

Cadeira de Sistemas Operativos

Frequência 1º Semestre

8 de Fevereiro de 2006

Leia atentamente as perguntas seguintes e responda de forma breve e precisa. Pode acompanhar a suas respostas com figuras de forma a torná-las mais explícitas.

1. Estrutura de Sistemas Operativos (4 valores)

Observe a Figura 1 e responda às seguintes perguntas:

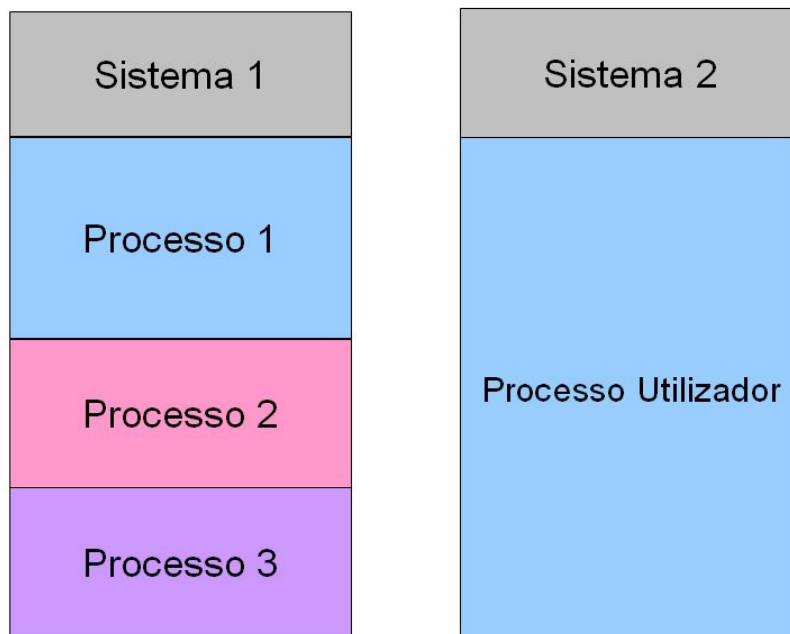


Figura 1

- 1.1 Pelo tipo de implantação memória associadas aos Sistema 1 e Sistema 2, que diferenças fundamentais lhe parecem existir entre eles?
- 1.2 O que se entende por Mutiprogramação e que funcionalidades essenciais são necessárias ao nível do Sistema Operativo para permitir o seu suporte?
- 1.3 Nos Sistemas Operativos avançados existem dois modos de funcionamento distintos, que permitem uma separação clara entre aplicações e sistema. Como se denominam e que características têm?
- 1.4 Para que servem os *System Calls* e qual são os mecanismos básicos que utilizam? Dê dois exemplos de *System Calls* do Sistema Linux indicando resumidamente a sua finalidade.

2. Gestão de Processos (4 valores)

- 2.1 Descreva brevemente a maneira como são criados novos processos num sistema operativo de tipo Unix. Que zonas do processo são partilhadas entre o processo pai e o processo filho?
- 2.2 Num sistema com multiprogramação, vários processos residem em memória mas só um é executado de cada vez. Identifique e descreva os vários estados nos quais os processos podem permanecer quando não estão a ser executados.
- 2.3 Que entende por contexto de um processo? Identifique alguns dos elementos que o constituem.
- 2.4 Indique dois mecanismos distintos que permitem que dois processos comuniquem entre si e quais as suas diferenças fundamentais.

3. Escalonamento de Processos (5 valores)

Suponha que na *Ready Queue* dum sistema multitarefa estão os seguintes processos, para os quais são indicados os tempo de CPU previstos e os instantes de chegada:

Processo P_i	Instante de Chegada	Ciclo CPU previsto
1	0	40
2	20	15
3	40	35
4	60	20

Desenhe o cronograma de execução do conjunto dos quatro processos e calcule o tempo de espera médio no caso do algoritmo de escalonamento ser sucessivamente:

- 3.1 **First Come First Served (FCFS)**.
- 3.2 **Shortest Job First (SJF) com Preempção**.
- 3.3 **Round Robin** com um **Time Quantum** de 20.

4. Sincronização de Processos (4 valores)

Considere dois processos (ou threads) concorrentes que contam o número de carros que entram e saem de um parque de estacionamento, de forma a permitir mostrar o número de lugares livres. O processo A é activado quando entra um carro e o processo B quando sai um carro, podendo haver carros a sair e a entrar simultaneamente.

Os processos podem ser representados pelos seguintes extractos de pseudo-código:

```
global numLugares = NUM-MAX-LUGARES;

contaSaidas ()
{
    while (TRUE) {
        waitfor(saida);
        numLugares = numLugares + 1;
        displayLugares (numLugares);
    }
}

contaEntradas ()
{
    while (TRUE) {
        waitfor(entrada);
        numLugares = numLugares - 1;
        displayLugares (numLugares);
    }
}
```

- 4.1 Identifique a secção crítica existente entre estes dois processos e modifique o código de forma a garantir que os dois processos nunca se encontram na secção crítica ao mesmo tempo.

Imagine que o dispositivo de detecção de entradas e saídas gera eventos (ou sinais) de cada vez que entra ou sai uma viatura, que activam dois outros processos (ou threads) **detectaEntrada()** e **detectaSaida()**.

- 4.2 Utilizando duas variáveis de condição **entrada** e **saída**, escreva o pseudo código desses processo e da função de sincronização **waitfor()** que implementa o mecanismo entre os processos de detecção e contagem.

5. Gestão de Memória (3 valores)

5.1 Que diferença existe entre espaço de endereçamento lógico e físico de um processo? Indique qual o dispositivo que permite criar esta distinção.

A Figura 2 representa o estado da memória física de um sistema com alocação dinâmica de memória em que se pretende carregar o segmento de código de um novo processo P_i , com um tamanho de n_i bytes. Na memória existem 4 partições livres de tamanhos N_0, \dots, N_3 , sendo $N_1 > N_0 > N_2 > n_i > N_3$.

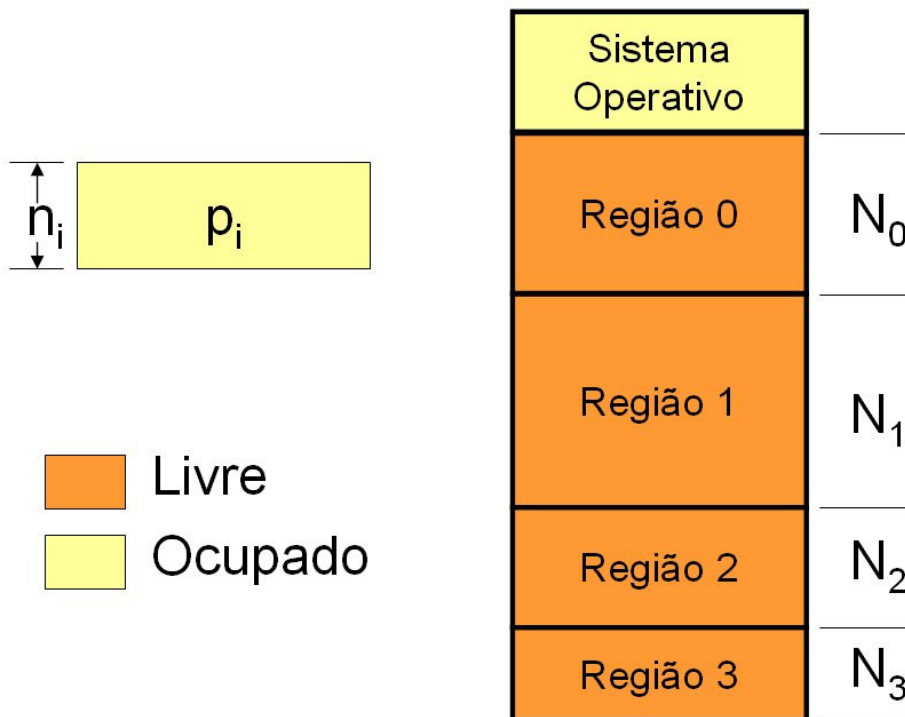


Figura 2

5.2 Indique onde seria carregado o processo P_i nos casos de se utilizarem os seguintes algoritmos: First-Fit, Best-Fit e Worst-Fit.

A Figura 3 representa a correspondência entre o espaço de endereçamento lógico e memória física de uma secção de um processo com 6 páginas de tamanho, num sistema paginado utilizando uma tabela de páginas directa.

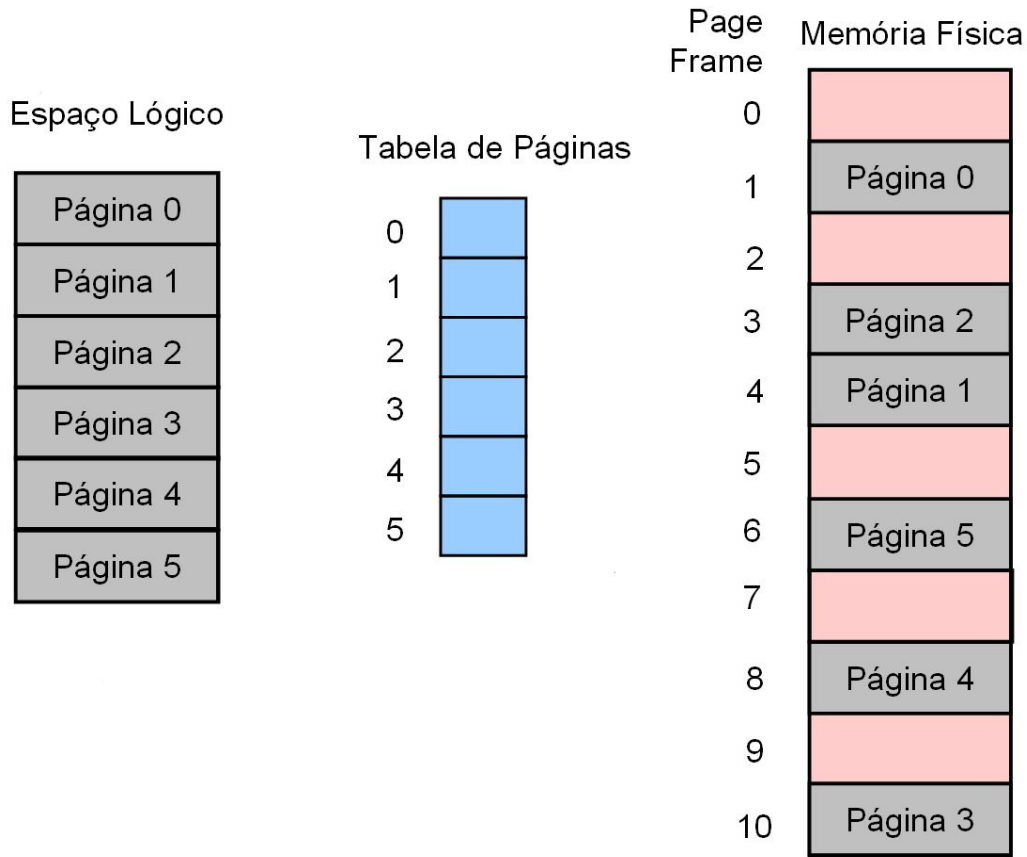


Figura 3

5.3 Indique o conteúdo da tabela de páginas que permite realizar o mapeamento correcto entre os espaços lógicos e físicos representados.