

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.
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.
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.
4. Porque é que a imagem binária do espaço de endereçamento de um processo é necessariamente relocável num ambiente de multiprogramação?
5. Distinga *linkagem estática* de *linkagem dinâmica*. Qual é a mais exigente? Justifique a sua resposta.
6. O ficheiro executável não contém normalmente todo o código necessário à execução de um programa. Parte dele, normalmente as bibliotecas de sistema, só são carregadas mais tarde pelo *loader* ou *run time*, conforme se trate respectivamente de mecanismos de *linkagem* estática ou dinâmica. O que se procura minimizar em cada um destes casos?
7. Explique detalhadamente como se processa em *run time* o mecanismo de *linkagem* dinâmica.
8. 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?
9. 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?
10. Caracterize a organização de memória designada de *memória real*. Quais são as consequências decorrentes deste tipo de organização?
11. Descreva detalhadamente o mecanismo de tradução de um *endereço lógico* num *endereço físico* numa organização de memória real.
12. O que é que distingue a *arquitetura de partições fixas* da *arquitetura de partições variáveis* numa organização de memória real? Indique quais são as vantagens e desvantagens de cada uma delas.
13. 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.
14. 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.
15. Caracterize a organização de memória designada de *memória virtual*. Quais são as consequências decorrentes deste tipo de organização?

16. 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.
17. Descreva detalhadamente o mecanismo de tradução de um *endereço lógico* num *endereço físico* numa organização de memória virtual paginada.
18. 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*.
19. Porque é que a *área de swapping* desempenha papeis diferentes nas organizações de *memória real* e de *memória virtual*?
20. 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*?