Wednesday, March 23, 2022 10:43 AM

PCB = info about process -pid, state, loaction
Ready queue = queue in which process waits for CPU
IO-Wait-State = Process waits for IO by DMA
Process address space = process space = code, data, stack,heap/extra
Mirco kernel = windows
Kernel space and User space
Kernel mode and User mode
Wait time
Throughput
ISR
IVT
Ex Maskable interrupt = RST interrupts , INTR
Non Maskable interrupt = TRAP

RTOS = LinuxRT , ADA ,
Real Time OS = the processes with highest priority are those which has earliest deadline !!

CPU Schedular is the part of the kernel that will schedule a process.

Schedule a process = it will select process from ready and allocate CPU to it for a certain duration

At this time it must consider the factors like Wt, Ta, Throughput ,response time !!!

Basic scheduling techniques ---

1. FCFS = FIFO
    - Which process is selected from the ready Queue = the frontmost process
    - How long will process get the CPU ? = ideally until termination
        - Practically = till the IO instruction occurs
        - OR till interrupt occurs

    Adv : simple , all processes will get a change to run sooner or later
    Disadv : no priority , throughput of system may reduce
    Ta of low Tcpu process may be high as it is waiting after the high Tcpu process time

    Operating System - Silberschatz , galvin

Process	Arrival Time	Tcpu
P1	2	5ms
P2	0	3ms
P3	1	6ms

Calculate Avg Ta and Avg Wt using FIFO scheduling algo ---

GANTT CHART / TIMELINE

P2	P3	P1			

0--------3	3---------9	9----14			

Process	Wt (Start - arrival)	Ta (Wt + Tcpu)
P1	9-2 = 7	7 + 5 = 12
P2	0-0 = 0	0+3 = 3
P3	3 - 1=2	2 + 6=8

Avg Wt = (7+0+2)/3 = 3
Avg Ta = (12+3+8)/3 = 7.6667

Ex2---

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

P1	P2	P3
0----3	3---6	6---10

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

2 . SJF = Shortest job first

Which process is selected from the ready Queue = processes with lowest Tcpu is selected
How long will process get the CPU ? = ideally until termination
Practically = till the IO instruction occurs
OR till interrupt occurs

Adv - Throughput improves, as Ta reduces as small Tcpu process waits for less time
Disadv - 1. Practical implementation is not possible (theoretical)
Scanf("%d",&val)
for(i=0;i<val;i++) }}}} Kernel does not know val ? -- kernel cannot say how many times loop will run --can it predict Tcpu before the code runs ?
...
2. STARVATION = a process with high Tcpu MAY never get a chance to run WHY ? As lower Tcpu processes keep coming in the queue

Process	Arrival time	Tcpu

P1	0	6
P2	2	3
P3	0	4

Calculate avg Wt and avg Ta using SJF scheduling algo .

Ready Queue R<-------------P1--->F

Gantt chart/timeline

P3	P2	P1
0-----4	4----7	7----13

Process	Wt (start-arrival)	Ta (Wt +Tcpu)
P1	7-0=7	7+6 = 13
P2	4-2=2	2 + 3=5
P2	0-0=0	0+4 = 4
Avg	(7+2+0)/3 = 3	(13+5+4)/3 = 7.3333

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

Calculate avg Wt and avg Ta using Non Preemptive SJF

Ready Queue	R<----------------------------P3-P2-P1-->F	P1 is selected
	R<----------------------------P3-P2--->F	P3 us selected
	R<----------------------------P2-->F	P2 is selected
	R<---------------------P4--------->F	

P1	P3	P2	P4
0---1	1---3	3-------8	8-----9

Process	Wt	Ta
P1	0-0	0+1=1
P2	3-0=3	3+5=8
P3	1-0=1	1+2=3
P4	8-4=4	4+1=5
Avg	(0+3+1+4)/4=2	(1+8+3+5)/4 = 4.25

Preemptive and Non Preemptive SJF

To preempt = Apply force to get the resource
Who can preempt ? Process with higher priority can preempt a process with lower priority

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

Calculate avg Wt and avg Ta using **Preemptive** SJF

Ready Queue	R<-----------------------------P3-P2-P1-->F	P1 is selected to use the CPU
	R<-----------------------------P3-P2--->F	P3 us selected
	R<-----------------------------P2-->F	P2 is selected
	R<----------------------P4--------->F	**P4 preempts P2**
	R<----------------------P2(4ms remains)--------->F	

P2 remaining Tcpu(4) Incoming P4 Tcpu (1)

P1	P3	P2	P4 preempts P2	P2 resumes
0---1	1---3	3--4	4---5	5-------9

Process	Wt (start -arrival) (resume -preempt)	Ta
P1	0-0	0+1=1
P2	3-0=3 5-4=1 (3+1)=4	4+5=9
P3	1-0=1	1+2=3
P4	4-4=0	0+1=1
Avg	(0+4+1+0)/4=1.25	(1+9+3+1)/4 = 3.5

Windows OS ---- WSL = Windows Subsystem for Linux = it allows us to use Linux OS through windows !!!
We do not have to install Linux separately for practice

Linux distribution -----------

Linux based OS ---------------- Flavors of Linux

1. Red Hat Linux
2. **Ubuntu**
3. Fedora Core
4. Suse
5. Mint
6. Mandriva
7. BOSS
8. Debian
9. Kali
10. …
11. ….
12. …
13. …
14. …
15. ….
16. …

Once the WSL is set up !!!!
Go to start ----Ubuntu ------------->open = The terminal is opened !!!
You see a prompt !!! ============= THIS IS THE Ubuntu CLI (interface between System and end-user)

EndUser ---------enters commands <------------->System

Ex1	Once you see the **prompt** , enter many times	See the prompt is repeated on every line
Ex2	Type a command clear	See that the prompts and cleared and we are at the top
Ex3	Type a command ls	This will show the contents of the current folder(directory)
Ex4	Type a command pwd	It will show the **P**resent **W**orking **D**irectory = current folder

/home/unix = What does it mean ?

/ = topmost folder
------ sub folder ---- home
-----sub folder -----unix

Windows topmost folder = many top folders = C: , D: , E: etc
But Linux has ONLY one topmost folder = / (this folder is denoted by forward slash)

Ex5	Go to the topmost folder	Type command cd space forward slash	cd /	Cd = change directory (folder)
Ex6	Type pwd and observe current folder name			
Ex7	Go to the earlier folder	cd /home/unix		
Ex8	Again type pwd and observe output			

____

____