

GS1018 – SISTEMAS OPERACIONAIS

Operating Systems – William Stallings – 7th Edition Chapter 10 – Multiprocessor and Real-Time Scheduling

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

REVIEW QUESTIONS

10.1 List and briefly define five different categories of synchronization granularity.

Paralelismo independente: não há sincronização explícita entre processos, ou seja, cada processo representa uma tarefa.

Granularidade grosseira ou muito grosseira: há sincronização entre processos, mas em um nível grosseiro, que em instruções representa de 200 a 1.000.000 de instruções.

Granularidade média: constitui-se uma coleção de threads dentro de um único processo. Programadores devem explicitar o paralelismo potencial de uma aplicação, exigindo assim um grau maior de coordenação e interação entre as threads de uma mesma aplicação.

Granularidade fina: contempla o uso mais complexo do paralelismo do que o encontrado entre threads, ou seja, paralelismo inerente ao fluxo de instruções.

10.2 List and briefly define four techniques for thread scheduling.

Load sharing: mantém-se uma fila global para “threads” e cada processador quando ocioso, seleciona uma “thread” da fila.

Gang scheduling: conjunto de “threads” relacionadas é escalonado para ser executado em um conjunto de processadores ao mesmo tempo no esquema um por um.

Dedicated processor: contempla escalonamento implícito tendo por base as associações de “threads” aos processadores.

Dynamic scheduling: número de “threads” por processo pode ser alterado no curso de seu ciclo de vida.

10.4 What is the difference between hard and soft real-time tasks?

Hard real-time task: a deadline deve ser atendida imediatamente, caso contrário causará erro fatal ou dano inaceitável.

Soft real-time task: o atendimento do deadline é desejável, mas não mandatório, assim há sentido ao atender a tarefa ainda que o “deadline” não tenha sido satisfeito no tempo esperado.

10.5 What is the difference between periodic and aperiodic real-time tasks?

Periodic task: o deadline deve ser atendido uma vez a cada período de tempo “T” ou exatamente a cada “T” unidades de tempo.

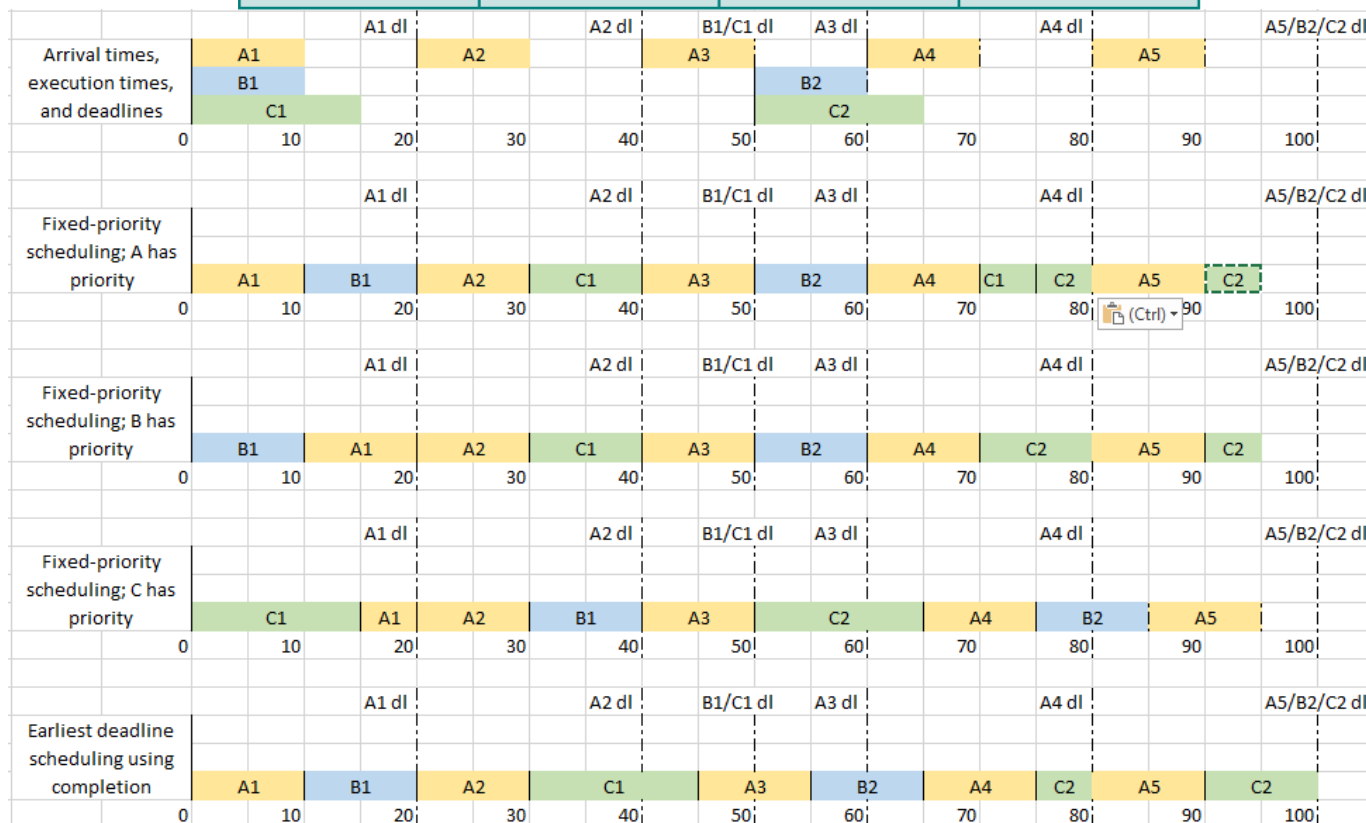
Aperiodic task: o “deadline” deve ser atendido uma única vez, seja no início ou no final de uma marca de tempo e sem repetição.

PROBLEMS

10.1 Consider a set of three periodic tasks with the execution profiles of Table 10.6. Develop scheduling diagrams similar to those of Figure 10.5 for this set of tasks.

Table 10.6 Execution Profile for Problem 10.1

Process	Arrival Time	Execution Time	Ending Deadline
A(1)	0	10	20
A(2)	20	10	40
•	•	•	•
•	•	•	•
•	•	•	•
B(1)	0	10	50
B(2)	50	10	100
•	•	•	•
•	•	•	•
•	•	•	•
C(1)	0	15	50
C(2)	50	15	100
•	•	•	•
•	•	•	•
•	•	•	•



10.2 Consider a set of five aperiodic tasks with the execution profiles of Table 10.7. Develop scheduling diagrams similar to those of Figure 10.6 for this set of tasks.

Table 10.7 Execution Profile for Problem 10.2

Process		Arrival Time		Execution Time		Starting Deadline	
A		10		20		100	
B		20		20		30	
C		40		20		60	
D		50		20		80	
E		60		20		70	

		0	10	20	30	40	50	60	70	80	90	100	110	120
Requirements	Arrival times		A	B		C	D	E						
	Starting deadline				B			C	E	D		A		

		0	10	20	30	40	50	60	70	80	90	100	110	120
Earliest deadline	Arrival times		A	B		C	D	E						
	Service		A			C			E		D			
	Starting deadline				B			C	E	D		A		

		0	10	20	30	40	50	60	70	80	90	100	110	120
Earliest deadline with unforced idle times	Arrival times		A	B		C	D	E						
	Service			B			C			E		D		A
	Starting deadline				B			C	E	D		A		

		0	10	20	30	40	50	60	70	80	90	100	110	120
First-come first-served(FCFS)	Arrival times		A	B		C	D	E						
	Service		A			C			D					
	Starting deadline			B(Missed)				C	E(Missed)		D		A	