

Aula 5 – Sincronismo

Multiprogramação

- Programação concorrente
 - execução de forma colaborativa de múltiplos processos ou threads que trabalham em uma mesma tarefa
 - aplicação concorrente compartilha recursos

Multiprogramação

- Programação concorrente
 - sincronizados com o objetivo de garantir o processamento correto dos programas
 - Evitar condições de disputa entre processos/threads

Condições de Disputa

- Compartilhamento de um arquivo em disco
 - ler e atualizar o mesmo arquivo de forma correta para evitar dados inconsistentes
- Compartilhamento de uma variável de memória
 - não alterar o valor da variável utilizada pelo outro processo

Exemplo 1

- Heisenbug

Thread 1	Thread 2		Valor
			0
lê valor		←	0
incrementa			0
escreve		→	1
	lê valor	←	1
	incrementa		1
	escreve	→	2

Thread 1	Thread 2		Valor
			0
lê valor		←	0
	lê valor	←	0
incrementa			0
	incrementa		0
escreve		→	1
	escreve	→	1

Exemplo 2

- Race condition

Processo A

```
i=0
while i<10:
    i+=1
print("A ganhou")
```

Processo B

```
i=0
while i>-10:
    i-=1
print("B ganhou")
```

- A variável i é compartilhada.
- Quem vai ganhar: processo a ou b?
- Pode-se garantir que eles vão terminar?
- Se um terminar, o outro também vai terminar?
- Ajuda se um começar bem na frente do outro?

Operação Atômica

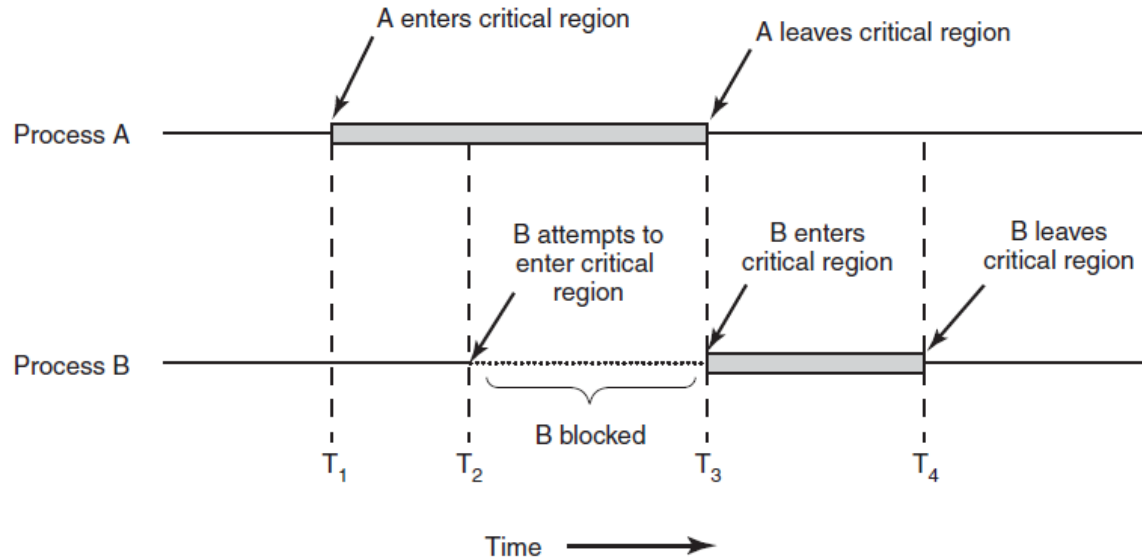
- Operações que não podem ser interrompidas
 - Não é possível ver a operação “em progresso”
 - Mas apenas seu efeito final
- Atômica:
 - Tocar campainha
 - Desligar a luz
- Não Atômica:
 - Encher copo de água
 - Caminhar até a porta

Região Crítica

- Uma seção de código ou uma coleção de operações
 - Apenas um processo pode executar de cada vez
- O objetivo é tornar atômico esse conjunto de operações

Exclusão Mútua

- Apenas um processo pode fazer alguma coisa em determinado momento
 - Os demais são excluídos



Exclusão Mútua

- Duas formas
 - Espera ocupada
 - Dormir e acordar

Como evitar Race Conditions?

- 1) Dois processos não podem estar ao mesmo tempo na região crítica
- 2) Não depender da velocidade ou número de CPUs
- 3) Processos fora da região crítica não podem bloquear outros processos
- 4) Nenhum processo pode esperar pra sempre

