

## **Gestão da Memória**

**1 Descreva os diferentes níveis em que se estrutura a memória de um sistema computacional, caracterizando- os em termos de capacidade, tempo de acesso e custo. Explique, face a isso, que funções são atribuídas a cada nível.**

cache vai de KBs a poucos MBs, é rápida cara e volátil. Tem como função conter a cópia das posições de memória (instruções e operandos) mais frequentemente utilizados pelo processador num passado próximo.

Memória principal vai de centenas de MBs a alguns GBs, tem velocidade e preços médios e é volátil. Tem como função o armazenamento do espaço de endereçamento dos vários processos. Para garantir uma elevada taxa de utilização do processador este nº de espaços de endereçamento deve ser elevado.

Memória de massa tem capacidade de centenas de GBs, é lenta, barata e não volátil. Tem como função fornecer um sistema de ficheiros onde são salvaguardados mais ou menos permanentemente os dados e código. Tb pode ser usada como área de swapping, extensão da memória principal para que haja um maior nº de processos a coexistir num determinado instante.

**2. Qual é o princípio que está subjacente à organização hierárquica de memória? Dê razões que mostrem porque é que a aplicação de tal princípio faz sentido.**

princípio da localidade de referência; trata-se de uma constatação heurística sobre o comportamento de um programa em execução que estabelece que as referências à memória durante a execução de um programa tendem a concentrar-se em fracções bem definidas do seu espaço de endereçamento durante intervalos mais ou menos longos;

- este tipo de organização baseia-se no pressuposto que quanto mais afastado uma instrução, ou um operando, está do processador, menos vezes será referenciado; nestas condições, o tempo médio de uma referência aproxima-se tendencialmente do valor mais baixo;

**3. Assumindo que o papel desempenhado pela gestão de memória num ambiente de multiprogramação se centra, sobretudo, no controlo da transferência de dados entre a memória principal e a memória de massa, indique quais são as actividades principais que têm que ser consideradas.**

transferência para a área de swapping do total ou parte do espaço de endereçamento de um processo quando o espaço em memória principal não é suficiente para conter todos os processos que coexistem.

- salvaguarda de um registo que indique as posições livres e ocupadas na memória principal.

- alocação de porções de memória principal para os processos que dela vão precisar, ou libertação qd não necessária.

**4. Porque é que a imagem binária do espaço de endereçamento de um processo é necessariamente relocatável num ambiente de multiprogramação?**

Uma vez que a situação de termos em memória principal processos bloqueados e na área de swapping processos prontos a executar, leva a uma menor eficiência na utilização do processador e a um desperdício do espaço de memória principal. Faz sentido, se a memória principal não tiver espaço livre, transferir os processos bloqueados em memória principal para a área de swapping e transferir os processos prontos a executar na área de swapping para a memória principal. Por outro lado um processo pode precisar de muita memória para armazenar as suas variáveis e esgotar o seu espaço de endereçamento. Neste caso é necessário alocar mais memória e associá-la ao processo que a requisita.

**5. Distinga linkagem estática de linkagem dinâmica. Qual é a mais exigente? Justifique a sua resposta.**

Na linkagem estática todos os ficheiros objecto e bibliotecas do sistema são juntas num ficheiro único. Temos assim que todas as referências existentes no programa existem no espaço de endereçamento desde o início.

Na linkagem dinâmica as referências a funções de bibliotecas do sistema são substituídas por stubs, que determina a localização da função de sistema em memória principal ou a carrega em memória principal. Sendo em seguida a referência ao stub substituída pela referência da função, ou seja, em linkagem dinâmica não temos todas as referências satisfeitas ao início, elas vão-se satisfazendo ao longo do decorrer do programa, quando necessário.

A mais exigente é a linkagem estática, pois temos um maior espaço de endereçamento é quando há mudança de contexto e o processo passa ao estado blocked ou suspended-blocked em que vai haver transferência do espaço de endereçamento, como o espaço é grande e a memória de massa é lenta, demora-se um pouco.

A linkagem estática é mais existente em termos de quantidade de memória para armazenar os programas mas o código é sequencial logo exige menos saltos entre posições e consequentemente reduz o acesso ao disco para resolver dependências. Assim sendo a dinâmica tornaria-se mais exigente.

**6. Assuma que um conjunto de processos cooperam entre si partilhando dados residentes numa região de memória, comum aos diferentes espaços de endereçamento. Responda justificadamente às questões seguintes – em que região do espaço de endereçamento dos processos vai ser definida a área partilhada? – será que o endereço lógico do início da área partilhada é necessariamente o mesmo em todos os processos? – que tipo de estrutura de dados em Linguagem C tem que ser usada para possibilitar o acesso às diferentes variáveis da área partilhada?**

Uma vez que se trata de memória partilhada, temos que assumir que os dados partilhados podem tomar diferente tamanho ao longo da execução do programa. Como tal esses dados encontram-se na zona de definição dinâmica, região global;

- Não, pois o espaço ocupado pelo código e pela zona de definição estática pode ser diferente, o que resulta num endereço lógico inicial da zona de definição dinâmica diferente para os diversos processos. O tamanho das duas primeiras zonas (código e definição estática) não é fixo, é definido pelo loader.

- Ponteiros. Na execução dinâmica o conteúdo do ponteiro é o endereço físico da variável para a qual aponta. Este endereço, e não o endereço lógico, é válido para todos os processos. Deste modo, o acesso às variáveis partilhadas deve ser feito usando ponteiros.

**7. Distinga relativamente a um processo espaço de endereçamento lógico de espaço de endereçamento físico. Que problemas têm que ser resolvidos para garantir que a gestão de memória num ambiente de multiprogramação é eficiente e segura?**

Endereçamento físico refere-se à localização na memória principal do espaço de endereçamento do processo.

O endereçamento lógico refere-se à imagem binária do espaço de endereçamento do processo e é um endereçamento relativo relativo a um dado endereço, offset. Problemas que têm de ser resolvidos:

- conversão em runtime de endereçamento lógico para endereçamento físico;
- garantir que o endereçamento físico obtido na conversão anterior está dentro da gama de endereços do espaço de endereçamento do processo.

**8. Caracterize a organização de memória designada de memória real. Quais são as consequências decorrentes deste tipo de organização?**

(memória real) Neste tipo de organização existe uma correspondência biunívoca entre o espaço de endereçamento lógico e o espaço de endereçamento físico de um processo. Isto tem como consequências:

- limitação do espaço de endereçamento de um processo (não é possível em nenhum caso que o espaço de endereçamento de um processo seja maior que o tamanho da memória principal;
- contiguidade do espaço de endereçamento físico (é mais simples e eficiente supor que o espaço de endereçamento é contíguo);
- área de swapping (o seu papel é assumir de arrecadação do espaço de endereçamento de todos os processos que não possam ser carregados em memória principal por falta de espaço.

**9. Descreva detalhadamente o mecanismo de tradução de um endereço lógico num endereço físico numa organização de memória real.**

```
if(end_logico>limite)
```

```
    gera_excecao(); //erro
```

```
else
```

```
    end_fisico=end_logico+end_base;
```

- ocorre comutação de contexto

- registo base e limite são preenchidos de acordo com a entrada correspondente ao novo processo

- endereço base: endereço correspondente ao início do espaço de endereçamento físico do processo

- endereço limite: tamanho do espaço de endereçamento do processo

**10. O que é que distingue a arquitectura de partições fixas da arquitectura de partições variáveis numa organização de memória real? Indique quais são as vantagens e desvantagens de cada uma delas.**

Na arquitectura de partições fixas a memória principal é dividida num conjunto fixo de partições mutuamente exclusivas, mas não necessariamente iguais. Cada uma contém o espaço de endereçamento de um processo. Tem como vantagens o facto de ser simples de implementar (HW e estruturas de dados simples) e é eficiente (selecção é feita rapidamente). Tem como desvantagens a grande fragmentação interna e serve só para aplicações específicas.

Na arquitectura de partições variáveis toda a parte disponível de memória constitui à partida um bloco. Reserva-se sucessivamente regiões de tamanho suficiente para carregar o espaço de endereçamento dos processos que vão surgindo e liberta-se quando não é necessário. Para gerir a memória precisamos de 2 listas, lista das regiões ocupadas e das regiões livres. Para não existir risco de existirem regiões livres tão pequenas que não possam ser utilizadas e que tornem a pesquisa mais ineficiente, tipicamente a memória principal é dividida em blocos de tamanho fixo e a reserva é feita em entidades do tamanho desses blocos. Tem como vantagens uma menor fragmentação da memória, pois cada processo ocupa o espaço estritamente necessário. Tem como desvantagens o facto de uma pesquisa ser mais elaborada e ineficiente.

Nota: garbage collection – compactação do espaço livre agrupando as regiões num dos extremos da memória, exige a paragem de todo o processamento

**11. A organização de memória real conduz a dois tipos distintos de fragmentação da memória principal.**

**Caracterize-os e indique a que tipo de arquitectura específica cada um está ligado.**

Os 2 tipos de fragmentação da memória principal são partições fixas e partições variáveis.

Nas partições fixas a fragmentação é interna e a parte da partição interna que não contém espaço de endereçamento de processos é desperdiçada, não sendo usada para nada.

Nas partições variáveis a fragmentação é externa, as sucessivas reservas e libertações de espaço resultam em fragmentos de espaço livre tão pequenos que não podem ser utilizados. Esta fracção de memória principal desperdiçada pode em alguns casos atingir 1/3 do total de memória.

**12. Entre os métodos mais comuns usados para reservar espaço em memória principal numa arquitectura de partições variáveis, destacam-se o next fit e o best fit. Compare o desempenho destes métodos em termos do grau e do tipo de fragmentação produzidos e da eficiência na reserva e libertação de espaço.**

O next fit consiste em iniciar a pesquisa a partir do ponto de paragem da pesquisa anterior. É muito rápido a pesquisar e pode causar fragmentação interna pois se o espaço reservado for maior que o espaço de end do processo existe espaço desperdiçado. Mas este espaço desperdiçado faz com que na libertação do espaço reservado não existam fragmentos muito pequenos.

O best fit escolhe a região mais pequena disponível onde cabe todo o espaço de end. do processo. É mais lento a pesquisar e não há fragmentação interna, apenas externa.

Este último método resulta em mais memória desperdiçada, visto que tende a deixar fragmentos de memória livre demasiado pequenos para serem utilizados.

**13. Caracterize a organização de memória designada de memória virtual. Quais são as consequências decorrentes deste tipo de organização?**

Numa organização de memória virtual o espaço de endereçamento lógico e físico de um processo estão completamente dissociados. As consequências são as seguintes:

- pode ser estabelecido uma metodologia conducente à execução de processos em que o seu espaço de endereçamento pode ser maior que a memória principal;
- não contiguidade do espaço de endereçamento físico (o espaço de endereçamento de um processo dividido em blocos de tamanho fixo estão dispersos por toda a memória, procurando-se desta maneira obter uma ocupação mais eficiente do espaço disponível);
- área de swapping (é criada nesta uma imagem atualizada de todo o espaço de endereçamento dos processos que coexistem correntemente, nomeadamente da sua parte variável).

**14. Indique as características principais de uma organização de memória virtual. Explique porque é que ela é vantajosa relativamente a uma organização de memória real no que respeita ao número de processos que correntemente coexistem e a uma melhor ocupação do espaço em memória principal.**

- não existe limitação para o espaço de endereçamento do processo (EEP) uma vez que a área de swapping funciona como extensão da memória principal, assim os processos que tenham parte do seu espaço de endereçamento na área de swapping, ou mesmo todo lá, não estão obrigatoriamente bloqueados, existindo assim um maior número de processos na memória principal;
- a parte mais inactiva de um EEP pode estar na área de swapping, deixando espaço na memória principal para outros processos.

Como o espaço de endereçamento físico de um processo não é necessariamente contíguo, não existem problemas de fragmentação.

**15. O que é que distingue a arquitectura paginada da arquitectura segmentada / paginada numa organização de memória virtual? Indique quais são as vantagens e desvantagens de cada uma delas.**

O que distingue uma arquitectura paginada da arquitectura segmentada/paginada é que na arquitectura seg/pag não se faz uma divisão do espaço de endereçamento lógico do processo (EEL) às cegas, pois 1º faz-se uma divisão do EEL em segmentos e depois esses segmentos são divididos em paginas.

Vantagens da arquitectura paginada:

- não conduz a fragmentação externa e a interna é desprezável (tem-se grande aproveitamento da memória principal);
- é geral, isto é, é independente do tipo dos processos que vão ser executados (nº e tamanho do seu EE);
- não exige requisitos especiais de hardware, pois a unidade de gestão de memória dos processadores actuais de uso geral já está preparado para a sua implementação.

Desvantagens da arquitectura paginada:

- acesso à memória mais longo;
- operacionalidade muito exigente, pois a sua implementação exige por parte do SO a existência de um conjunto de operações de apoio e que são complexas e que têm de ser cuidadosamente estabelecidas para que não haja muita perda de eficiência).

Vantagens da arquitectura segmentada/paginada:

- iguais às 2 primeiras vantagens da arquitectura paginada;
- gestão mais eficiente da memória no que respeita às zonas de crescimento dinâmico
- minimização do nº de paginas que têm de estar residentes em memória em cada etapa de execução do processo.

Desvantagens da arquitectura segmentada/paginada:

- iguais a todas as desvantagens da arquitectura paginada;
- exige requisitos especiais de HW.

**16. Descreva detalhadamente o mecanismo de tradução de um endereço lógico num endereço físico numa organização de memória virtual paginada.**

O processador contém dois registros, o npag (número de página) e o desloc (que nos dá o deslocamento dentro dessa página). O número de página é enviado à unidade de gestão de memória que o vai somar ao endereço do registro base. O resultado é enviado para a memória principal, que vai à tabela de paginação, que vai informar a UGM, através do bit M/AS, se a página está ou não em memória principal, através do registro base, qual o número de frame, e informar a ocupação ou não desta entrada (a não ocupação significa que ainda não foi reservado espaço na área de swapping para esta página), pelo bit O/L. O registro base irá fornecer o número de frame e do processador vem o deslocamento, resultando no endereço físico, na memória principal. Há duas hipóteses de erro, uma caso não exista a página, e outra caso não exista o endereço. O primeiro é dado por subtrair AS ao bit M/AS, e o segundo é dado subtraindo L a O/L. Caso o bloco não esteja em memória, a UGM gera uma exceção por falta de bloco e a rotina de serviço à exceção deve por em marcha ações que visem a transferência desse bloco da área de swapping para a memória principal, e após a sua conclusão a repetição da execução da instrução que produziu a referência. Todas estas operações são realizadas de uma forma completamente transparente ao utilizador que não tem consciência das interrupções introduzidas na execução do processo.

**17. Explique porque é que hoje em dia o sistema de operação dos computadores pessoais supõe, quase invariavelmente, uma organização de memória de tipo memória virtual.**

A implementação de uma organização de memória do tipo memória virtual não impõe limites ao tamanho do espaço de endereçamento do processo (EEP) e não dá quaisquer problemas em termos de fragmentação. Embora a sua implementação em HW seja mais complexa, o tempo de alocação e libertação de segmentos de memória seja maior, no global este tipo de implementação é mais vantajoso.

**18. Porque é que a área de swapping desempenha papéis diferentes nas organizações de memória real e de memória virtual?**

Porque ao contrário da organização de memória real, em que temos uma correspondência biunívoca entre espaço de end. lógico e espaço de end. físico, na organização de memória virtual estes 2 estão completamente dissociados, podendo assim um processo não estar bloqueado e ter parte do seu espaço de end. na área swapping.

Enquanto na organização de memória real o espaço ou está totalmente na área swapping ou na memória principal, por causa da correspondência biunívoca. Logo na organização de memória virtual convém ter uma imagem dos EEP que coexistem correntemente, para que qd seja necessária uma parte do EEP residente na área de swapping ela seja transferida para a memória principal pelo SO.

**19. Considere uma organização de memória virtual implementando uma arquitectura segmentada /paginada. Explique para que servem as tabelas de segmentação e de paginação do processo. Quantas existem de cada tipo? Descreva detalhadamente o conteúdo das entradas correspondentes.**

Numa arquitectura segmentada/paginada o espaço de end. lógico do processo é dividido em segmentos, segmentos estes que são divididos em páginas às quais corresponde um frame na memória principal. Logo temos uma tabela de segmentos e uma tabela de páginas, por processo.

Conteúdo da entrada da tabela de segmentos: endereço da tabela de paginação do segmento, bits de controlo.

Conteúdo da entrada da tabela de páginas: nº do frame em memória (as frames têm tamanho fixo), nº do bloco na área de swapping (memória de massa é dividida em blocos), informação de controlo.

**20. Qual é a diferença principal existente entre a divisão do espaço de endereçamento de um processo em páginas e segmentos? Porque é que uma arquitectura segmentada pura tem pouco interesse prático?**

A diferença reside no facto de que a divisão em segmentos não é uma divisão às cegas, pois tem em conta a estrutura modular na implementação de um programa. Ao contrário da paginada, que é feita às cegas, só coloca no início de uma nova página zonas funcionalmente distintas do espaço de end. d um process.

A arquitectura segmentada pura tem pouco interesse, pois ao tratar a memória principal como um espaço contínuo, leva ao uso de técnicas de reserva de espaço usadas na arquitectura de partições variáveis, o que origina grande fragmentação externa e desperdício de espaço. Os segmentos de crescimento contínuos tb dão problemas, pois os acréscimos podem não caber na localização actual, levando à sua transferência total para outra região de memória ou se não houver espaço em memória principal é bloqueado o processo e o seu espaço de end. ou o segmento que não cabe são transferidos para a área swapping.

**21. As bibliotecas de rotinas linkadas dinamicamente (DLLs) são ligadas ao espaço de endereçamento de diferentes processos em run time. Neste contexto, responda justificadamente às questões seguintes – que tipo de código tem que ser gerado pelo compilador para que esta ligação seja possível? – este mecanismo de ligação pode ser utilizado indiferentemente em organizações de memória real e de memória virtual?**

. – (compilador) Tem de ser gerado código de localização na memória principal das rotinas necessárias, ou que promova a sua carga em memória principal (stubs).

- Pode, pois quer numa ou noutra organização de memória o espaço de endereçamento da rotina linkada dinamicamente terá de estar ou ser carregada na memória principal, logo podendo haver correspondência biunívoca da organização real.

**22. Quer numa arquitectura paginada, quer numa arquitectura segmentada / paginada, a memória principal é vista operacionalmente como dividida em frames, onde pode ser armazenado o conteúdo de uma página de um processo. Porque é que é conveniente impor que o tamanho de cada frame seja uma potência de dois?**

Primeiro porque a memória principal tem um tamanho que é uma potência de 2, logo todos os frames têm o mesmo tamanho.

Segundo, pois sendo uma potência de 2, parte dos bits, por exemplo os mais significativos indicam-nos o nº do frame e os menos significativos a posição dentro do frame. O que simplifica a unidade de gestão de memória em termos de HW e SW.

**23. Foi referido que nem todos os frames de memória principal estão disponíveis para substituição. Alguns estão locked. Incluem-se neste grupo aqueles que contêm as páginas do kernel do sistema de operação, do buffer cache do sistema de ficheiros e de um ficheiro mapeado em memória. Procure e aduzir razões que justifiquem esta decisão para cada um dos casos mencionados.**

Locked significa que não estão disponíveis para substituição.

Como é lógico as páginas do kernel têm de estar locked, pois se não estivessem podiam ser substituídas (transferidas para a área swapping), logo se fossem precisas tinham de ser novamente transferidas para a memória principal e até talvez substituir uma página de outro processo, o que fazia perder algum tempo, podendo bloquear todo o sistema durante esse tempo.

Buffer cache e ficheiro mapeado em memória são ambos as técnicas usadas para uma maior rapidez de acesso, logo a sua substituição implicaria uma redução do tempo de acesso, o que vai contra o princípio com que eles foram implementados.

**24. O que é o princípio da optimalidade? Qual é a sua importância no estabelecimento de algoritmos de substituição de páginas em memória principal?**

O princípio da optimalidade diz-nos que o frame que deve ser substituído é aquele que não vai ser mais usado, referenciado, ou se o for se-lo-á o mais tarde possível.

A sua importância é que ele é o princípio ótimo de substituição, mas é não casual, pois tem de se conhecer o futuro para o usar. Logo tenta-se a partir do conhecimento passado prever o futuro para saber qual o frame que não vai ser usado ou vai ser referenciado tardiamente

**25. O que é o working set de um processo? Conceba um método realizável que permita a sua determinação.**

Carrega-se a 1ª e a última página do espaço de endereçamento do processo (correspondete ao início do código e ao topo da stack). Ao longo da execução do processo serão gerados pages-fault, sendo estes carregados, mas ao longo do tempo o número de pages-fault diminui até ser zero (devido ao princípio da localidade de referência). Aí temos o chamado working set do processo.

**26. Dê razões que expliquem porque é que o algoritmo do relógio, numa qualquer das suas variantes, é tão popular. Descreva uma variante do algoritmo do relógio em que são consideradas as quatro classes de frames características do algoritmo NRU.**

(algoritmo do relógio) É popular visto que as operações de fifo\_in e fifo\_out tornam-se num incremento de um ponteiro (módulo e nº de elementos da lista).

```
while (! frame_adequado)
{
    if(Ref==0)
        substituicao(pag_associada);
    else
        Ref=0;
        ponteiro ++ % nº elementos
}
```

**27. Como distingue a estratégia demand paging da prepaging? Em que contexto é que elas são aplicadas?**

Quando um processo é introduzido na fila dos processos Ready-to-run, mais tarde, em resultado de uma suspensão, é necessário decidir que páginas se vão carregar na memória principal.

Se não se carregar nenhuma página e se se esperar pelo mecanismo de page-fault para formar o working-set do processo temos demand paging.

Se se carregar a 1ª e a última página do processo da 1ª vez que passa a Ready-to-run ou as páginas residentes no momento da suspensão temos prepaging.

**28. Assuma que a política de substituição de páginas numa dada organização de memória virtual é de âmbito global. Explique detalhadamente como procederia para detectar a ocorrência de thrashing e como procederia para a resolver.**

Para detectar, o working set dos processos teria de que ser maior que o nº de frames unlocked disponíveis na memória principal. A solução seria ir suspendendo processos até que o problema, geração de page-fault, thrashing desapareça.