Thursday, October 21, 2021 11:03 AM

From we can access windows folders
WSL ---- the windows folders **mounted** on /mnt

How many topmost folders are there in windows ?
Linux has only one topmost folder!!

Process Scheduling ----------

Wt and Ta of a process !!!

Disadvantage of FIFO ---------- !!!

SJF = Shortest Job First
Select a process that is having **lowest Tcpu** from all the processes in the ready queue
The cpu will be allocated till the process completes!!!

Advantage : if a process has low Tcpu then it will get the chance first- average Wt will come down
Average Ta will come down , throughput will increase!!

Disadvantage : -- a large Tcpu process may wait indefinitely , if more and more small Tcpu processes keep coming
STARVATION

--the kernel cannot PREDICT the Tcpu of a process before the process is executed!!! CANNOT be practically implemented.

THE Tcpu cannot be predicted
Scanf %d &input

For(i=0;i<input;i++)
x=x+I } for 1 execution it takes 2ms

-----Response time is Poor

Process	Arrival time	Tcpu
P1	0	4
P2	0	3
P3	1	5
P4	2	2

Calculate Average Wt and Ta using **SJF** (non preemptive)

P2 0--------3	P4 3------5	P1 5-----9	P3 9------14
Queue == P1,P2	Queue == P1,P3,P4	Queue = P1,P3	

Process	Wt = Starttime- arrival time	Ta = Wt + Tcpu
P1	5-0=5	5+4=9
P2	0-0=0	0+3=3
P3	9-1=8	8+5=13
P4	3-2=1	1+2=3
Average	5+0+8+1=14 14/4=3.5	9+3+13+3=28 28/4=7

To Preempt = To replace lower priority process forcefully by higher priority process !!!! = Preemption

Preemptive = the kernel that supports preemption is a PREEMPTIVE Kernel

Non Preemptive = the kernel that does not support preemption is a NON PREEMPTIVE Kernel

Process	Arrival time	Tcpu
P1	0	9
P2	2	2
P3	3	3

Calculate Avg Wt and Avg Ta using Preemptive SJF

P1	P2 **preempts** P1	P3	P1 resumes	
0---2	2---------4	4------7	7----14	
Queue = P1	Queue= P1,P3	Queue P1		

Process	Wt	Ta
P1	Start-arrival 0-0=0 Resume-preempt 7-2=5	5+9=14
P2	2-2=0	0+2=2
P3	4-3=1	1+3 = 4
Average	5+0+1=6 6/3=2	14+2+4=20 20/3=6.6667

HW

Process	Arrival time	Tcpu
P1	0	6
P2	2	2
P3	5	1
P4	1	7

Calculate avg Wt, Ta using NonPreemptive and Preemptive SJF!!

Priority Scheduling ---------
Processes get explicit priority = ranging from 1 to 10 for example
Select a process that has highest priority amongst all processes in the READY Q
How long does it get the CPU ?

Non preemptive --till process completes
Preemptive --- till it is not preempted or till it completes

Priority is assigned depending on

1. Kernel process , user process
2. By default all user programs get normal priority = 5
3. Multi User OS --- depending on which user starts the process ---the priorities may change
4. From a program we can reset the priority to different priority

Preemptive Priority = if a process with higher prio arrives then running process is preemptive!!!
Non Preemeptive Priority scheduling ---- if a process with higher prio arrives it waits till running process is completed

Advantage --- priority of a process is considered
Disadvantage --- STARVATION of lower priority process
--- Response time is poor

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

Can u find Wt and Ta Avg using preemptive priority scheduling

Rule ---select highest priority

Subrule --- if 2 processes have same priority then front of the queue is executed first
If 2 processes have same arrival time
Randomly select

Queue = F --P1 P2-------R	P3 preempts P2, Queue = F--P1,P4, P2		Queue--P1,P2	P1
P2 0----------3	P3 3--------7	P4 7----9	P2 9-----12	P1 12---17

Process	Wt	Ta
P1	12-0=12	
P2	0-0=0 9-3=6	
P3	3-3=0	
P4	7-1=6	
Avg		

Can u find Wt and Ta Avg using Non preemptive priority scheduling

Queue = P1,P2	Queue = P1,P4,P3	Queue= P1,P4	Queue= P1	
P2 0--------6	P3 6----------10	P4 10----12	P1 12-----17	

Process	Wt	Ta
P1	12-0=12	12+5=17
P2	0-0=0	0+6=6
P3	6-3=3	3+4=7
P4	10-1=9	9+2=11
Avg	12+0+3+9=24 24/4=6	17+6+7+11=41 41/4=10.25

Round Robin Scheduling -------

This is same as FIFO scheduling
CPU allocation is done for a particular **time slice = time quantum**
Select the process in the **front** of the queue
Allow the process to run for the allocated duration
After that a timer interrupt occurs
Process returns to the **rear** of the ready queue

Advantages : every process gets a chance to go ahead without **waiting for other process to complete**

---- IMPROVE the response time
-----Giving an effect of MULTITASKING (user gets a feel that many applications are running simultaneously)
: No Starvation

Disadvantage : **Switching** processes takes extra time
Wait time increases, Ta increases , Throughput is LOW

Time slice = time quantum = 2ms

Process	Arrival time	Tcpu
P1	0	4
P2	1	5
P3	2	3

Queue=F ---P1,P2,P3	Queue=F---P2,P3,P1	P3,P1,P2	P1,P2,P3	P2,P3	P3,P2	P2			
P1 0-------2	P2 2---4	P3 4----6	P1 6---8	P2 8----10	P3 10---11	P2 11-12			

Process	Wt	Ta = Wt + Tcpu
P1	0-0=0 6-2=4	4+4=8
P2	2-1=1 8-4=4 11-10=1 4+1+1=6	6+5=11
P3	4-2=2 10-6=4 4+2=6	6+3=9

Using **RR** , TS = 3ms , find avg Wt, ta

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