

# Universidade Federal do Ceará - Campus Russas

## Ciência da Computação/Engenharia de Software

### Trabalho Prático de Sistemas Operacionais

#### Professor: Pablo Soares

**DESCRIÇÃO:** Desenvolver um simulador para a detecção de **Impasses (Deadlocks)**. O seu simulador deve receber como entrada um arquivo padronizado de acordo com o arquivo entrada.txt em anexo, e que também está disponível no perfil da turma no Sigaa. Como resultado o simulador deve mostrar como resposta uma das três opções a seguir:

1. Existe processos em Deadlock (2 processos ou mais);
  - a. Imprimir quais os processos que estão **em deadlock**;
  - b. Imprimir quantas instâncias de qual(is) recursos estão em falta em cada processo;
2. Não existe processo em DeadLock
  - a. Imprimir “Todos os processos foram finalizados”
3. Existe um único processo esperando por recursos externos;
  - a. Imprimir qual o processo que está **em espera**;
  - b. Imprimir quantas instâncias de qual(is) recursos estão em falta;

Informações adicionais: Você deve assumir que se existir recursos disponíveis para um processo  $P$  qualquer, o mesmo irá obter esses recursos, irá executar e ao terminar a execução irá devolver **todos** os recursos que estavam alocados para ele.

Informações adicionais 2: Seu simulador não precisa considerar a questão de estados **seguros**<sup>1</sup> e **inseguros**<sup>2</sup> (mas caso você queira implementar esse item fique à vontade).

1. Os Grupos serão de no máximo 3 alunos;
2. Linguagem de programação é livre;
3. Data da apresentação (testar com os exemplos do Professor): **03/11/2022**;
4. Não é preciso entregar o código do simulador;
5. É importante que **TODOS** os componentes do grupo estejam de fato entendendo sobre o assunto em questão, além de saber o que se passa na codificação do simulador desenvolvido (Professor fará perguntas aleatórias a qualquer momento).

Bons Estudos  
Divirtam-se  
e Boa Sorte

---

<sup>1</sup> Verificar o livro do Tanenbaum a partir da página 279 ou notas de aula.

<sup>2</sup> Verificar o livro do Tanenbaum a partir da página 279 ou notas de aula.

## ANEXO - EXEMPLO DE ARQUIVO DE ENTRADA

3 4

4 2 3 1

2 1 0 0

0 0 1 0

2 0 0 1

0 1 2 0

2 0 5 1

1 0 1 0

2 1 0 0

Explicação do Arquivo de *Entrada.txt*

\*: **3 4** → 3 Processos ( $P_1$ ,  $P_2$  e  $P_3$ ) e 4 Recursos ( $R_1$ ,  $R_2$ ,  $R_3$  e  $R_4$ )

\*: **4 2 3 1** → Vetor de Recursos Existentes

4 Instâncias de  $R_1$ ,

2 Instâncias de  $R_2$ ,

3 Instâncias de  $R_3$ ,

1 Instância de  $R_4$ .

\*: **2 1 0 0** → Vetor de Recursos Disponíveis

2 Instâncias disponíveis de  $R_1$ ,

1 Instância disponível de  $R_2$ ,

0 Instância disponível de  $R_3$ ,

0 Instância disponível de  $R_4$ .

\*: Matriz de alocação de recursos por Processo

**0 0 1 0** →  $P_1$  com 1 instância de  $R_3$

**2 0 0 1** →  $P_2$  com 2 instâncias de  $R_1$  e 1 instância de  $R_4$

**0 1 2 0** →  $P_3$  com 1 instância de  $R_2$  e 2 instâncias de  $R_3$

\*: Matriz de requisição de recursos por Processo

**2 0 5 1** →  $P_1$  requisitando 2 instâncias de  $R_1$ , 5 de  $R_3$  e 1 de  $R_4$

**1 0 1 0** →  $P_2$  requisitando 1 instância de  $R_1$  e 3 de  $R_3$

**2 1 0 0** →  $P_3$  requisitando 2 instâncias de  $R_1$  e  $R_2$

Para o Arquivo *entrada.txt* seu simulador deve imprimir na tela:

O processo  $P_1$  está em espera e aguardando 2 instâncias de  $R_3$