

### Atividade 3 – Sistemas Operacionais (SO)

Júlia Simone Araújo

1.

Também conhecido como o escalonamento por revezamento, ou fila circular, é originado a partir da adição da preempção por tempo no FCFS (ou FIFO). Isso quer dizer que seu modelo é guiado por fatias de tempo, o quantum, e a partir disso levamos em consideração uma fila de tarefas prontas. Quando o quantum determina o fim do período de execução, a tarefa mesmo não finalizada vai para o fim da fila e assim segue a execução até o fim de todas as tarefas previstas. Utilizado em sistemas operacionais multitarefas, tem como motivação principal os sistemas time-sharing devido ao seu temporizador (quantum).

2.

Sabendo que:

$$E = T_q / T_q + T_{tc} \text{ e}$$

$$T_{tc} = p/100 \cdot T_q, \text{ temos:}$$

$$E = T_q / T_q + (p/100 \cdot T_q) \text{ e assim: } E = 100 \cdot T_q / T_q (100 + p), \text{ que simplificando fica: } E = 100 / 100 + p.$$

3.

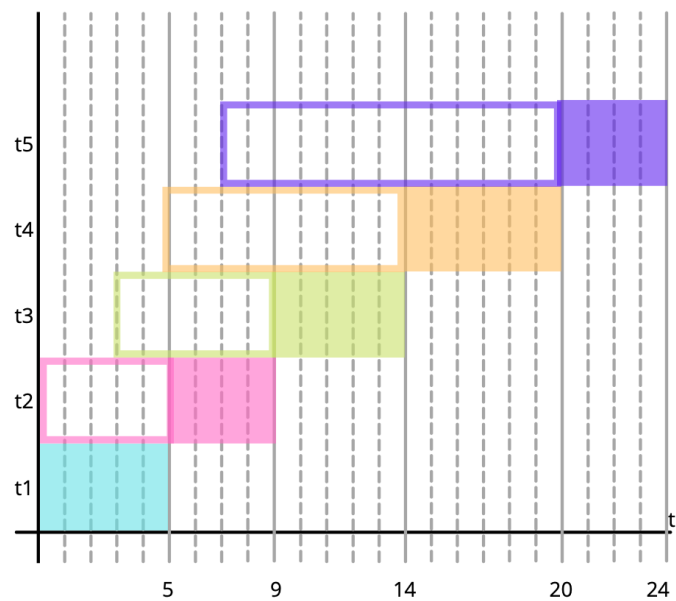
Primeiro precisamos reconhecer o problema de prioridade Starvation, quando reconhecemos a possibilidade de algumas tarefas de baixa prioridade serem sempre colocadas para o fim da fila, por surgir prioridades acima dela.

Por isso, o Task Aging (ou também Envelhecimento) vem como uma proposta de “envelhecer” a idade da tarefa permitindo um reajuste na prioridade da tarefa. Isso é feito por meio de um fator de envelhecimento, ao longo da execução do sistema computacional.

4.

Quando tratamos de uma escala de prioridade negativa, quer dizer que o processador levará em consideração os valores numéricos de menor de prioridade para indicar o maior valor. Com isso, seria necessário decrementar o fator de envelhecimento, em vez de somá-lo.

5. a)



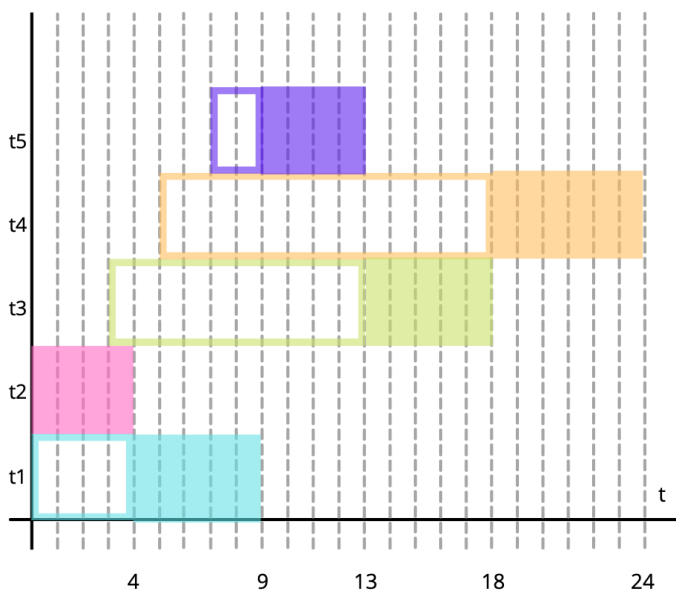
Tempo médio de execução

$$T_t = (5-0) + (9-0) + (14-3) + (20-5) + (24-7) / 5 = 11,4 \text{ s}$$

Tempo médio de espera

$$T_e = (0-0) + (5-0) + (9-3) + (14-5) + (20-7) / 5 = 6,6 \text{ s}$$

b)



Tempo médio de execução:

$$T_t = (9-0) + (4-0) + (18-3) + (24-5) + (13-7) / 5 = 10,6 \text{ s}$$

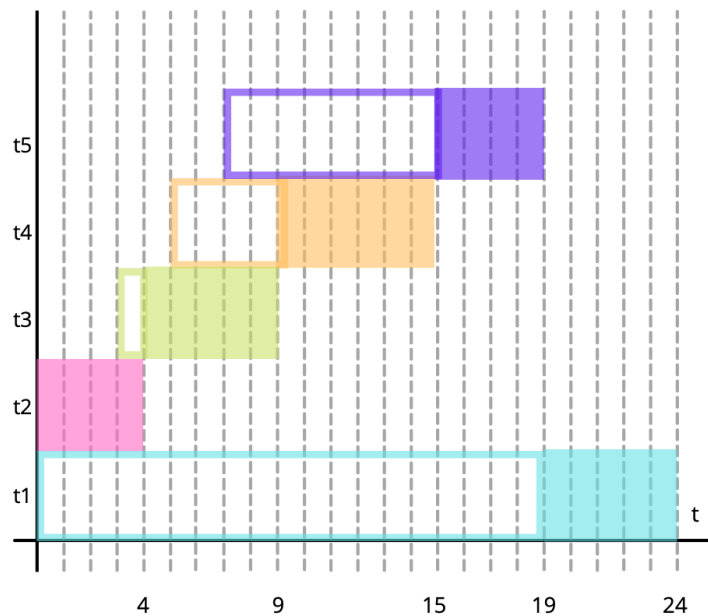
Tempo médio de espera:

$$Te = (4-0) + (0-0) + (10-3) + (13-5) + (2-7) = 2,8 \text{ s}$$

c)

No caso da tabela apresentada, não houve tarefa com menor tempo restante ao longo da execução para que pudesse ocorrer a repriorização de execução. Isso quer dizer, que ela se mantém igual ao caso de SJF cooperativa.

d)



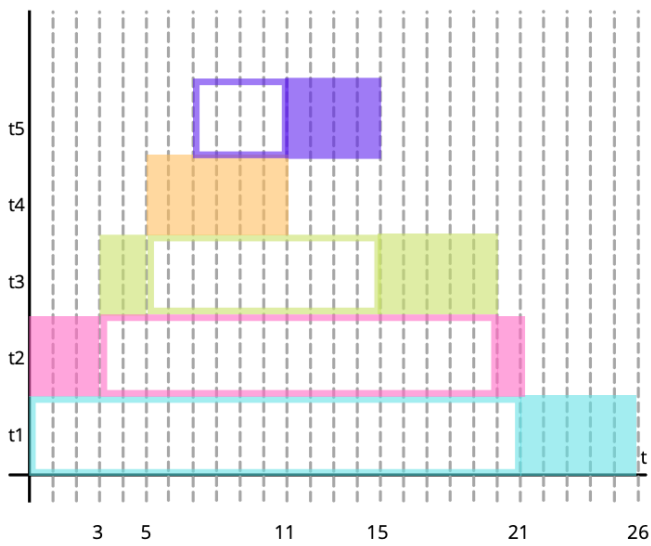
Tempo médio de execução:

$$Tt = (24-0) + (4-0) + (9-3) + (15-5) + (19-7)/5 = 9,2 \text{ s}$$

Tempo médio de espera:

$$Te = (19-0) + (0-0) + (1-3) + (4-5) + (8-7)/5 = 3,4 \text{ s}$$

e)



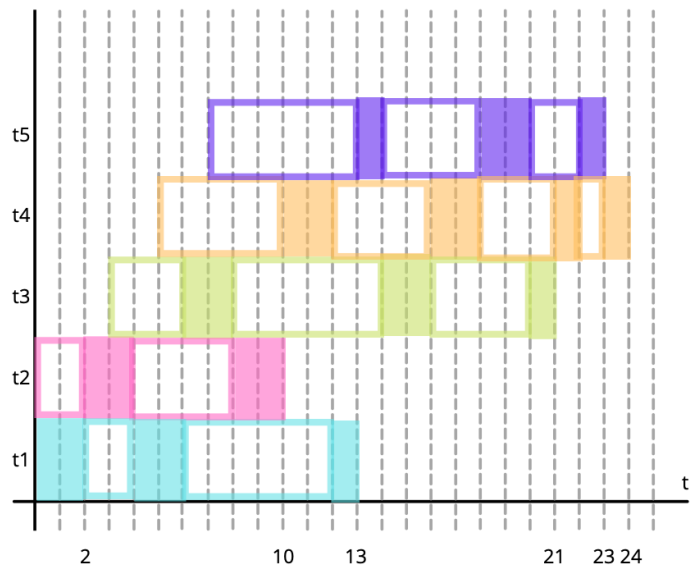
Tempo médio de execução:

$$T_t = (26-0) + (21-0) + (20-3) + (11-5) + (15-7)/5 = 15,6 \text{ s}$$

Tempo médio de espera:

$$T_e = (21-0) + (17-0) + (10-3) + (0-5) + (4-7)/5 = 7,4 \text{ s}$$

f)



Tempo médio de execução:

$$T_t = (13-0) + (10-0) + (21-3) + (24-5) + (23-7)/5 = 15,2 \text{ s}$$

Tempo médio de espera:

$$T_e = (8-0) + (6-0) + (13-3) + (13-5) + (12-7)/5 = 7,4 \text{ s}$$

6.

Aplicando os valores de uso recente da CPU na fórmula de prioridade, dada na questão, temos como resultado  $P1 = 80$ ,  $P2 = 69$ ,  $P3 = 65$ . A partir disso, pela afirmativa: “quanto mais alto o número, menor a prioridade” entendemos que a priorização será do menor para o maior, ou seja,  $P3 \rightarrow P2 \rightarrow P1$ .

Entendemos que o UNIX tende a diminuir a prioridade daqueles que passam mais tempo na CPU.

7.

Com as atividades de entrada e saída como foco, temos em mente que parte considerável do tempo passam aguardando as operações, no estado de tarefas suspensas. Por isso, é importante que quando elas recebem o que precisam e passam a serem classificadas como tarefas prontas, sejam priorizadas pelo escalonador. Isso é possível graças a filas multiníveis com retroalimentação, por ter como característica favorecer tarefas curtas e limitadas por E/S, garantindo assim uma boa utilização dessa parte do sistema.