1 - No Round-Robin cada processo recebe um intervalo de tempo, chamado de quantum, durante o qual ele pode executar. Caso ocorra de o tempo dado a ele acabar e o processo não finalizar a execução, a cpu interrompe este processo e dá a vez ao próximo processo na fila de execução. Caso o processo bloqueie ou termine antes do final do quantum, a troca da cpu para o outro processo ocorre assim que for realizado este bloqueio ou ocorrer o término do processo. Por ser preemptivo, o Round-Robin distribui o uso do processador ao longo do tempo, possibilitando a ele atender mais tarefas interativas.
Exemplo: Se o quantum é 100 milissegundos e a tarefa leva 250 milissegundos para completar, o agendamento round-robin suspenderá a tarefa após os primeiros 100 milissegundos, a tarefa em execução irá para o final da fila e o sistema dará a outra tarefa da fila a mesma quantidade de tempo. A tarefa seguirá retornando à fila até completar sua execução(os 250 milissegundos) , o mesmo vale para qualquer outra tarefa que não finalize sua execução durante os 100 milissegundos iniciais.

2- Sabendo que: E = Tq / (Tq + Ttc) e Ttc = p/(100 * Tq), temos: E = Tq/Tq + (p/100 * Tq)
e assim: E = 100 * Tq/ Tq (100 + p), que simplificando fica: E= 100/100 +p.

3 - É utilizada para ajustar o valor da prioridade de uma tarefa, como forma de evitar inanição. Basicamente ela aumenta a prioridade da tarefa conforme o tempo de execução vai passando e ela não vai sendo executada, agindo da seguinte forma: Sempre que uma tarefa fica pronta, é atribuída a ela uma prioridade estática inicial(default) e uma prioridade dinâmica com o mesmo valor da prioridade estática. Sempre que ocorre a nova escolha de tarefas, o sistema vai escolher a com maior prioridade dinâmica, resetar o valor da escolhida para seu valor default e as tarefas que não foram escolhidas tem a sua prioridade dinâmica somada a um fator de envelhecimento x.

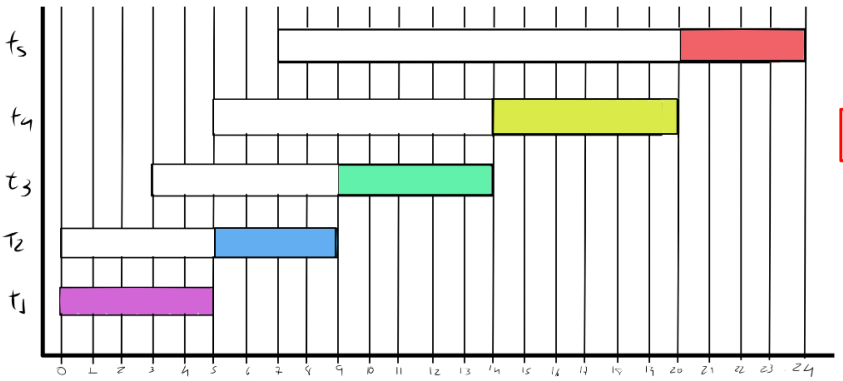
4 - Para um sistema de prioridades negativas, o que pode ser feito é: Ao invés de, resetar a prioridade dinâmica da tarefa escolhida ao valor default, pode ser somado a ela o valor do fator de envelhecimento e as demais continuam com seu valor dinâmico inalterado. Ou, resetar a tarefa escolhida ao valor default e subtrair o fator de envelhecimento das demais tarefas.

5 - Resolução nas páginas finais.

7 - Tarefas limitadas a E/S consomem muito pouco do processador, pois realizam muitas operações de entrada e saída. Então, durante uma fração grande do seu período ativo elas passam aguardando as operações de entrada e saída. Tendo isso em mente, por serem tarefas que consomem muito pouco do processador e passam a maior parte do tempo aguardando na fila de tarefas, é interessante que elas sejam escolhidas pelo escalonador assim que estiverem prontas, tornando então o sistema com filas multiníveis o ideal para

elas.

a) FCFS cooperativo



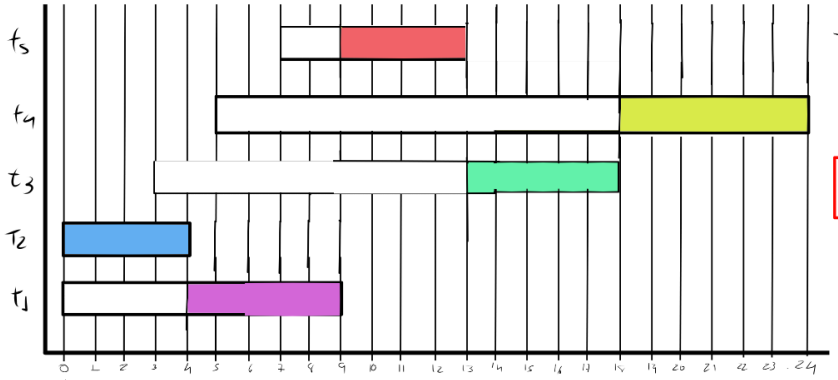
$T_t = (4-0)+(9-0)+(14-3)+(24-7)+(20-5)$ / 5

$T_t = 11,2$ s

$T_w = (0-0)+(5-0)+(9-3)+(14-5)+(20-7)$ / 5

$T_w = 6,6$ s

b) SJF



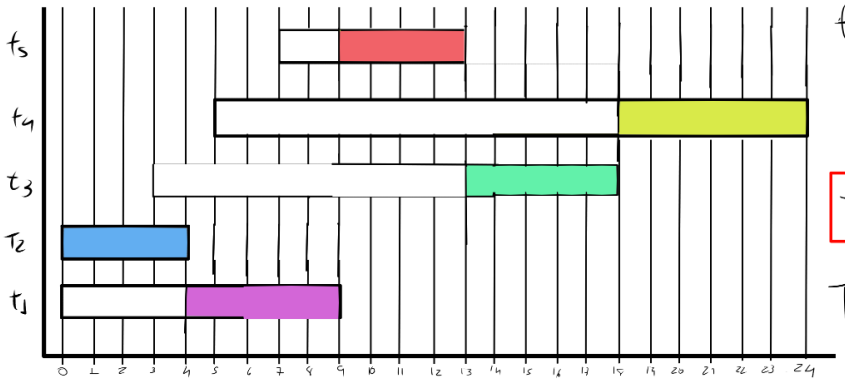
$T_t = (9-0)+(4-0)+(18-3)+(24-5)+(13-7)$ / 5

$T_t = 10,6$ s

$T_w = (4-0)+(0-0)+(13-3)+(18-5)+(9-7)$ / 5

$T_w = 5,8$ s

c) SRTF



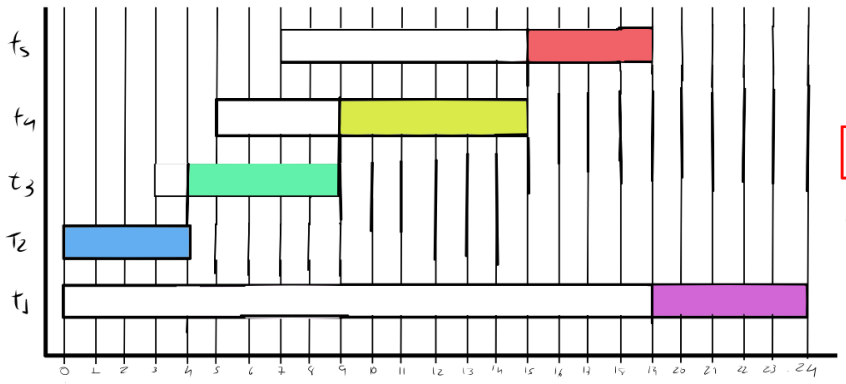
$T_t = (9-0)+(4-0)+(18-3)+(24-5)+(13-7)$ / 5

$T_t = 10,6$ s

$T_w = (4-0)+0+(13-3)+(18-5)+(9-7)$ / 5

$T_w = 5,8$ s

d) PRIO cooperativa



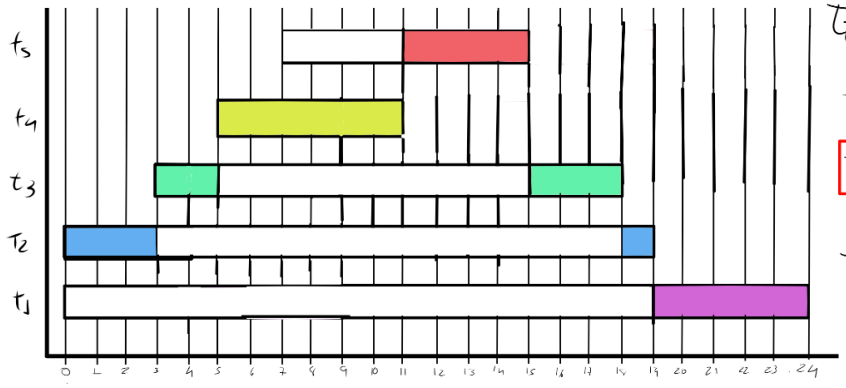
$$t_t = \frac{(24-0)+(4-0)+(9-3)+(15-5)+(19-7)}{5}$$

$t_t = 11,2$ u

$$t_w = \frac{(19-0)+0+(4-3)+(9-5)+(15-7)}{5}$$

$t_w = 7,2$ u

e) PRIO Preemptiva



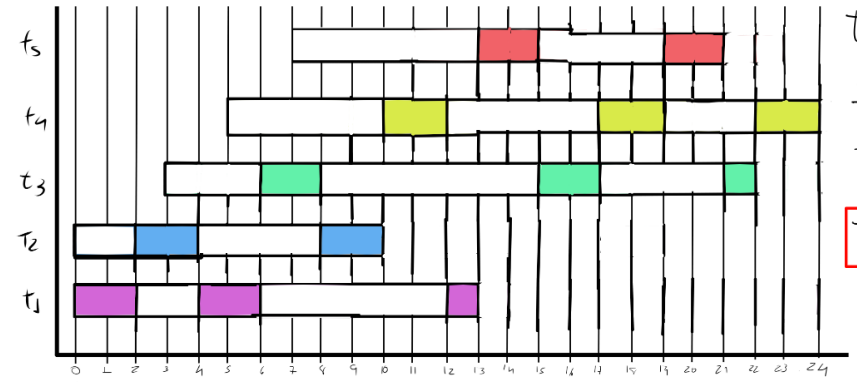
$$t_t = \frac{(24-0)+(19-0)+(18-3)+(11-5)+(15-7)}{5}$$

$t_t = 14,4$ u

$$t_w = \frac{(19-0)+(18-3)+(15-5)+0+(11-7)}{5}$$

$t_w = 9,6$ u

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



$$t_t = \frac{(13-0)+(10-0)+(22-3)+(24-5)+(21-7)}{5}$$

$t_t = 15$ u

$$t_w = \frac{(2+6)+(2+4)+(3+7+4)+(5+5+3)+(6+4)}{5} \Rightarrow t_w = 10,2$$ u