

Nome: Pedro Gabriel Garcia Ribeiro Balestra		Matrícula: 1551
Curso: GEC	Período: P8	Matéria: C012

Cap.5 – Scheduling de Cpu

1. Scheduler de Curto Prazo são algoritmos que realizam o processo de selecionar os processos presentes na Fila de Prontos para ser executado, podendo ser algoritmos preemptivos (Podem ser frustados pelo SO) e não-preemptivos (não podem ser frustados pelo SO). E o despachante é o modulo que passa o controle da Cpu para o processo selecionado pelo Scheduler de Curto Prazo, sendo chamado a cada mudança de contexto
2. Pode ser mantido utilizando a abordagem de **migração por expulsão**, verificando constantemente a carga de cada processador, e a abordagem de **migração por absorção**, que é quando um processador ocioso extrai uma tarefa que está esperando em um processador ocupado
3. Podemos comparar algoritmos por 3 maneiras: a **utilização da CPU**, indicando o quanto a CPU permanece ocupada, **throughput**, quantidade de processos completados por unidade de tempo, **tempo de turnaround**, intervalo de tempo entre a submissão e término do processo, dentre outros.
4. Existem vários algoritmos de Scheduling de Processos, dentre eles temos
 - **FIFO/FCFS** – Onde o primeiro processo a entrar na fila será “servido”, algoritmos Não-Preemptivo.
 - **SJF** – O Processo que tem o pico de CPU mais curto será “servido”, pode oferecer o menor tempo médio de espera, algoritmo Não-Preemptivo e Preemptivo.
 - **PS** – Uma prioridade é associada a cada processo, executando do que tem maior prioridade ao que tem menos, algoritmo Não-Preemptivo e Preemptivo.
 - **RR** – É dado um Quantum de Tempo ao algoritmo, onde a cada Quantum de Tempo o algoritmo verifica qual processo irá executar, semelhante ao FIFO, porém com Preempção.
5. **CPU-Bound:** Processos com longos picos de CPU
IO-Bound: Processos com curtos picos de CPU

6. A)

<u>Process</u>	<u>Burst Time</u>	<u>Priority</u>
P_1	10	3
P_2	1	1
P_3	2	3
P_4	1	4
P_5	5	2

- FCFS:



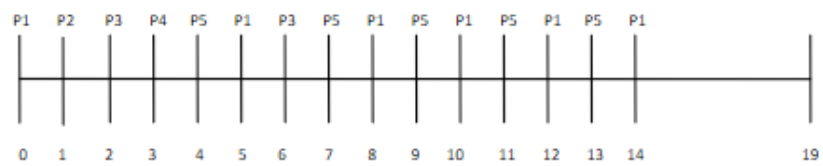
- SJF(NP)



- PS(NP)



- RR(Q=1)



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)

TM FCFS

$$TM = \frac{0 + 10 + 11 + 13 + 14}{5} = 9.6 \text{ ms}$$

TM SJF

$$TM = \frac{0 + 1 + 2 + 4 + 9}{5} = 3.2 \text{ ms}$$

TM PS

$$TM = \frac{0 + 1 + 6 + 16 + 18}{5} = 8.2 \text{ ms}$$

TM RR

$$TM = \frac{9 + 1 + 5 + 3 + 8}{5} = 5.2 \text{ ms}$$

$$Tep1 = 14 - 5 - 0 = 9 \text{ ms}$$

$$Tep2 = 1 - 0 - 0 = 1 \text{ ms}$$

$$Tep3 = 6 - 1 - 0 = 5 \text{ ms}$$

$$Tep4 = 3 - 0 - 0 = 3 \text{ ms}$$

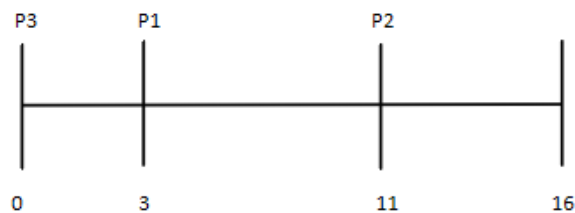
$$Tep5 = 13 - 5 - 0 = 8 \text{ ms}$$

C) O algoritmo SJF resulta no menor tempo de espera com **3.2 ms**

7.

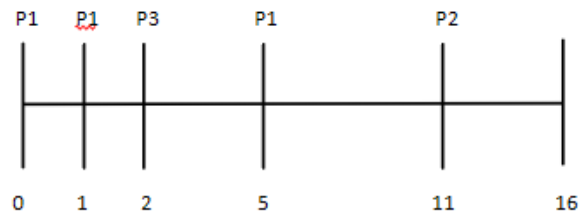
<u>Process</u>	<u>Arrival Time</u>	<u>Priority</u>	<u>Burst Time</u>
P1	0	2	8
P2	1	5	5
P3	2	1	3

A) PS(NP)



$$TM = \frac{0 + 3 + 11}{3} = 4.67 \text{ ms}$$

B) PS(P)



$$Tep1 = 5 - 2 - 0 = 3 \text{ ms}$$

$$Tep2 = 11 - 0 - 1 = 10 \text{ ms}$$

$$Tep3 = 2 - 0 - 2 = 0 \text{ ms}$$

$$TM = \frac{3 + 10}{3} = 4.33 \text{ ms}$$

8. Afinidade com o processador é quando o processo busca sempre ser executado no mesmo processador, podemos ter 2 tipo de afinidades, **leve** quando o SO tenta manter o o processo executando no mesmo processador, porem não garante, já o **forte** o SO obriga o processo a executar no mesmo processador

9. **Memory Stall** é o tempo de espera do processador para receber os dados da memória acessada, podendo gastar ate 50% do tempo esprando. Para resolver esse problema atribuímos 2 Threads de hardware para cada nucleo, assim quando um for interrompido, o nucleo passa para a outra Thread, chamamos isso de Hyper-Threading