Thursday, March 24, 2022 2:15 PM

PC , Response time , Starvation

Preemption for SJF = process with Lower Tcpu can preempt process with higher Tcpu
Adv and Disadv SJF

Interrupt Vector Table !!!!!!

Context Switching =

Context = it is the current state of the CPU registers for a given process

OR

Snapshot of CPU registers at a given point of time/ given instance of time

Process P1 is running

CPU has some context of P1

Some interrupt occurs

Process returns to ready queue

Context of P1 is saved in the PCB

&

Context of ISR is loaded in the CPU

This is called as Context Switching !!! **Saving** the context of outgoing process and **loading** the context of incoming process is context switching .

The switching happens every time a process leaves CPU and another process resumes or starts on the CPU.

Context Switching enables the process to RESUME from the point it had left

---

Process scheduling algorithms --------

FCFS (FIFO) , Non Preemptive SJF, Preemptive SJF

3. Priority scheduling -----

Which process is selected for using CPU? The process with highest priority amongst the ready queue processes is selected.

For How long it gets the CPU ? Ideally till process completes

Practically --- till IO instruction occurs, till interrupt occurs, preemption occurs

How does a process get priority ?

Basis of priority

1. Kernel process may have higher default priority than user process
2. In multiuser system one user process may have higher priority than process created by another user
3. A process starts running and it can make system calls to set specific priority for itself or for the processes it is creating.
4. Auto priority promotion --- the priority of a process in increased by the kernel after some time

Priority ranges can change from kernel to kernel !
1 to 10 priority in Solaris , Unix based OS
1 to 25
Some priority is Highest = 10
Some priority is Lowest = 1
Some normal priority = 5

Adv - Priority is considered
Disadv - Starvation of low priority process
Some kernels wait for a threshold time for the process to get a chance to execute
If the process is not getting a chance then the kernel automatically promotes the process to higher priority

Assume the range 1 to 10 , 1 is lowest 10 is highest !!!!

Process	Arrival time	Tcpu	Priority
P1	0	4	8
P2	0	6	3
P3	2	4	5

Use Non Preemptive Priority to calculate avg Ta and avg Wt

Gantt chart/timeline

Ready queue at time 0	P1,P2	Select P1
Ready queue at time 2	P2 , P3	P3 will be selected at time 4
Ready queue at time 8	P2	P2 will be selected

P1	P3	P2
0-----4	4 ------8	8------14

Process	Wt = start - arrival	Ta = Wt + Tcpu
P1	0-0=0	0+4 = 4
P2	8-0=8	8+6 = 14
P3	4-2=2	2+ 4=6
Avg	(0+8+2)/3=3.3333	(4+14+6)/3=8

1to 10 1 is lowest

Process	Arrival	Tcpu	Priority
P1	2	4	7
P2	1	5	5
P3	5	3	8

Use Preemptive **priority** to calculate avg Wt and avg Ta

RQ at time 0	Empty

RQ at time 1	P2(5)
RQ at time 2	P2(4)
RQ at time 5	P2(4) , P1(1)
RQ at time 8	P2(4)
RQ at time 9	-----

P2	P1 preempts P2	P3 preempts P1	P1 resumes	P2 resumes
1---2	2-------5	5----------8	8---9	9----13

Process	Wt = Start -arrival Resume -preempt	Ta Wt+ Tcpu
P1	2-2 = 0 8-5=3 3+0=3	3 +4=7
P2	1-1=0 9-2=7 7+0=7	7+5=12
P3	5-5=0	0+3=3
Avg	(3+7+0)/3=3.3333	(7+12+3)/3 = 7.3333

HW ---practice question

Priority from 1 to 10 1is lowest

Process	Arrival	Tcpu	Priority
P1	0	5	5
P2	3	3	8
P3	4	2	9
P4	1	6	3

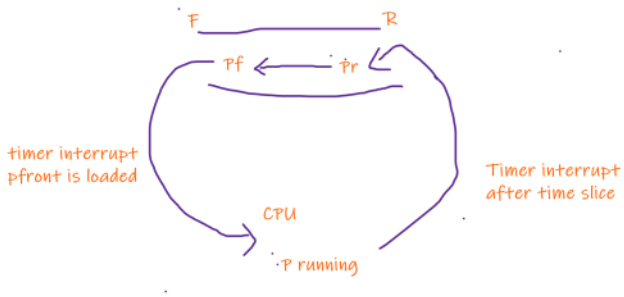
Calculate avg Wt and avg Ta using PREEMPTIVE PRIORITY SCHEDULING

Round Robin Scheduling

1. This is same as FIFO in terms of selecting which process = Process at the front of the ready queue is selected
   For how long does the process ideally get the CPU ? = **for a stipulated time quantum** (for a given time slice)

   Timer Interrupt Occurs after every TIME SLICE
   The ISR switches the context and loads the next process in the queue

   Kernel decides the time slice



DISADVANTAGE ----------1. Wt per process increases, Ta increases , Throughput decreases

2. Too much of context switching overhead

ADVANTAGES ---------- 1. End User gets a effect of MULTITASKING

Multitasking = **feeling** of running many programs simultaneously

Context switching is frequent and time slice is very small !!!

Fast switching is beyond human perception

1. Response time IMPROVES.
2. Starvation does not occur

Process	Arrival	Tcpu
P1	0	6
P2	1	4
P3	2	3

Assuming time slice as 2ms , use round robin to find avg Wt and avg Ta

Gant chart

RQ @ 0	P1	P1 is selected
RQ @ 1	P2	
RQ @ 2	R --P1(4)--P3----- F	
RQ@ 4	R---P2(2)---P1(4)--	
RQ @ 6	R--P3(1) --P2(2)	
RQ @ 8	R---P1(2) ---P3(1)	
RQ @10	R --- P1(2)	
RQ @ 11	R----------F	

P1	P2	P3	P1	P2	P3	P1	
0---2	2 ----4	4...6	6----8	8---10	10---11	11---13	

Process	Wt	Ta
P1	0-0=0 6-2=4 11-8=3 (0+4+3)=7	7+6 = 13
P2	2-1=1 8-4=4 1+4=5	5+4=9
P3	4-2=2 10-6=4 2+4=6	6+3=9
Avg	(7+5+6)/3=6	(13+9+9)/3=10.3333
