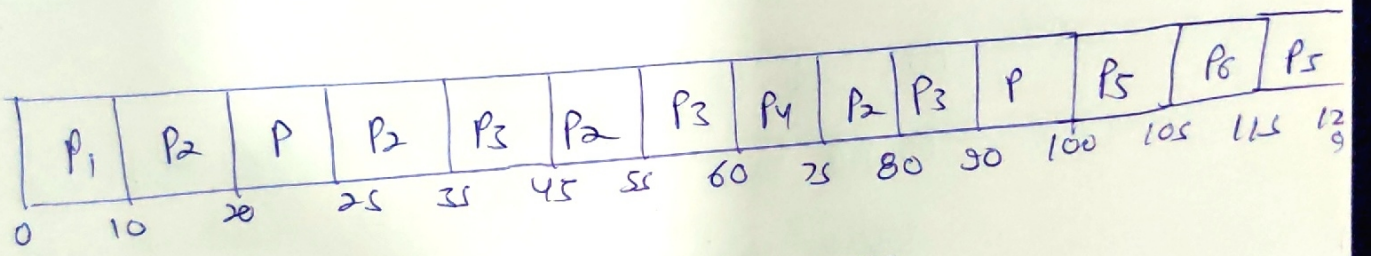
A12

A process which has finished the execution but still has entry in process table to report to its parent process is known as zombie process. General child processes become Zombie.

To Kill :-

- Sending SIGCHLD signal to parent of zombie so that parent remove it.
- Killing the parent process will also kill zombies.

A13, 14



	TT = C - AT	WT = TT - BT
P_1	20	0
P_2	55	30
P_3	60	35
P_4	15	0
P_5	20	10
P_6	10	0

$$\text{CPU Utilization rate} = \frac{120 - \text{Idle time}}{120} \times 100$$

$$= \frac{105}{120} \times 100 = 87.5\%$$

A15

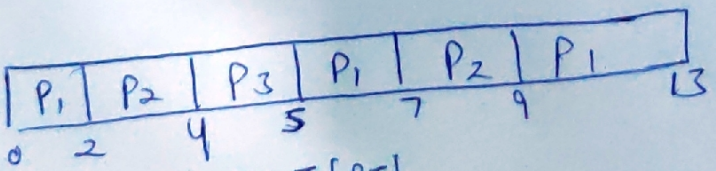
a) CPU Utilization is increased if number of overhead associated with context switching is minimized. The context switching overheads can be lowered by performing context switch infrequently. This could however increase the response time for process.

b) Average TT is minimized by executing the shortest task first, such a scheduling could however starve long running task & thereby increases their waiting time.

c) CPU Utilization is maximized by running long CPU bound task w/o performing context switch. I/O device utilization is maximized by switching I/O jobs as soon as they become ready to run, thereby increasing the overheads of context switches.

A16

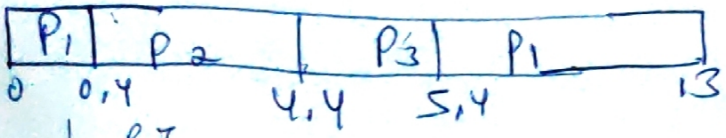
Gantt Chart for RR



	TT		WT	RT
P1	13 − 0	= 13	5	0
P2	9 − 0.4	= 8.6	4.6	2
P3	5 − 1	= 4	3	1

Avg TT = 8.534
Avg WT = 4.2
Avg RT = 1.534

Gantt Chart for Priority Scheduling



	TT	RT
P_1	13	0
P_2	4.0	0
P_3	4.4	3.4

Avg TT = 7.134

Avg RT = 1.134

Process switching is context switching from one process to another. It involves switching out all the process abstractions & resources in favours of those belongs to a new process. Most notable this means switching memory address space

Thread switching is context switching from one thread to another in same process. Its much cheaper & involves switching out the abstractions unique to threads.

⇒ Thread switching is much faster than process switching

A18

Assuming $a = 0$ & $\&a = 100$

Output

5 100 (child) or -5 100 (parent)
-5 100 (parent) 5 100 (child)

A19

a) False as Context_Switch time is less in user threads.

b) True as if user thread is blocked whole system is blocked.

c) True

d) True

∴ b, c, d are True.

A20

a) Benefit of Synchronous com is that it allows a rendezvous b/w sender & receiver. A disadv of Blocking Send is that rendezvous may not be req & the message could be delivered asynchronously.

b) Automatic Buffering provides a queue with indefinite length thus ensuring send do will never have to block while waiting to copy a message. Memory could be wasted. Explicit buffering specifies how large the buffer is. In this situation senders may be blocked while waiting for the available space in Queue. Less likely for memory to be wasted.

c) Send by Copy & Send by reference : Send by Copy doesn't allow to alter the state of parameter, send by reference does allow it. Benefit of send by ref is it allows programmer to write a distributed version of centralized application.

d) Fixed size & Variable size messages :-

Implications of this are mostly related to buffering issues with fixed size messages, buffer with fixed size can hold a known no. of messages. No. of variable sized messages that can be held by such a buffer are unknown.

A21 Three modes :-

① One to one

- One to One node relationship b/w Kernel & User thread.
- Multiple threads can run on single processor.
- Limit no of models because one User thread per one kernel thread

② Many to Many

- Multiple User threads multiplex to same or lesser number of kernel threads.
- No. of Kernel threads are machine specific.
- If User thread gets blocked we can schedule other user thread to other Kernel thread. i.e System doesn't block.

③ Many to One

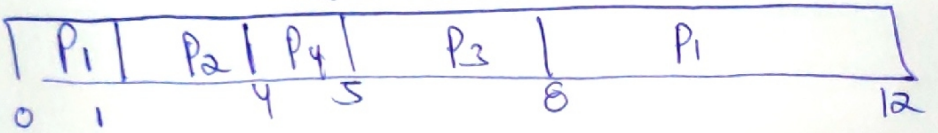
- Multiple User threads mapped to one Kernel thread
- When user thread makes blocking call entire process blocks
- Not able to access multiprocessor at the same time.

A22

Question Incomplete

A23

Gantt Chart



	TT = CT - AT	
P_1	12 - 0	= 12
P_2	4 - 1	= 3
P_3	5 - 4	= 1
P_4	8 - 2	= 6

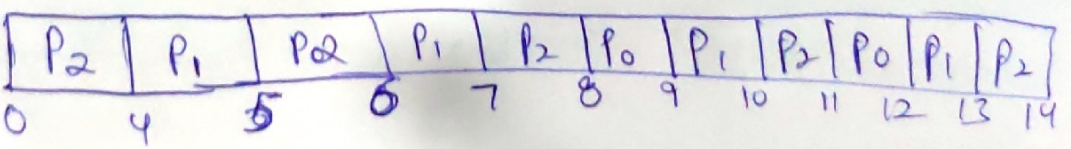
$\therefore$ Avg Turnasoundtime $= \frac{(12+3+6+1)}{4}$

$= 5.5$

A24

	BT	AT
P_0	2	0
P_1	4	0
P_2	8	0

Gantt Chart



	TT
P_0	12
P_1	13
P_2	14

$\therefore$ Avg Turnaround time = 13 units

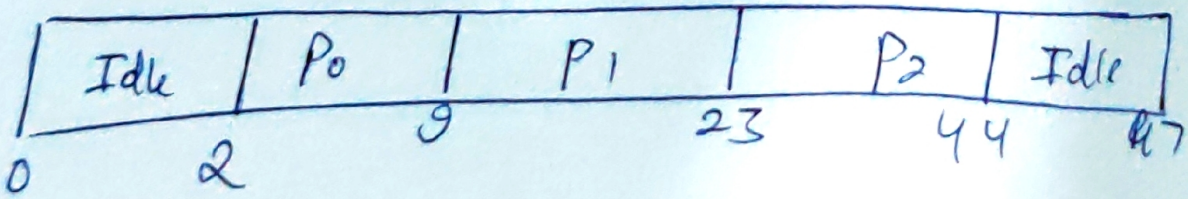
A25

	AT	ET
P_0	0	10
P_1	0	20
P_2	0	30

ET DIVISION

	I/O	CPU	I/O
P_0	2	7	1
P_1	4	14	2
P_2	6	21	3

Gantt Chart



Total time spent = 47 units

Idle time = 2 + 47 − 44 = 5 units

% of Idle time = $\frac{5}{47} \times 100 = 10.6\%$