

GS1018 – SISTEMAS OPERACIONAIS

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

Pedro Henrique Silva Santana – 12011BSI218 – pedro.santana@ufu.br
Victor Hugo Martins Alves – 12011BSI217 – victor.alves1@ufu.br

REVIEW QUESTIONS

9.2 What is usually the critical performance requirement in an interactive operating system?

Tempo de Resposta.

9.3 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; processingtime = 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.

Process	A	B	C	D	E	
Arrival Time	0	1	3	9	12	
Service Time(T_s)	3	5	2	5	5	Mean
FCFS						
Finish Time	3	8	10	15	20	
Turnaround Time(T_r)	3	7	7	6	8	6,2
$T_r \mid T_s$	1	1,4	3,5	1,2	1,6	1,74
RR q = 1						
Finish Time	6	11	8	18	20	
Turnaround Time(T_r)	6	10	5	9	8	7,6
$T_r \mid T_s$	#DIV/0!	2	2,5	1,8	1,6	#DIV/0!
RR q = 4						
Finish Time	3	10	9	19	20	
Turnaround Time(T_r)	3	9	6	10	8	7,2
$T_r \mid T_s$	1	1,8	3	2	1,6	1,88

SPN						
Finish Time	3	10	5	15	20	
Turnaround Time(T_r)	3	9	2	6	8	5,6
$T_r T_s$	1	1,8	1	1,2	1,6	1,32
SRT						
Finish Time	3	10	5	15	20	
Turnaround Time(T_r)	3	9	2	6	8	5,6
$T_r T_s$	1	1,8	1	1,2	1,6	1,32
HRRN						
Finish Time	3	8	10	15	20	
Turnaround Time(T_r)	3	7	7	6	8	6,2
$T_r T_s$	1	1,4	3,5	1,2	1,6	1,74
FB $q = 1$						
Finish Time	7	11	6	18	20	
Turnaround Time(T_r)	7	10	3	9	8	7,4
$T_r T_s$	2,33	2,00	1,50	1,80	1,60	1,85
FB $q = 2^i$						
Finish Time	4	10	8	18	20	
Turnaround Time(T_r)	4	9	5	9	8	7
$T_r T_s$	1,33	1,80	2,50	1,80	1,60	1,81

Table 9.5 - A Comparison of Scheduling Policies

Process	A	B	C	D	E	
Arrival Time	0	2	4	6	8	
Service Time (T_s)	3	6	4	5	2	Mean

FCFS / RR $q=1$ / RR $q=4$ / SPN / SRT / HRRN / FB $q=1$ / FB $q=2^i$

Finish Time						
Turnaround Time (T_r)						
T_r/T_s						

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 ocorrer 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.

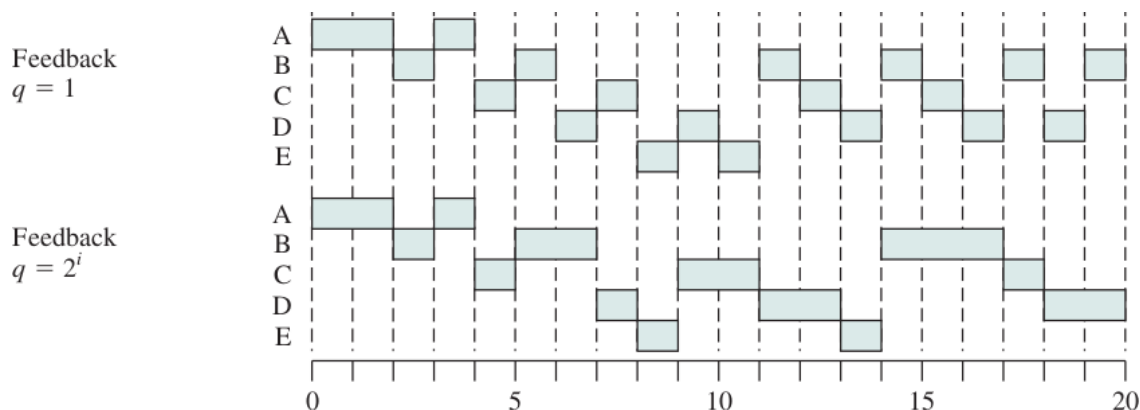


Figure 9.5 A Comparison of Scheduling Policies