**LISTA DE EXERCÍCIOS**

1. **GERÊNCIAMENTO DE MEMÓRIA**
2. Considere a tabela de página a seguir:

| Tabela de Página | |
| --- | --- |
| p | f |
| 0 | 23 |
| 1 | 7 |
| 2 | 0 |
| 3 | 1 |
| 4 | 13 |
| 5 | 18 |
| 6 | 5 |
| 7 | 22 |
| 8 | 25 |
| 9 | 14 |
| 10 | 3 |
| 11 | 2 |
| 12 | 19 |
| 13 | 31 |
| 14 | 12 |
| 15 | 15 |

1. Qual o tamanho do campo p?
2. É possível deduzir o tamanho de f? Caso positivo, informe o seu tamanho.
3. Faça a tradução dos seguintes endereços lógicos para endereços físicos: 129, 57, 23, 191, 93, 137, 29, 12, 46, 20, 150. Considere que o campo d possui 4 bits.

2. Considere o seguinte esquema de tabelas multinível.

| Tabela de Páginas (nível 1) | |  | Tabela de Páginas # 0 (nível 2) | |  | Tabela de Páginas #1 (nível 2) | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 |  | 0 | 9 |  | 0 | 11 |
| 1 | 3 |  | 1 | 13 |  | 1 | 8 |
| 2 | 0 |  | 2 | 10 |  | 2 | 4 |
| 3 | 2 |  | 3 | 2 |  | 3 | 1 |

|  | |  | Tabela de Páginas #2 (nível 2) | |  | Tabela de Páginas #3 (nível 2) | |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | 0 | 15 |  | 0 | 3 |
|  |  |  | 1 | 0 |  | 1 | 14 |
|  |  |  | 2 | 12 |  | 2 | 6 |
|  |  |  | 3 | 7 |  | 3 | 5 |

1. Qual o tamanho do campo p1 e p2?
2. É possível deduzir o tamanho de f? Caso positivo, informe o seu tamanho.
3. Faça a tradução dos seguintes endereços lógicos para endereços físicos:

27, 202, 190, 15, 116, 162, 29, 12, 47, 5, 132. Considere que o campo d possui 4 bits.

3. Considere o seguinte esquema de tabelas de página invertida.

| f | pid p |  | f | pid p |
| --- | --- | --- | --- | --- |
| 0 | 22 |  | 32 | 53 |
| 1 | 5 |  | 33 | 36 |
| 2 | 12 |  | 34 | 41 |
| 3 | 35 |  | 35 | 61 |
| 4 | 57 |  | 36 | 7 |
| 5 | 43 |  | 37 | 23 |
| 6 | 15 |  | 38 | 56 |
| 7 | 25 |  | 39 | 48 |
| 8 | 26 |  | 40 | 31 |
| 9 | 39 |  | 41 | 58 |
| 10 | 29 |  | 42 | 27 |
| 11 | 62 |  | 43 | 34 |
| 12 | 4 |  | 44 | 52 |
| 13 | 37 |  | 45 | 8 |
| 14 | 33 |  | 46 | 28 |
| 15 | 1 |  | 47 | 40 |
| 16 | 0 |  | 48 | 50 |
| 17 | 45 |  | 49 | 32 |
| 18 | 59 |  | 50 | 6 |
| 19 | 54 |  | 51 | 3 |
| 20 | 38 |  | 52 | 13 |
| 21 | 24 |  | 53 | 11 |
| 22 | 14 |  | 54 | 16 |
| 23 | 2 |  | 55 | 9 |
| 24 | 42 |  | 56 | 17 |
| 25 | 49 |  | 57 | 19 |
| 26 | 47 |  | 58 | 60 |
| 27 | 30 |  | 59 | 63 |
| 28 | 10 |  | 60 | 44 |
| 29 | 55 |  | 61 | 51 |
| 30 | 20 |  | 62 | 18 |
| 31 | 21 |  | 63 | 46 |

1. Qual o tamanho dos campos pid, p e d, considerando as seguintes informações. O sistema operacional executa no máximo 4 processos, cada processo possui 16 páginas e cada página apresenta 8 endereços.
2. É possível deduzir o tamanho de f? Caso positivo, qual o tamanho?
3. Converta os seguintes endereços virtuais em endereços físicos reais: 431, 510, 152, 235, 315, 92, 2, 51, 389.

**2. DEADLOCK**

1. As tabelas a seguir apresentam as matrizes alocação, máximo e o vetor disponível para um conjunto de processos/recursos em um dado sistema operacional. Para cada um dos cenários, verifique se o sistema está ou não em deadlock. Em caso de não deadlock, apresente uma sequência de execução acompanhada do valor do vetor disponível após execução de cada processo. Em caso de deadlock, justifique sua resposta, apresentando a matriz necessário.

| A) | Disponível | | | | | | |  | B) | Disponível | | | | | | |  | C) | Disponível | | | | | | |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | | B | | | C | | A | | B | | | C | | A | | B | | | C | |
| 1 | | 2 | | | 1 | | 1 | | 1 | | | 2 | | 0 | | 1 | | | 3 | |
| Alocação | | | Máximo | | | | Alocação | | | Máximo | | | | Alocação | | | Máximo | | | |
| A | B | C | A | B | | C | A | B | C | A | B | | C | A | B | C | A | B | | C |
| P0 | 2 | 2 | 3 | 5 | 4 | | 3 | P0 | 1 | 2 | 1 | 4 | 3 | | 1 | P0 | 1 | 5 | 0 | 3 | 4 | | 2 |
| P1 | 3 | 1 | 0 | 7 | 2 | | 2 | P1 | 2 | 3 | 1 | 5 | 3 | | 2 | P1 | 1 | 0 | 3 | 2 | 0 | | 5 |
| P2 | 1 | 2 | 0 | 3 | 3 | | 1 | P2 | 1 | 3 | 1 | 2 | 4 | | 6 | P2 | 1 | 1 | 0 | 3 | 2 | | 1 |
| P3 | 0 | 1 | 1 | 2 | 4 | | 2 | P3 | 1 | 0 | 0 | 3 | 4 | | 1 | P3 | 1 | 0 | 2 | 1 | 0 | | 4 |
| P4 | 4 | 1 | 0 | 4 | 2 | | 0 | P4 | 1 | 2 | 2 | 5 | 3 | | 4 | P4 | 1 | 1 | 1 | 5 | 4 | | 5 |

| D) | Disponível | | | | | | |  | E) | Disponível | | | | | | |  | F) | Disponível | | | | | | |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | | B | | | C | | A | | B | | | C | | A | | B | | | C | |
| 2 | | 2 | | | 3 | | 0 | | 4 | | | 2 | | 3 | | 1 | | | 0 | |
| Alocação | | | Máximo | | | | Alocação | | | Máximo | | | | Alocação | | | Máximo | | | |
| A | B | C | A | B | | C | A | B | C | A | B | | C | A | B | C | A | B | | C |
| P0 | 2 | 2 | 3 | 2 | 2 | | 3 | P0 | 4 | 2 | 1 | 4 | 4 | | 2 | P0 | 0 | 4 | 0 | 3 | 4 | | 2 |
| P1 | 3 | 1 | 0 | 5 | 1 | | 2 | P1 | 2 | 3 | 1 | 6 | 3 | | 3 | P1 | 1 | 0 | 3 | 2 | 2 | | 5 |
| P2 | 1 | 2 | 0 | 3 | 3 | | 1 | P2 | 2 | 3 | 1 | 2 | 4 | | 6 | P2 | 1 | 1 | 0 | 3 | 1 | | 1 |
| P3 | 2 | 1 | 1 | 2 | 3 | | 2 | P3 | 1 | 0 | 0 | 2 | 3 | | 1 | P3 | 1 | 0 | 2 | 1 | 0 | | 4 |
| P4 | 4 | 1 | 0 | 4 | 2 | | 0 | P4 | 1 | 2 | 2 | 5 | 3 | | 4 | P4 | 1 | 1 | 1 | 4 | 2 | | 5 |

| G) | Disponível | | | | | | |  | H) | Disponível | | | | | | |  | I) | Disponível | | | | | | |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | | B | | | C | | A | | B | | | C | | A | | B | | | C | |
| 2 | | 5 | | | 0 | | 1 | | 1 | | | 1 | | 0 | | 0 | | | 3 | |
| Alocação | | | Máximo | | | | Alocação | | | Máximo | | | | Alocação | | | Máximo | | | |
| A | B | C | A | B | | C | A | B | C | A | B | | C | A | B | C | A | B | | C |
| P0 | 2 | 2 | 3 | 4 | 4 | | 3 | P0 | 3 | 5 | 1 | 4 | 6 | | 1 | P0 | 1 | 4 | 0 | 3 | 4 | | 2 |
| P1 | 4 | 1 | 0 | 7 | 1 | | 0 | P1 | 2 | 3 | 1 | 5 | 3 | | 2 | P1 | 1 | 2 | 3 | 2 | 0 | | 5 |
| P2 | 1 | 2 | 3 | 3 | 3 | | 3 | P2 | 1 | 3 | 5 | 2 | 4 | | 6 | P2 | 3 | 2 | 0 | 3 | 2 | | 1 |
| P3 | 2 | 1 | 1 | 2 | 4 | | 2 | P3 | 2 | 3 | 0 | 3 | 4 | | 1 | P3 | 1 | 0 | 3 | 1 | 0 | | 4 |
| P4 | 4 | 1 | 1 | 4 | 2 | | 0 | P4 | 1 | 2 | 2 | 5 | 3 | | 4 | P4 | 3 | 5 | 3 | 5 | 5 | | 5 |

1. **Analise os seguintes gráfos de alocação de recursos. Há presença de deadlock? Justifique sua resposta:**

| **A** | **B** |
| --- | --- |
| **C** | **D** |

**3. SINCRONIZAÇÃO**

1. Operações com semáforos

A seguir, apresentamos uma sequência de operações do semáforo no início e no final das tarefas A, B, C. Considere que cada tarefa executa em um núcleo de processador dedicado. E considere que cada ação (P(Sx), V(Sx) ou .) possui tempo igual a 1T.

|  | Task A | Task B | Task C |
| --- | --- | --- | --- |
| 1 | P(SA) | P(SB) | P(SC) |
| 2 | P(SA) | . | P(SC) |
| 3 | P(SA) | . | P(SC) |
| 4 | . | . | . |
| 5 | . | . | . |
| 6 | . | V(SC) | V(SB) |
| 7 | V(SB) | V(SA) | V(SB) |
| 8 | END | . | V(SA) |
| 9 |  | END | END |

Determine para os 6 casos a,b,c,d,e,f apresentados na tabela abaixo, se e em qual sequência as tarefas são executadas, usando as inicializações das variáveis do semáforo dadas na tabela.

| Semáforos | a) | b) | c) | d) | e) | f) | g) |
| --- | --- | --- | --- | --- | --- | --- | --- |
| SA | 2 | 3 | 2 | 0 | 3 | 1 | 1 |
| SB | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| SC | 2 | 2 | 1 | 3 | 3 | 3 | 1 |

A seguir, apresentamos uma nova sequência de operações do semáforo no início e no final das tarefas A, B, C. Considere que cada tarefa executa em um núcleo de processador dedicado. E considere que cada ação (P(Sx), V(Sx) ou .) possui tempo igual a 1T.

|  | Task A | Task B | Task C |
| --- | --- | --- | --- |
| 1 | P(SA) | P(SB) | P(SC) |
| 2 | P(SA) | P(SA) | P(SC) |
| 3 | V(SA) | . | P(SB) |
| 4 | . | . | . |
| 5 | . | . | . |
| 6 | . | P(SC) | V(SB) |
| 7 | V(SC) | V(SA) | V(SB) |
| 8 | END | END | V(SA) |
| 9 |  |  | END |

Determine para os 6 casos a,b,c,d,e,f apresentados na tabela abaixo, se e em qual sequência as tarefas são executadas, usando as inicializações das variáveis do semáforo dadas na tabela.

| Semáforos | a) | b) | c) | d) | e) | f) | g) |
| --- | --- | --- | --- | --- | --- | --- | --- |
| SA | 2 | 1 | 2 | 0 | 3 | 2 | 1 |
| SB | 0 | 0 | 1 | 0 | 1 | 2 | 1 |
| SC | 2 | 1 | 1 | 2 | 1 | 2 | 1 |

**2. Códigos em Python.**

**Nos seguintes códigos explique o comportamento do código e o conteúdo que será exibido ao final de sua execução.**

| **#A**  **from threading import \***  **import time**  **l=Lock()**  **def wish(name,age):**  **for i in range(3):**  **l.acquire()**  **print("Hi",name)**  **time.sleep(2)**  **print("Your age is",age)**  **l.release()**  **t1=Thread(target=wish, args=("Sireesh",15))**  **t2=Thread(target=wish, args=("Nitya",20))**  **t1.start()**  **t2.start()** |
| --- |
| **#B**  **from threading import \***  **import time**  **s=Semaphore(2)**  **def wish(name,age):**  **for i in range(3):**  **s.acquire()**  **print("Hi",name)**  **time.sleep(2)**  **s.release()**  **t1=Thread(target=wish, args=("Sireesh",15))**  **t2=Thread(target=wish, args=("Nitya",20))**  **t3=Thread(target=wish, args=("Shiva",16))**  **t4=Thread(target=wish, args=("Ajay",25))**  **t1.start()**  **t2.start()**  **t3.start()**  **t4.start()** |

| **#C**  **from threading import Lock, Thread**  **lock = Lock()**  **g = 0**  **def add\_one():**  **global g**  **lock.acquire()**  **g += 1**  **lock.release()**  **def add\_two():**  **global g**  **lock.acquire()**  **g += 2**  **lock.release()**  **threads = []**  **for func in [add\_one, add\_two, add\_two, add\_one, add\_one, add\_two]:**  **threads.append(Thread(target=func))**  **threads[-1].start()**  **for thread in threads:**  **thread.join()**  **print(g)** |
| --- |

**3. Resolvendo problemas com Sincronização**

1. A seguir é apresentado trecho de código Python. Análise o código e responda as seguintes questões:
2. Explique a finalidade do código apresentado?
3. Qual o resultado após execução do código?
4. Execute o código 10 vezes. Os resultados foram iguais? Caso negativo, por qual motivo?
5. Utilize mecanismos de sincronização de forma que ao final da execução do código conta2 possua saldo 100 e conta1 possua saldo 0.

**import threading**

**import time**

**class ContaBancaria():**

**def \_\_init\_\_(self, nome, saldo):**

**self.nome = nome**

**self.saldo = saldo**

**def \_\_str\_\_(self):**

**return self.nome**

**conta1 = ContaBancaria("conta1", 100)**

**conta2 = ContaBancaria("conta2", 0)**

**class ThreadTransferenciaEntreContas(threading.Thread):**

**def \_\_init\_\_(self, origem, destino, valor):**

**threading.Thread.\_\_init\_\_(self)**

**self.origem = origem**

**self.destino = destino**

**self.valor = valor**

**def run(self):**

**origem\_saldo\_inicial = self.origem.saldo**

**origem\_saldo\_inicial -= self.valor**

**time.sleep(0.001)**

**self.origem.saldo = origem\_saldo\_inicial**

**destino\_saldo\_inicial = self.destino.saldo**

**destino\_saldo\_inicial += self.valor**

**time.sleep(0.001)**

**self.destino.saldo = destino\_saldo\_inicial**

**if \_\_name\_\_ == "\_\_main\_\_":**

**threads = []**

**for i in range(100):**

**threads.append(ThreadTransferenciaEntreContas(conta1, conta2, 1))**

**for thread in threads:**

**thread.start()**

**for thread in threads:**

**thread.join()**

**print('Saldo da', conta1, ':', conta1.saldo)**

**print('Saldo da', conta2, ':', conta2.saldo)**