

Aluno: João Victor da Silva Prado  
Sistemas Operacionais  
Avaliação 3

1) R: O escalonamento round-robin é chamado também de fila circular ou escalonamento por revezamento. Seu modelo se guia através de fatias de tempo (quantum) e com isso uma fila de tarefas prontas é levada em consideração. Isso ocorre porque esse escalonamento surge da adição da preempção por tempo no FIFO (ou FCFS). Se um processo não termina no tempo estabelecido, ele vai para o fim da fila e a execução continua até o fim dos processos previstos. Um bom exemplo é uma fila em que se tem pessoas (representando processos) e um caixa (representando a CPU); cada pessoa é limitada a "pagar" um número  $x$  de contas por vez e esse  $x$  é o nosso quantum; Se o processo terminar sua execução (todas as contas forem pagas) a pessoa sai da fila, caso contrário, volta para o fim da fila e espera sua vez novamente.

2) R: Sabemos que  $E = \frac{T_q}{T_q + T_{tc}}$  e  $T_{tc} = \frac{P \cdot T_q}{100}$   
então:

$$E = \frac{T_q}{T_q + \frac{P \cdot T_q}{100}} \Rightarrow \frac{T_q}{T_q \left(1 + \frac{P}{100}\right)} \Rightarrow \frac{1}{1 + \frac{P}{100}}$$

$$E = \frac{1}{1 + \frac{P}{100}} \Rightarrow E = \frac{100}{100 + P}$$

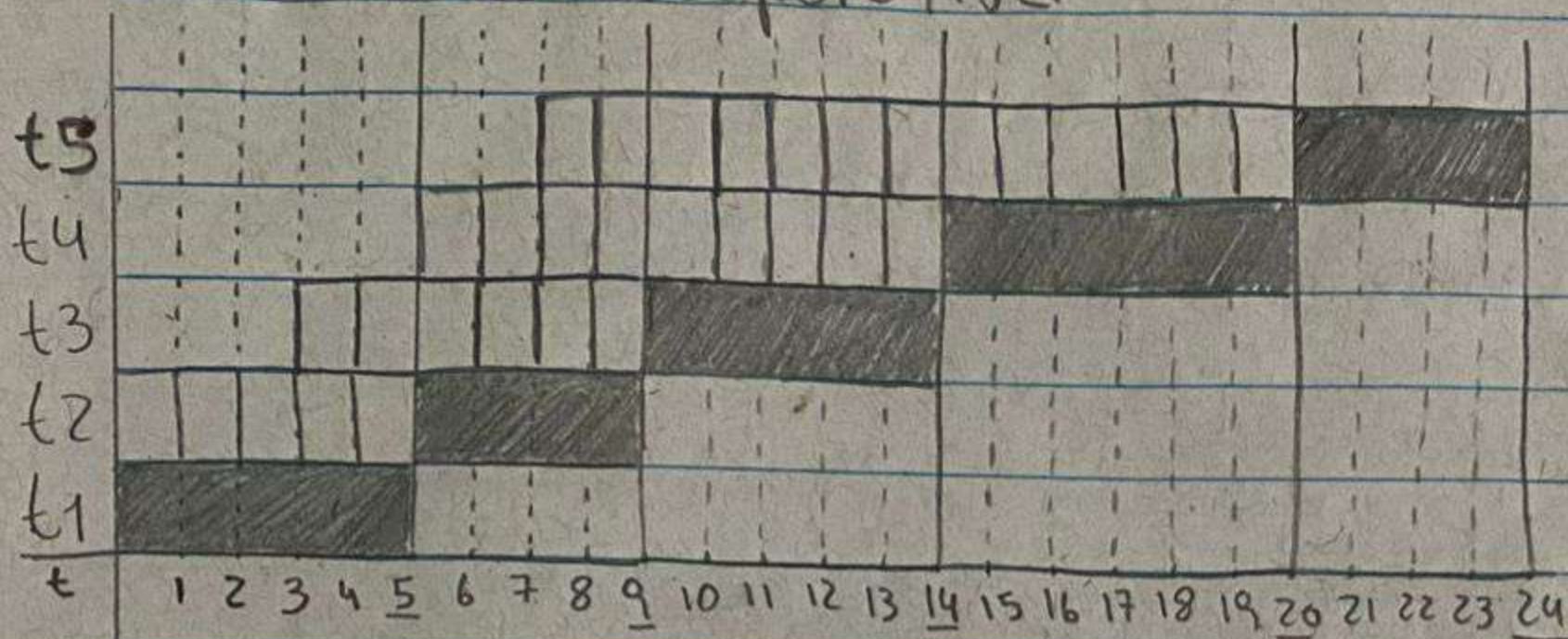


3) R: Um dos principais problemas do escalonamento circular é o starvation (que é quando um processo nunca é executado; pois processos de prioridade maior sempre o impedem de ser executado). Uma solução para esse problema é a técnica de aging. O aging tem a proposta de "envelhecer" a idade da tarefa de maneira que possa ser feito um reajuste na prioridade da tarefa, ou seja, a prioridade de uma tarefa aumentará gradualmente, levando em conta seu tempo de espera na fila de espera.

4) R: No cenário em que há escala de prioridades negativa o processador estará considerando os números com a prioridade mais baixa para definir o maior valor. Dessa forma seria preciso que o fator de envelhecimento fosse decrementado e não somado.



### 5) a) FCFS cooperativa



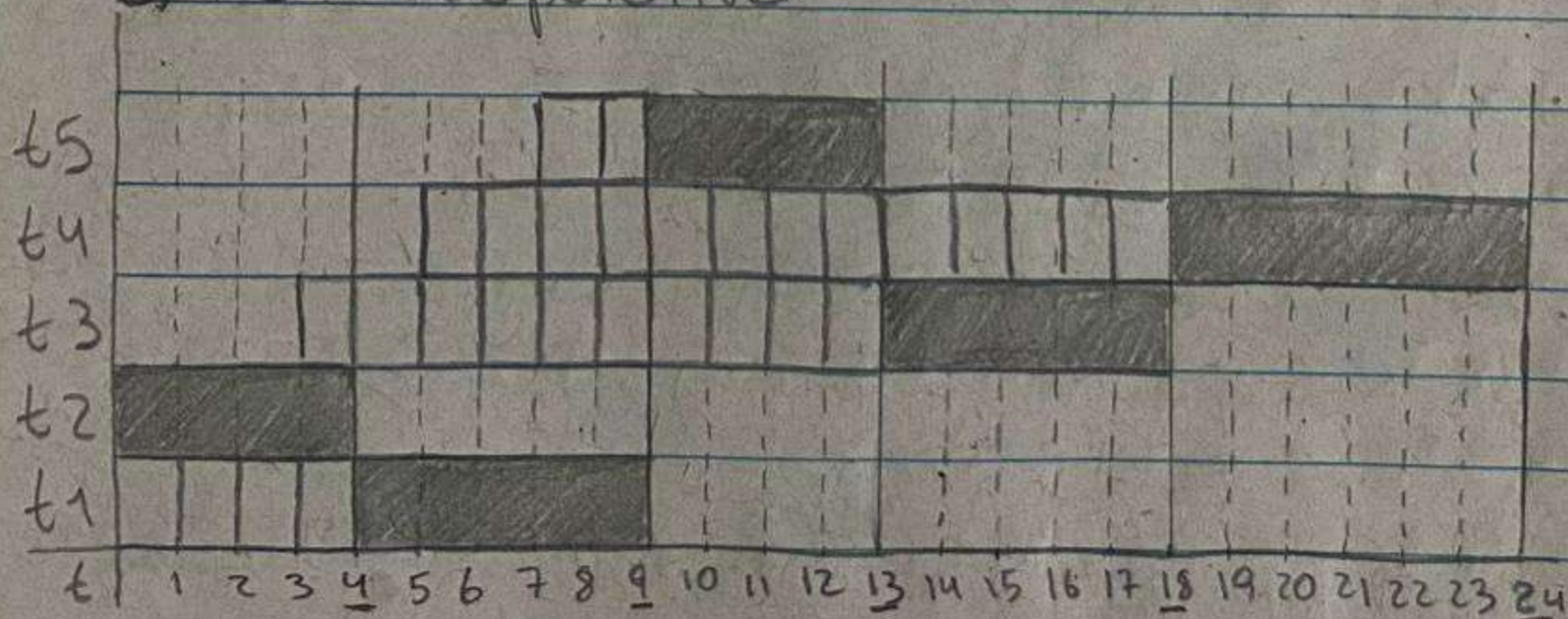
\* tempo médio de execução:

$$T_t = \frac{(5-0) + (9-5) + (14-9) + (20-14) + (24-20)}{5} \Rightarrow T_t = 11,4s$$

\* tempo médio de espera:

$$T_e = \frac{(0-0) + (5-0) + (9-5) + (14-9) + (20-14)}{5} \Rightarrow T_e = 6,6s$$

### b) SJF cooperativa



\* tempo médio de execução:

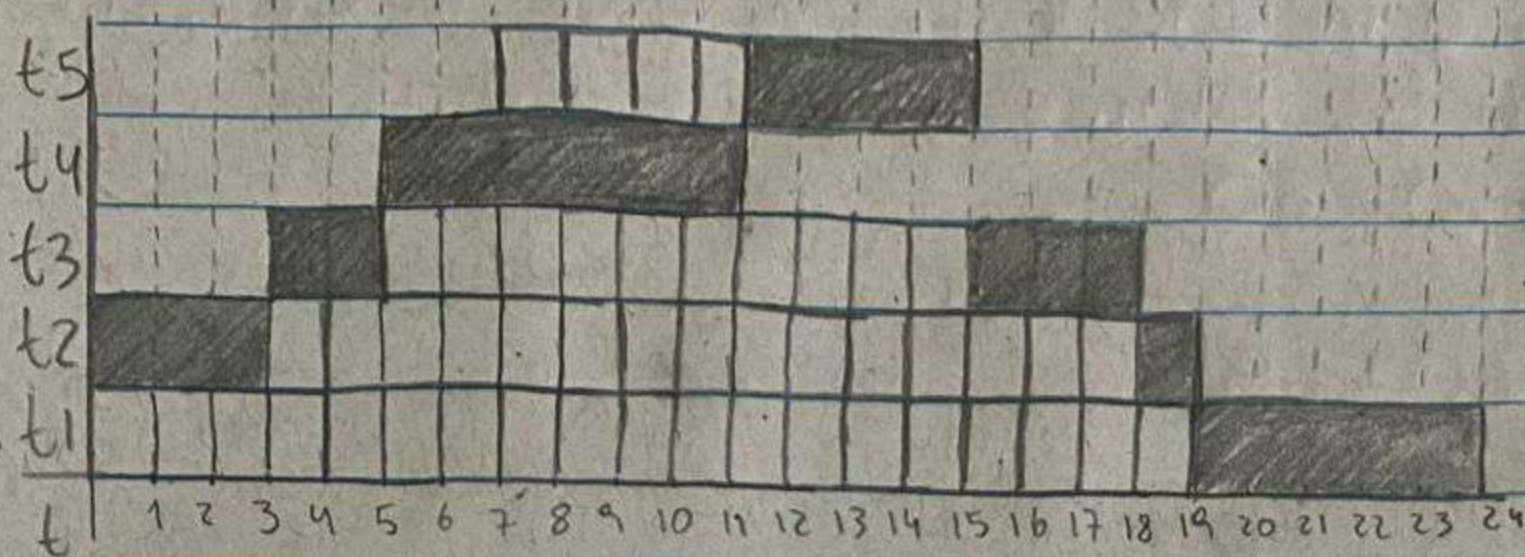
$$T_t = \frac{(4-0) + (9-4) + (13-9) + (18-13) + (24-18)}{5} \Rightarrow T_t = 10,6s$$

\* tempo médio de espera:

$$T_e = \frac{(0-0) + (4-0) + (9-4) + (13-9) + (18-13)}{5} \Rightarrow T_e = 5,8s$$



e) PRIO preemptiva



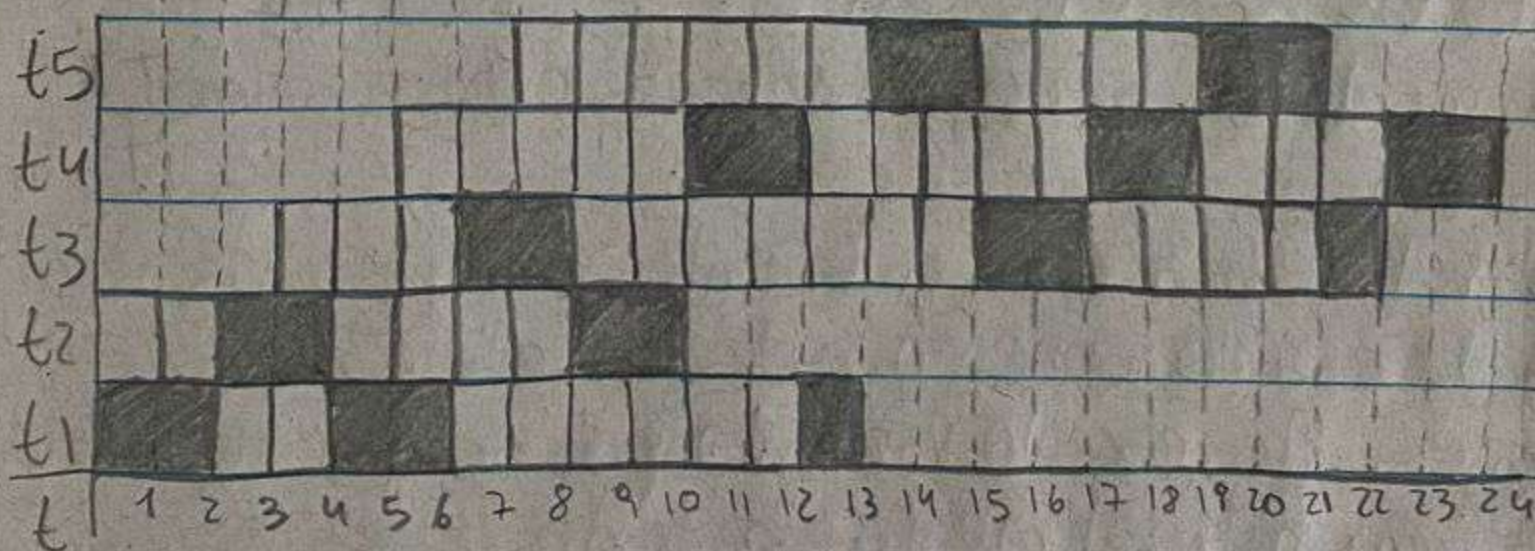
\* Tempo médio de execução:

$$T_t = \frac{(24-0) + (19-0) + (18-3) + (11-5) + (15-7)}{5} \Rightarrow T_t = 14,4s$$

\* Tempo médio de espera:

$$T_e = \frac{(19-0) + [(0-0) + (18-3)] + [(3-3) + (15-5)] + (5-5) + (11-7)}{5} \Rightarrow T_e = 9,6s$$

f) RR com  $t_q = 2$ , sem envelhecimento



\* Tempo médio de execução:

$$T_t = \frac{(13-0) + (10-0) + (22-3) + (24-5) + (21-7)}{5} \Rightarrow T_t = 15s$$

\* Tempo médio de espera:

$$T_e = \frac{\{[(0-0) + (4-2) + (12-6)] + [(2-0) + (8-4)] + [(6-3) + (5-8) + (21-17)] + [(10-5) + (7-12) + (22-19)] + [(13-7) + (19-15)]\}}{5} = 10,2s$$



6) R: Sabemos que a prioridade será a soma da base com a metade do uso recente da CPU, então aplicando os valores dados teremos:

$$P_{p1} = \left(\frac{40}{2}\right) + 60 = 80$$

$$P_{p2} = \left(\frac{18}{2}\right) + 60 = 69$$

$$P_{p3} = \left(\frac{10}{2}\right) + 60 = 65$$

Como o escalonador tradicional do UNIX estabelece um relacionamento em que "quanto mais alto o número, menor a prioridade", a prioridade relativa de um processo limitado pela CPU vai ser rebaixada. Já no caso de um processo limitado por E/S a prioridade relativa será elevada pois os módulos de E/S tem frequência menor que processos limitados pela CPU, então o valor de prioridade desses módulos seria menor.

7) R: Por existir tarefas com diferentes tempos de resposta, a forma de escalonamento poderá adotar necessidades diferentes. No escalonamento por meio de filas Multiníveis sem Retroalimentação o processo nunca é trocado de fila, já no com Retroalimentação o processo pode ser trocado de fila. Esse segundo permite a separação de processos com características de picos de CPU semelhantes (um processo que usa muito tempo de CPU é movido para fila de mais baixa prioridade; assim processos IO-bound e interativos ficam nas filas com mais prioridade; Processos que aguardam muito tempo por CPU podem ser movidos também para as filas



de mais alta prioridade, evitando starvation). O escalonador é definido por alguns parâmetros como: número de filas; algoritmos de escalonamento para cada fila; método que determina quando elevar um processo e quando rebaixar um processo; método usado para definir em que fila um processo entrará quando ele precisar de atendimento.