

Gestão de Memória

– Objetivo?

1. Gerir o espaço de endereçamento dos processos
2. Assegurar que cada processo dispõe da memória que precisa
3. Garantir que cada processo só acede à memória a que tem direito (proteção)
4. Otimizar o desempenho dos acessos.

◦ Espaço de Endereçamento

É o conjunto de posições de memória que um processo pode referenciar.

◦ Endereçamento Real

Um endereço real é o endereço indicado no programa e aquele que é acedido na memória principal

– Quais as limitações de um endereçamento real?

1. Limitações devido ao espaço de memória física da RAM (resolução por overlays).
2. Risco de correrem 2 programas um sobre o outro.

– Quais os tipos de partições e fragmentação do endereçamento real?

1. Partições fixas → A fragmentação é interna.
2. Partições variáveis → A fragmentação é externa.

– Descreva e identifique as principais diferenças entre a fragmentação interna e externa.

A Fragmentação interna ocorre em ambientes de partição fixa quando é alocado a um processo mais espaço do que ele necessita, resultando em desperdício de espaço de memória dentro de cada partição.

◦ Endereçamento Virtual

O endereço virtual são endereços indicados no programa convertidos em tempo real de execução, pelo MMU (unidade de gestão de memória do processador)

Caso a palavra referenciada esteja em memória principal, a MMU obtém o seu endereço real e acede à memória. Caso contrário, a MMU avisa o SO para este carregar a palavra em causa.

– Como é constituído o endereço virtual e quais os seus 2 tipos de blocos?

1. Bloco (páginas/segmentos)
2. Deslocamento

◦ Algoritmos de Ordenação de Segmentos

– Em relação aos algoritmos de gestão de memória, quais os tipos de decisões que o sistema operativo tem de tomar em relação à memória principal?

1. Reserva
2. Transferência
3. Substituição

– Quais os algoritmos de reserva de segmentos e qual o seu critério de escolha de blocos livres?

1. best-fit (o menor possível)
2. worst-fit (o maior possível)
3. first-fit (o primeiro possível)
4. next-fit (o primeiro possível a seguir ao anterior)

- **Explique porque é que a fragmentação interna pode ser, ou não ser, um problema quando usamos o algoritmo best-fit para a alocação de memória.**

Sim pode ser um problema porque o algoritmo best-fit gera elevado número de pequenos fragmentos, e como a fragmentação interna ocorre quando um arquivo ou fragmento de arquivo não ocupa completamente o espaço, o que resulta em desperdício de espaço de memória.

- **No que consiste o algoritmo Buddy?**

Quando a memória livre é dividida em blocos de dimensão b^n , b designa-se por buddy.

Um dos buddies é subdividido quantas vezes for preciso para se obter um bloco da dimensão desejada.

Se possível, na libertação um bloco é re combinado com o seu buddy, sendo a associação entre buddies repetida até se obter um bloco com a maior dimensão possível e consegue-se um bom equilíbrio entre o tempo de procura e a fragmentação interna e externa.

- **Quais as 3 situações em que pode ocorrer a transferência de blocos?**

1. A pedido (on request)
2. Por necessidade (on demand)
3. Por antecipação (prefetching).

- **Quais os algoritmos de substituição de páginas?**

1. Ótimo
2. Least Recently Used (LRU)
3. Not Recently Used (NRU)
4. FIFO

- **O Gestor de Processos utiliza três critérios para decidir qual o processo a transferir para o disco:**

1. Estado e prioridade do processo
2. Tempo de permanência na memória principal
3. Dimensão do processo

◦ **Overlay**

1. Consiste na possibilidade de executar programas com dimensão superior à memória principal em sistemas com endereçamento real, dividindo o programa em diversos overlays.
2. Implica que o programador indique quando deve ser carregado um novo overlay em memória.
3. É desvantajoso quando em excesso, pois o processo de troca atrasa o processamento.

◦ **Segmentação**

A segmentação consiste na divisão dos programas em segmentos lógicos que refletem a sua estrutura funcional e na gestão de memória que suporta abstração das linguagens de programação.

O segmento é a unidade de carregamento em memória (eficiência) e proteção.

A dimensão dos segmentos é limitada pela arquitetura e não pode exceder a dimensão da memória principal.

Na arquitetura Segmentada, processos diferentes têm tabelas de segmentos diferentes, o número de segmento e o deslocamento são verificados, comparando-os respetivamente com a dimensão da tabela e com a dimensão do segmento, cada segmento tem associado em código de acesso (leitura, escrita, execução).

– **Vantagens**

1. Adapta-se à estrutura lógica dos programas.
2. Permite a realização de sistemas simples sobre hardware simples.
3. Permite realizar eficientemente as operações que agem sobre segmentos inteiros
4. Não existe fragmentação interna.

– **Desvantagens**

1. O programador tem de ter sempre algum conhecimento dos segmentos subjacentes.
2. Os algoritmos tornam-se bastantes complicados em sistemas mais sofisticados.
3. O tempo de transferência de segmentos entre memória principal e disco torna-se incomportável para segmentos muito grandes.
4. A dimensão máxima dos segmentos é limitada.
5. Existe fragmentação externa. Fragmentação externa ocorre em ambientes de partição variável, quando a memória é desperdiçada devido às lacunas que se desenvolvem na memória entre as partições.

– **Proteção**

1. Processos diferentes têm tabelas de páginas diferentes.
2. O número de segmento e o deslocamento são verificados, comparando-os respetivamente com a dimensão da tabela e com a dimensão do segmento.
3. Cada segmento tem associado um código de acesso (leitura, escrita e execução).

– **Partilha de memória entre processos**

A entidade partilhada é uma divisão lógica do programa pelo que a partilha é natural.

◦ **Paginação**

O espaço de endereçamento virtual tem dimensão superior à da memória principal.

Nos Sistemas Paginados, processos diferentes têm tabelas independentes, o número de página é validado com o tamanho da tabela, cada página tem um código de acesso.

1. A dimensão das páginas é constante o que influencia:
2. A fragmentação interna (depende do padrão de acesso do processo)
3. O número de falta de páginas
4. O tempo de resolução de falta de páginas (limitado pela lentidão do disco)
5. O número de entradas das tabelas de páginas
6. As listas de páginas são mantidas pelo SO

– **Falta de página ou Page Fault**

Quando um programa tenta aceder a uma página que está localizada no espaço de endereçamento virtual, logo está em memória secundária e tem que transferi-la para Memória principal.

– **Vantagens**

1. O programador não tem que se preocupar com a gestão de memória.
2. Os algoritmos de reserva, substituição e transferência são mais simples e eficientes.
3. O tempo de leitura de uma página de disco é razoavelmente pequeno.
4. A dimensão dos programas é virtualmente ilimitada.
5. Não existe fragmentação externa.

– Desvantagens

1. O hardware é mais complexo que o de memória segmentada.
2. Operações sobre segmentos lógicos são mais complexos e menos elegantes, pois têm de ser realizadas sobre um conjunto de páginas e tratamento das faltas de páginas representa uma sobrecarga adicional de processamento.
3. Quanto menor for o tamanho das páginas, maior é o tamanho da tabela de páginas.
4. Influencia a fragmentação interna.

– Proteção

1. Processos diferentes têm tabelas de páginas diferentes
2. É feita a verificação dos tipos de acesso (leitura, escrita e execução).
3. O número de página é validado com o tamanho da tabela.
4. Cada página tem um código de acesso.
5. A granularidade mínima da proteção é a página.

– Partilha de memória entre processos

Tem que ter em conta a granularidade das páginas. Apenas se pode partilhar páginas ou múltiplos de páginas.

– Para que serve uma TLB?

Para manter parte da tabela de páginas em estruturas de hardware, e mantém cópia da parte mais referenciada da tabela de páginas.

– Bits de Estado

P : Presence Bit

R : Reference Bit : 1 = Referenciada; Isto implica que antes de essa página ser recolocada necessita de ser atualizada na memória principal (e depois meter o DB a 0)

M : Dirty Bit : 1 = Página correspondente a esse bit tenha sido Modificada.

Comunicação entre Processos

◦ Memória partilhada

É um espaço de endereçamento partilhado por dois ou mais processos dentro da mesma máquina.

– Primitivas

1. Criar região (nome, tamanho)
2. Associar região (nome)
3. Eliminar região (nome)

- **Vantagens**

1. Mais rápido
2. Mais eficiente.

- **Desvantagens**

1. A sincronização tem de ser explicitamente programada.
2. Programação é complexa. São necessários mecanismos de sincronização adicionais para garantir exclusão mútua sobre a zona partilhada.
3. Memória partilhada só funciona se os processos estiveram dentro da mesma máquina.

- **Objeto do SO** (Objeto de Comunicação do Sistema)

- **Primitivas**

1. Criar canal (nome)
2. Associar canal (nome)
3. Eliminar canal (id canal)
4. Enviar (id canal, mensagem)
5. Receber (id canal, mensagem)

- **Vantagens**

1. A sincronização é implícita e fácil de utilizar.
2. Não são necessários mecanismos de sincronização adicionais porque são implementados pelo núcleo do Sistema Operativo.

- **Desvantagens**

1. Velocidade de transferência limitada pelas duas cópias da informação e pelo uso das chamadas sistema para Enviar e Receber.

- **Modelos de Comunicação**

1. Difusão consiste no envio da mesma informação a um conjunto de processos consumidores.
2. Mestre/escravo, o processo consumidor (escravo) tem a sua ação totalmente controlada por um processo produtor (mestre), a ligação entre a fonte e o destino é fixa e a sincronização entre ambos é estrita.
3. Caixa de correio (canal sem ligação), Transferência assíncrona de informação (mensagens) de vários processos produtores para um canal de comunicação associado, direta ou indiretamente, com um ou mais processos consumidores, os produtores não têm qualquer controlo sobre os consumidores/recetores.
4. Diálogo (canal com ligação), dois processos que pretendam interatuar negociam o estabelecimento de um canal fixo mas temporário de comunicação entre ambos.

- **Características do canal de comunicação.**

1. Nomes;
2. Capacidade de Armazenamento;
3. Sincronização;
4. Segurança;
5. Fiabilidade.
6. Semântica da ligação entre um emissor e o recetor de informação;

- **Identifique e explique três das características de um canal de comunicação.**

1. Nomes que são utilizados pelas partes.
2. Capacidade de armazenamento um canal, que pode ou não ter capacidade para memorizar

várias mensagens e o armazenamento de mensagens num canal de comunicação permite desacoplar os ritmos de produção e consumo de informação

3. Sincronização assíncrona em que o cliente envia o pedido e continua a execução, síncrona em que o cliente fica bloqueado até que o processo servidor leia a mensagem e o cliente/servidor em que o cliente fica bloqueado até que o servidor envie uma mensagem de resposta.

Entradas e Saídas

◦ Entidades

– Indique e descreva as entidades do modelo de entrada e saída.

1. Os periféricos virtuais que são entidades abstratas sobre as quais se realizam todas as operações de E/S.
2. As funções de E/S que são um conjunto reduzido e uniforme de funções necessárias à interação com qualquer periférico.
3. O gestor de periféricos, que é a componente do modelo (processo autónomo ou parte do programa do sistema operativo) que efetua a interação real com o periférico.

◦ Modelo Computacional

O periférico virtual pode ser visto como um canal de comunicação que é explicitamente estabelecido, e sobre o qual são trocadas informações entre o processo utilizador e o periférico.

– Função de E/S

1. Suporte de vários tipos de sincronização associados à operação.
2. Suporte de dois tipos de interface para a transferência de dados.
3. Efetua E/S (id canal, operação, endereço de dados, dimensão, semáforo)

– Sincronização

1. Escrita: Processo cliente fica bloqueado até que os dados tenham sido transferidos para o periférico.
2. Leitura: Processo cliente fica normalmente bloqueado até que os dados pedidos lhe tenham sido transmitidos.

◦ Gestor de Periféricos fora do núcleo (processo independente):

– Vantagens

1. Maior flexibilidade

– Desvantagens

1. Necessidade de privilégios especiais que permitam ultrapassar as proteções habituais
2. Isolamento de espaços de endereçamento entre os gestores e o núcleo do sistema operativo.

◦ Gestor de Periféricos dentro do núcleo (integrado): (Caso mais comum)

– Vantagens

1. Poupa o tempo gasto na comutação entre processos.
2. Maior simplicidade e redução das operações do núcleo do sistema operativo.

- **Desvantagens**
 1. Necessidade de privilégios especiais

- **Objetivos**

- **Descreva quatro dos objetivos do modelo de entradas/saídas de um sistema operativo. Para cada objetivo, sugira as respetivas soluções.**
 1. Uniformização do acesso. **Solução:** Conceito de periférico virtual.
 2. Independência do periférico do SO. **Solução:** Decomposição da interação com os periféricos.
 3. Redireção das E/S. **Solução:** Uniformização entre os mecanismos de comunicação e E/S
 4. Adaptação automática a novos periféricos. **Solução:** Carregamento dinâmica de módulos no núcleo.

Sistema de Ficheiros

Conjunto de ficheiros, directórios, descritores e estruturas de dados auxiliares geridos por um módulo do sistema operativo. Os sistemas de ficheiros permitem estruturar o armazenamento e a recuperação de dados persistentes em um ou mais dispositivos de memória secundária (discos ou bandas magnéticas).

- **Indique as funções e objetivos do sistema de ficheiros.**

- **Ficheiro**

É um conjunto de dados persistentes, geralmente relacionados, identificado por um nome. Composto por: nome (identifica o ficheiro perante o utilizador), descritor de ficheiro (estrutura de dados em memória secundária com informações sobre o ficheiro tais como a dimensão, a data de criação e de modificação, dono e autorizações de acesso), informação (dados guardados em memória secundária)

- **Diretório**

Um diretório é um catálogo de nomes de ficheiros (ou diretórios) que estabelece a associação entre os nomes e os seus descritores. Um diretório pode conter os descritores ou apenas os seus identificadores e composto por um nome, um descritor, e informação.

- **O tipo de um ficheiro depende do seu conteúdo e da sua forma de acesso. Indique que formas de acesso existem e explique sucintamente como funcionam.**
 1. Acesso sequencial que para se ler o registo N é necessário ler os N-1 registos anteriores e para alterar um registo é preciso ler o ficheiro todo e escrevê-lo de novo com o registo alterado.
 2. Acesso direto em que se pode aceder diretamente a um registo sem aceder aos anteriores e não se pode inserir um novo registo entre outros dois.
 3. Acesso por chave em que os registos são identificados por chaves alfanuméricas reconhecidas pelo sistema de ficheiros.

- **Proteção (entender como é feita a proteção de ficheiros)**

1. Estabelecimento de uma proteção quando o ficheiro é criado:
 - a. É guardada no descritor do ficheiro e é definida pelo utilizador responsável pela sua criação (o dono).

- b. O dono de um ficheiro pode ser alterado durante o tempo de vida de um ficheiro.
- 2. Especificação dos direitos de acesso a um ficheiro:
 - a. Listas de Controlo de Acesso para múltiplos utilizadores.
 - b. Lista de Controlo de Acesso simplificada – definição de grupos de utilizadores e de direitos de acesso a um ficheiro para o dono, para um grupo de utilizadores e para os demais utilizadores (UNIX).

◦ **Discos magnéticos (como otimizar o seu acesso)**

– **Indique e explique sucintamente as formas de otimização dos Acessos a um disco.**

- 1. Minimização da frequência de acesso, através de uma cache de blocos em memória principal
- 2. Minimização do tempo de posicionamento, ordenando os pedidos segundo a ordem mais favorável
- 3. Minimização do tempo de latência, minimizando o tempo de acesso a múltiplos sectores por pista fator de entrelaçamento e leitura de pistas inteiras para uma cache local ao controlador.

– **Aumentar o desempenho consulta de ficheiros**

Três etapas:

- 1. Abertura do ficheiro dado o nome
 - a. É pesquisado o diretório, copiado o FD para memória e é devolvido ao utilizador o identificador de ficheiro aberto.
 - b. Verificar os direitos de acesso do processo ao ficheiro em causa.
- 2. Leitura ou escrita de informação dado o identificador de ficheiro aberto
 - a. Este identificador permite chegar rapidamente ao seu descritor em memória.
- 3. Fecho do ficheiro.