Nome:	WESLEY MARCOS BORGES					Curso:	GEC
Matrícula:	1651	Período:	P8	Matéria:	C012	Turma:	

Cap. 5 – Scheduling de CPU

1) O ***Scheduler*** de curto prazo seleciona o processo que está na Ready Queue para entrar em execução na CPU, para que não fique parada. O ***Dispatcher*** funciona como um mensageiro, pois é responsável por levar esse processo da Ready Queue para a CPU.

2) O balanceamento de cargas é utilizado pelo SO para distribuir uniformemente as cargas de processamento entre todos os processadores. As políticas usadas podem ser:

- **Migração por Absorção =** um processador que está ocioso "puxa" um processo de outro processador que está ocupado para si, equilibrando as cargas entre eles.
- **Migração por Expulsão =** é feita uma verificação constante por uma tarefa em cada um dos processadores. Quando houver um desequilíbrio, ela tenta expulsa alguns processos para equilibrar o sistema.

3) Existem diversos critérios utilizados para fazer a avaliação dos algoritmos de scheduling. Os três que julgo os mais importantes são:

- **Utilização da CPU =** avalia a quantidade de tempo em que a CPU ficará ocupada, o qual não poderá executar novos processos.
- **Time waiting =** tempo total em que um processo fica na Ready Queue, ou seja, o quanto ele espera até ser executado.
- **Throughput =** quantidade de processos completos por unidade de tempo.

4) Os principais algoritmos de scheduling utilizados são:

- **First Come First Served (FCFS) =** o primeiro processo que chega na CPU é o primeiro a usá-la, até que termine o processo.
- **Shortest Job First (SJF) =** o processo de menor custo tem prioridade na execução.
- **Priority Scheduling (PS) =** o processo de maior prioridade, denominada Burst, tem uma maior prioridade.
- **Round Robin (RR) =** é muito semelhante ao FCFS, todavia ele é um algoritmo preemptivo.

5) Os processos ***CPU Bound*** são processos que passam a maior parte do tempo utilizando o processador, ou seja, tem altos picos de CPU. Já os processos ***I/O Bound***, os processos passam a maior parte do tempo em espera de alguma I/O, ou seja, tem baixos picos de CPU.

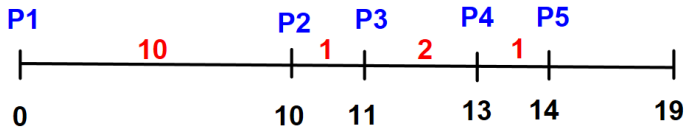
6)

Process	Burst Time	Priority
P_1	10	3
P_2	1	1
P_3	2	3
P_4	1	4
P_5	5	2

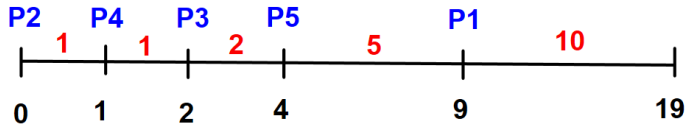
Ordem de chegada: P1, P2, P3, P4, P5 (todos no tempo 0)

a.

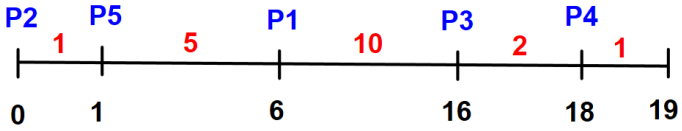
FCFS



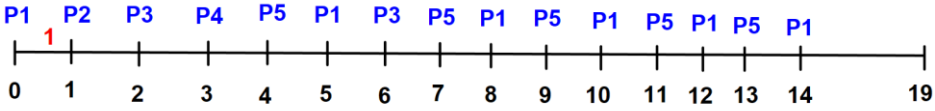
SJF



PS (NP)



RR



P1 => ~~10 9 8 7 6 5 4 3 2 1~~ 0
P2 => ~~1~~ 0
P3 => ~~2 1~~ 0
P4 => ~~1~~ 0
P5 => ~~5 4 3 2 1~~ 0

b.

FCFS

$$T_m = \frac{0 + 10 + 11 + 13 + 14}{5} = \frac{48}{5} = 9,6 \text{ ms}$$

SJF

$$T_m = \frac{9 + 0 + 2 + 1 + 4}{5} = \frac{16}{5} = 3,2 \text{ ms}$$

PS (NP)

$$T_m = \frac{6 + 0 + 16 + 18 + 1}{5} = \frac{41}{5} = 8,2 \text{ ms}$$

RR

$TEP_1 = 14 - 5 - 0 = 9\ ms$
$TEP_2 = 1 - 0 - 0 = 1\ ms$
$TEP_3 = 6 - 1 - 0 = 5\ ms$
$TEP_4 = 3 - 0 - 0 = 3\ ms$
$TEP_5 = 13 - 5 - 0 = 8\ ms$

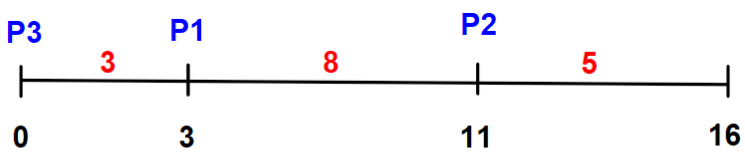
$$T_m = \frac{9 + 1 + 5 + 3 + 8}{5} = \frac{26}{5} = 5,2 \text{ ms}$$

c. O menor tempo de espera foi o SJF (Shortest Job First) com **3,2 ms.**

7)

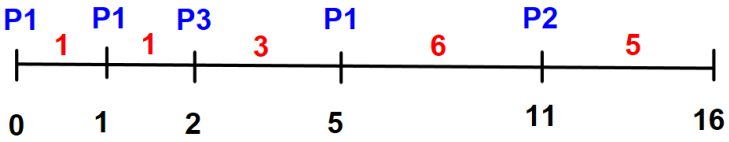
Process	Arrival Time	Priority	Burst Time
P1	0	2	8
P2	1	5	5
P3	2	1	3

a.



$$T_m = \frac{3 + 11 + 0}{3} = \frac{14}{3} = 4,67 \text{ ms}$$

b.



$TEP_1 = 5 - 2 - 0 = 3\ ms$
$TEP_2 = 11 - 0 - 1 = 10\ ms$
$TEP_3 = 2 - 0 - 2 = 0\ ms$

$$T_m = \frac{3 + 10 + 0}{3} = \frac{13}{3} = 4,33\ \text{ms}$$

8) A ***afinidade*** com o processador acontece quando um processo, ao já ter sido executado por um processador, escolhe ser executado novamente por esse mesmo processador para aproveitar os dados armazenados na memória cache, a fim de agilizar o processamento.

9) ***Memory Stall (Queda de memória)*** é o tempo de espera do processador ao acessar os dados da memória. Para evitar que isso aconteça, atribui-se duas threads de hardware para cada núcleo, ou seja, se o processo for interrompido, o núcleo usará a outra Thread. Esse processo é conhecido como **Hyper-Threading.**