# GSI018 – SISTEMAS OPERACIONAIS

**Operating Systems – William Stallings – 7th Edition Chapter 09 – Uniprocessor Scheduling**

**Pedro Henrique Silva Santana – 12011BSI218 –** [**pedro.santana@ufu.br**](mailto:pedro.santana@ufu.br)

**Victor Hugo Martins Alves – 12011BSI217 – victor.alves1@ufu.br**

## REVIEW QUESTIONS

* 1. What is usually the critical performance requirement in an interactive operating system?

Tempo de Resposta.

* 1. What is the difference between turnaround time and response time?

O **tempo de resposta** é o tempo decorrido entre a submissão da requisição até a resposta, enquanto **turnaround** é o tempo total que a requisição gasta no sistema (tempo de espera e serviço).

**9.5** What is the difference between preemptive and nonpreemptive scheduling?

**Preemptive**: O processo em execução é interrompido e movido para o estado Ready pelo SO. Isso ocorre quando um novo processo aparece, quando uma interrupção ocorre e um processo passa de Blocked para Ready ou ocorra uma interrupção clock.

**Nonpreemptive**: O processo em execução continua até ser finalizado ou bloqueado para esperar pelo I/O ou requisitar um serviço do SO.

## PROBLEMS

**9.1** Consider the following set of processes:   
A .. arrival time = 0; processing time = 3;

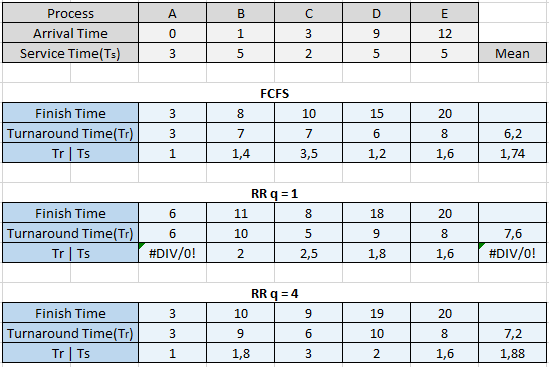
B .. arrival time = 1; processing time = 5;

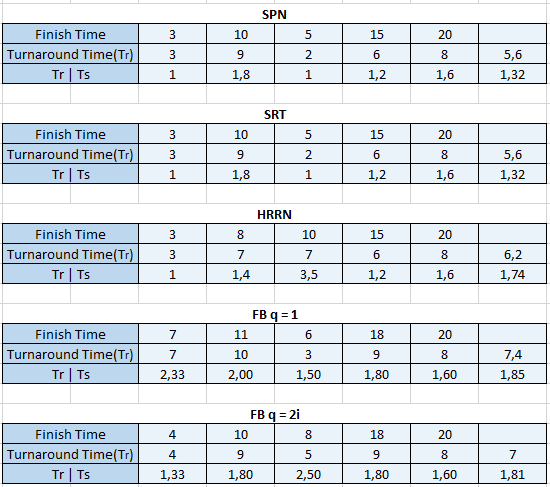
C .. arrival time = 3; processing time = 2;

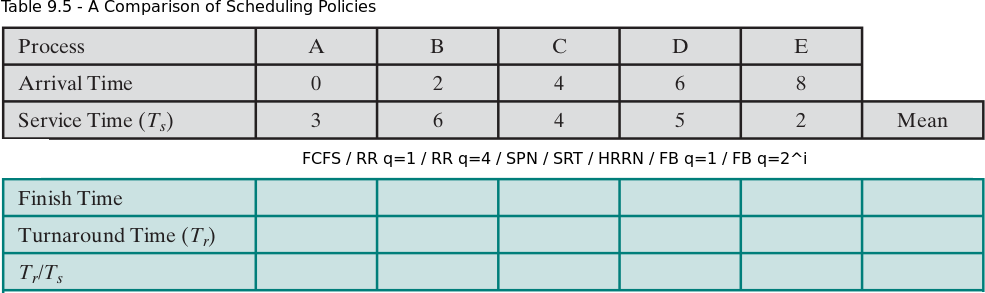
D .. arrival time = 9; processing time = 5;

E .. arrival time = 12; processing time = 5;

Perform the same analysis as depicted in Table 9.5 and Figure 9.5 for this set.







**9.3** Prove that, among nonpreemptive scheduling algorithms, SPN provides the minimum average waiting time for a batch of jobs that arrive at the same time. Assume that the scheduler must always execute a task if one is available.

Considerando que vários jobs chegam ao mesmo tempo, o algoritmo que tem o menor tempo de espera é aquele que gasta menos tempo para selecionar o processo a ser executado. Neste quesito o **Shortest Process Next** é o que possui melhor desempenho, pois ele faz o cálculo de uma somatória para estimar o menor processo e coloca esse processo na cabeça da lista imediatamente para assim que possível ser executado.

**9.6** In the bottom example in Figure 9.5 , process A runs for two time units before control is passed to process B. Another plausible scenario would be that A runs for three time units before control is passed to process B. What policy differences in the feedback-scheduling algorithm would account for the two different scenarios ?

A política que permite isso ocorra é a de escalonamento de maneira preemptiva (com quantum de tempo) e um mecanismo de prioridade dinâmica. Que faz com que um processo que entre primeiro seja colocado em RQ0 e após sua primeira preempção ao retornar ao estado Ready é colocado em RQ1 e a cada vez subsequente que isso acontece ele é rebaixado na fila de prioridade, a qual permite que outros processos menores sejam concluídos rapidamente sem descer muito na fila de prioridade.

