

Capítulo 1- Introdução

Uma das principais funções de um SO é apresentar aos programas (aplicações) e programadores uma interface virtual baseada em abstrações, dita máquina estendida. Explique porquê. (1.1.1)

O SO providencia aos programas e programadores um modelo/interface simples e intuitivo do computador, escondendo os detalhes da complexidade

Esta função de um SO é a de facilitação da utilização. As referidas abstrações são mais fáceis de utilizar do que lidar com a própria máquina.

A função “maquina estendida” esconde, do utilizador, a realidade sobre o HW e mostra uma interface amigável que o programador pode usar, mesmo que não perceba nada sobre os detalhes do HW de computadores.

O SO serve os programas ou os programadores através das chamadas ao sistema que disponibiliza.

Abstrações:

Gestão Processos – Cada processo é um programa, abstração contém dados necessários à execução do programa.

Gestão I/O – uma das principais funções que é esconder a complexidade do HW, contém dados referentes aos drivers dos componentes.

Gestão Memória Principal e Secundária – o que mantém em memória principal e o que liberta para a memória secundária e vice-versa.

Uma das principais funções de um SO é a gestão de recursos de um sistema complexo constituído por muitas partes. Explique porquê. Dê exemplos de gestão espacial e de gestão temporal. (1.1.2)

Porque o HW tem recursos limitados enquanto que o numero de processos em execução pode ser arbitrariamente grande.

Exemplo Gestão Espacial -> memória RAM

Exemplo Gestão Temporal -> CPU

Indique e descreva resumidamente quais são as duas funções principais de um SO. (1.1.)

SO como gestão de recursos.

Gestão espacial como memória RAM; gestão temporal como a CPU.

SO como máquina estendida por intermédio de abstrações simples e intuitivas de operar.

F1.1 Em que consiste o tratamento por lotes (batch systems) ?

O tratamento por lotes (batch systems); foi criado em meados da década de 50 e o seu objetivo era de reduzir custos no processamento dos computadores, pois os mesmos levam muito tempo a processar simples tarefas.

Para isso o Batch Systems gravava várias tarefas em fitas magnéticas que depois de gravadas eram lidas por um programa especial, lendo uma por uma e executando-as, gerando uma segunda fita com as tarefas do lote já executadas.

Esta segunda fita era levada para outro equipamento que por sua vez iria imprimir em modo off-line as tarefas desse lote já executado.

F1.2 Indique um dispositivo que contenha um sistema operativo do tipo embebido

Um exemplo poderá ser o de um leitor de DVD's. Nestes casos o sistema operativo está armazenado numa memória ROM ou EEPROM e não é feito para permitir a execução de aplicações instaladas pelos utilizadores.

Em que consiste uma função de sistema ? Explique sucintamente como é implementada a chamada a uma função de sistema. (1.3.1)

É uma abstração executada pelo SO no modo núcleo.

Como uma função disponibilizada pelo SO aos utilizadores/aplicações para cesso a serviços e recursos controlados pelo SO.

É implementada através de instrução TRAP da CPU para mudar para modo núcleo.

No núcleo é encaminhado o pedido para o handler da função respetiva, depois de executada, retorna ao modo de utilizador e ao processo.

Conceito sistema operativo tempo real e variantes (1.4.8)

Caracterizados por terem tempo como parâmetro fundamental.

No caso das acções a realizarem terem necessariamente que ocorrer em determinado momento, estamos perante um SO tempo real critico. (Ex: linha montagem auto)

Por outro lado, temos SO tempo real não critico, que apesar de não ser desejável, a ocorrência ocasional de falhas, não causa danos permanentes.

F1.4 O que entende por multiprogramação? Relacione este conceito com o conceito de processo. (1.5.1)

Capacidade de comutar execução de programas.

Multiprogramação consiste em vários programas em simultâneo carregados na memória fazendo com que o tempo do processador seja dividido, ou seja, quando um programa for bloqueado pela CPU outro poderá ser executado aumentando assim a sua utilização.

Um processo é a abstração que permite implementar a suspensão/retoma da execução de um programa.

Relacione os conceitos de troca de contexto (context switch) e de multiprogramação.. (1.5.1)

A troca de contexto é um procedimento da CPU para mudar de um processo para outro, enquanto assegura a inexistência de conflitos.

Efetivamente esse conceito é critico pra um sistema que pretenda implementar uma versão “amigável” de multiprogramação.

Em UNIX, que relação têm os ficheiros ditos especiais com as operações de entrada/saída (I/O) e quais são as consequências dessa relação a nível de programação ? (1.5.3)

Os ficheiros ditos especiais são abstrações que representam dispositivos de I/O que podem ser lidos ou escritos usando as mesmas funções utilizadas para leitura e escrita de ficheiros.

Explique em que consiste um interpretador de comandos (shell) e o seu modo de operação. Este é considerado como pertencente ao sistema operativo ? (1.5.6)

Um interpretador de comandos é interativo.

Recebe um comando do utilizador, executa e espera novo comando.

Não pertence ao SO!

F1.5 Para um processador, explique a diferença entre modo utilizador e modo núcleo ou supervisor.

O modo utilizador apenas disponibiliza um subconjunto de instruções da CPU. É neste modo que funcionam as aplicações.

O modo núcleo é um modo privilegiado do processador, para o qual todas as instruções estão disponíveis. Só o SO tem acesso a este modo.

Por exemplo as chamadas ao sistema são executadas em modo núcleo, a gestão é feita pelo SO, o utilizador não tem acesso por estarem aqui instruções/funções críticas cuja utilização errada poderia colocar em causa o funcionamento do sistema.

Caracterize um sistema operativo baseado numa estrutura de micronúcleo (microkernel). (1.7.3)

SO com estrutura modular onde apenas um dos módulos, o micronúcleo, executa em modo núcleo e os restantes em modo utilizador, como por exemplo os gestores de dispositivos e os sistemas de ficheiros de modo a conseguir um elevado nível de fiabilidade.

TRAP = é uma instrução que permite passar do modo utilizador para o modo nucleo.

MUTEX= serve para guardar uma região critica (entrada lock, saida unlock) quando existem varias tarefas de modo a garantir que apenas uma está em cada momento na região critica.

Não são usados para fazer exclusão mutua entre processos, apenas no mesmo processo entre as tarefas.

Capítulo 2 – Processos, Tarefas e Impasses

F2.2 Ao longo dos anos, os projectistas de sistemas operativos desenvolveram o conceito de processo, uma abstracção de um programa em execução. Caracterize o conceito de processo. (2.1)

Um processo é um programa em execução.

Associado ao processo temos o seu espaço de endereçamento, lista de posições memória que o processo pode escrever.

Também associado a um processo está um conjunto de recursos, lista arquivos abertos, alarmes pendentes, lista de processos relacionados e todas as demais informações necessárias à execução do processo.

Justifique porque não se deve programar com suposições baseadas no tempo de execução do código. (2.1.1)

Devido ao escalonamento dos processos ser aleatório, não se pode prever nem reproduzir as condições temporais associadas à execução de um processo.

F2.5 Explique o conceito de pseudoparalelismo (ou pseudoconcorrência). (2.1.1)

Mesmo quando na presença de um único processador, é possível que vários processos ativos em simultâneo.

A CPU alterna de um programa para outro, executando cada um por uma fração de tempo, mas em cada instante só um pode estar a usar o processador.

A esta ilusão de vários processos a correr em paralelo dá-se o nome de pseudoparalelismo. Esta ilusão ocorre quando a escala de tempo é muito superior à fração de tempo que cada processo corre de cada vez.

O sequenciador (scheduler) é responsável pela escolha de qual processo ocorre a cada momento.

O processo é a abstração que permite suspender e retomar a execução dos programas.

Explique porque razão em UNIX a criação de um subprocesso diferente do processo pai é uma operação efectuada em dois passos. (2.1.2)

Dois passos para permitir o redirecccionamento da entrada, saída e erro padrão.

O que entende por uma hierarquia de processos ?(2.1.4)

Estrutura em árvore formada quando processos criam outros processos. Esta existe em alguns SO, como o Linux, em que o processo pai permanece associado ao processo filho.

F2.1 Considere um processo que se encontra no estado bloqueado. A partir deste estado, para que outro estado pode o processo transitar ? (2.1.5)

Bloqueado passará para executável!

Estados:

Em Execução – Processo atribuido a um CPU e a ser executado

Executável – Processo Pronto a ser executado, aguarda atribuição de CPU

Bloqueado – Processo não pode ser executado até que se reunam as condições de desbloqueio.

Bloqueado <- Execução <-> Executável <-Bloquado

Em que consiste uma tabela de processos ? (2.1.6)

Uma tabela em que cada entrada, uma por processo, contem toda a informação sobre o processo, exceto o conteúdo do seu espaço de endereçamento.

F2.3 Explique o conceito de tarefa (thread). Qual a sua relação com o conceito de processo ?

Uma tarefa é um conjunto de atividades que executam, dentro de um processo existente, uma função do respetivo programa e que partilha, com o processo, o mesmo espaço de endereçamento e os mesmos recursos.

As tarefas existem para cooperarem umas com as outras e assim facilitarem o trabalho.

F2.4 Indique justificando três razões principais da existência de tarefas? (2.2.1)

Desempenho – As tarefas permitem a sobreposição de atividades computacionais, acelerando a aplicação

- Simplicidade – Mais fáceis de criar e destruir que os processos, dado não terem recursos associados

- Decomposição – Múltiplas atividades simultâneas, podem bloquear.

Decompondo a aplicação em várias tarefas sequenciais quase em paralelo, este paralelismo é o mesmo que existe para os processos, mas neste caso as tarefas de um processo utilizam todos o mesmo espaço de endereçamento.

Justifique por que razão não existe protecção de memória entre tarefas. (2.2.2)

As tarefas são concebidas para cooperarem entre si, não estrando sujeitas a acessos mal-intencionados entre elas.

Caracterize o conceito de processo multitarefa. (2.2.2)

Processo Multitarefa contém mais do que um percurso de execução.

Existindo múltiplas tarefas no mesmo processo, partilhando recursos entre elas.

O que entende por uma condição de disputa (race condition) ? (2.3.1)

Condição de disputa como dois ou mais processos acedem concorrentemente ao mesmo recurso.

Resultado final é indeterminado dependendo da ordem de escalonamento dos processos envolvidos.

F2.9 O que entende por um semáforo? (2.3.6)

Mecanismo de sincronização entre processos permitindo controlar o numero de processos que podem aceder a um recurso.

Consiste num contador com 2 operações UP (1) DOWN(0)

Têm dois atributos, uma variável de controlo (inteira) que corresponde ao número de autorizações de acesso aos recursos controlados pelo semáforo e uma fila que guarda os descritores dos processos bloqueados pelo semáforo.

F2.6 O que entende por uma região ou secção crítica?

Região crítica é a aparte do programa em que existe acesso à memória partilhada, em que é necessário assegurar que outros processos sejam impedidos de usar uma variável ou arquivo que já esteja em uso por outro processo. Para evitar as condições de disputa:

Nenhum processo deve esperar eternamente para entrar na região critica.

Nenhum processo executado fora da sua região crítica pode bloquear outros processos

Nunca dois processos podem estar em simultâneo nas suas regiões críticas

Nada pode ser afirmado sobre a velocidade ou sobre o número de Processadores.

F2.7 Descreva quatro situações que possam levar o SO a efectuar um escalonamento. (2.4)

O SO deve efetuar o escalonamento quando ocorrem as seguintes situações:

- É criado um novo processo
- Um processo em execução termina
- Um processo em execução bloqueia
- Ocorre uma interrupção de E/S

Explique o que se entende por separação entre mecanismo de escalonamento (scheduling mechanism) e política de escalonamento (scheduling policy), especificando quem é responsável por cada uma das duas vertentes do escalonamento. (2.4.5)

Mecanismo como o algoritmo de escalonamento responsabilidade do SO.

Política como a parametrização do algoritmo responsabilidade do utilizador.

Explique como pode evitar a ocorrência de situações de impasse (deadlocks) através de uma abordagem à condição de espera circular. Indique quais as limitações desta estratégia. (2.5.1)

Um processo só pode ser atribuído a um recurso de cada vez. Tem que libertar para requisitar outro.

Problema Exemplo – Imprimir a partir de uma tape.

Numeração global de todos os tipos de recursos, requisição feita por ordem, elimina impasse, mas ordenação pode ser complicada porque nunca satisfaz todos.

F2.8 Como se designa a situação em que um processo fica à espera indefinidamente por um recurso embora não se encontre numa situação de interblocagem (deadlock)? (2.5.1)

Este fenómeno designa-se por *starvation (inanição/mingua)* podendo aparecer em situações de carga elevada.

Privação como a espera por tempo indefinido de um recurso que é sistematicamente atribuído a outros processos.

Privação é diferente de impasse!

F2.10 Observe o seguinte programa com dois processos (down↔esperar e up↔assinalar):

```
/* pseudo-código */  
semaforo recurso1, recurso2;  
  
void processoA(void){  
    down(&recurso1);  
    down(&recurso2);  
    utilizar_ambos_os_recursos();  
    up(&recurso2);  
    up(&recurso1);  
}  
  
void processoB(void){  
    down(&recurso2);  
    down(&recurso1);  
    utilizar_ambos_os_recursos();  
    up(&recurso1);  
    up(&recurso2);  
}
```

Este programa pode potencialmente levar a uma situação de interblocagem (deadlock)? Justifique. (2.5.1)

Este programa leva a que acha um deadlock, pois o processo A adquire o recurso 1 e o B o recurso 2, mas quando eles tentam passar para o próximo recurso surge um deadlock pela utilização do outro processo, ou seja, o processo B quando tenta aceder ao recurso um fica bloqueado à espera que o processo A acabe de executar, isto acontece se ambos acabarem ao mesmo tempo, pois ficamos numa situação em que ambos esperam que ambos terminem para poderem passar para o estado seguinte e assim vice-versa.

Capítulo 3 – Gestão Memória

F3.1. O conceito de espaço de endereçamento virtual unidimensional permite atribuir um espaço próprio de endereçamento para cada processo ou tarefa (thread) ? Justifique.

Ao criar um espaço de endereçamento virtual, cada processo fica com o seu próprio espaço de memória (começando no 0) e sem ser afetado por outros processos.

Para o processo o acesso à memória é transparente e o CPU internamente faz a gestão da memória física.

F3.2. No âmbito da memória virtual, a que corresponde o conceito de página ?

A memória é dividida em vários blocos de tamanho fixo. Cada um desses blocos tem a denominação de página, a sua dimensão é uma potencia de 2.

F3.4. Suponha um CPU dotado de registos especiais base e limite. Explique como operam estes registos quando é efectuado um acesso à memória (RAM) de modo a solucionarem os problemas da Realocação e da Protecção. (3.2.1)

A realocação garante que os endereços referenciados por um programa sejam os corretos, independentemente da posição de memória da qual é carregado. O registo base resolve o problema da realocação somando, a todos os endereços referenciados pelo programa, o endereço base da partição onde este é carregado.

A protecção impede que um programa acceda aos endereços de um outro programa. Este problema é tratado com o registo de limite que verifica se os endereços referenciados pelo programa se encontram dentro da partição que lhe é atribuída.

Quando um processo é escolhido para ser executado o registo base é carregado com o endereço de início da partição alocada e o registo limite é carregado com o tamanho da partição, ficando assim o programa confinado à partição.

Em todos os acessos à memória é somado o registo base ao endereço (realocação) e testado se excede o endereço base+limite (protecção)

Em que consiste a técnica de “swapping” ? Que problemas levanta a nível de gestão de memória ? . (3.2.2)

Consiste na transferência de processos da memoria principal para o disco e vice-versa.

Sendo o processo executado na memoria principal.

Esta técnica conduz à proliferação de fragmentos(buracos).

Os processos com tamanho fixo ocupam espaços físicos fixos, já os dinâmicos, ocupam fragmentos adjacentes (alocação dinâmica) quando não existe troca com restantes processo ou o processo é terminado.

F3.5. Considere um sistema de memória virtual com um esquema de Tabela de Páginas mononível. Supondo que não ocorre uma falha de página, descreva qualitativamente como é obtido o endereço físico a partir do endereço virtual.

Quando o processo precisa de aceder a um determinado endereço (na memória virtual), primeiro é verificado em que página se encontra esse endereço (separando o número de página do offset). De seguida procura uma referência a esse número de página na tabela de páginas. Ao encontrar, fica a saber a que página corresponde na memória física. Então o endereço pretendido é obtido da memória física pelo endereço da página, somando o offset.

F3.6. Considere um sistema com memória virtual mononível. O espaço de endereçamento virtual é de 20 bits e a dimensão da página virtual é de 2KB. Quantas entradas tem a Tabela de Páginas ?

$$\text{Entradas} = (2^n \text{ bits endereço virtual}) / (\text{dimensão da página virtual})$$

$$\text{Entradas} = (2^{20}) / 2048 = 512$$

1.7. Considere um sistema com memória virtual e paginação. O espaço de endereçamento virtual é de 21 bits (2MB) e em determinado momento a tabela de páginas (mononível) tem o seguinte conteúdo,

nº entrada (decimal)	conteúdo (binário)	bit presente/ausente (1/0)
511	1011	1
510	1000	1
...
4	1010	1
3	0000	0
2	0101	1
1	1001	1
0	0000	0

1.7.1. [0,8] Considere as dimensões da página virtual e da moldura de página (page frame) iguais. Indique justificando a dimensão da página utilizada e do espaço de endereçamento físico.

1.7.2. [0,8] Considere o endereço virtual 0 0000 0100 0101 0111 1001. Indique justificando o endereço físico correspondente.

1.7.3. [0,8] Considere o endereço virtual 0 0000 0011 1001 1100 1101. Explique sucintamente o que acontece quando o programa efectua uma referência a este endereço.

$$1.7.1 = 2^{21} / 2^9 = 2^{12} \text{ dimensão 4Kb}$$

Dimensão da página tem 4kB, de 12bits deslocamento = 21bits endereço virtual – 9bits indexação da tabela (512 entradas).

Dimensão do espaço de endereçamento físico tem 64kB, de 16bits = 4bits conteúdo entrada tabela + 12bits deslocamento

$$1.7.2 \ 0 \ 0000 \ 0100 = 2^2 = 4 \rightarrow 1010$$

Os 9 bits mais significativos indicam o índice da tabela, que tem bit presente/ausente a 1. O endereço físico é 1010 0101 0111 1001

1.7.3 $0\ 0000\ 0011 = 2^1 + 2^0 = 3 \rightarrow 0000$, i.e. bit ausente

Os 9 bits mais significativos indicam o índice da tabela, que tem bit presente/ausente a 0. a página física que contém o endereço solicitado não se encontra na memória, sendo gerada uma falta de página.

O SO escolhe uma moldura de página pouco usada e salva em disco. Em seguida carrega a página virtual referenciada na moldura libertada e atualiza o mapeamento da tabela de páginas e reinicia a tarefa que provocou o page fault.

Explique sucintamente o algoritmo de substituição de páginas do conjunto de trabalho (Working set page replacement algorithm). (3.4.8)

O conjunto de páginas que um processo está a utilizar é chamado um conjunto de trabalho (Working Set), se todo estiver em memória o processo será executado com poucas falhas.

A ideia do algoritmo é a substituição das páginas que não fazem parte do conjunto de trabalho do processo.

Quando ocorre comutação de processo, antes deste correr o SO carrega para memória o respetivo Working Set.

F3.7. Explique em que consiste uma Tabela de Tradução de Endereços (TLB) e a importância da sua existência.

O principal objetivo da TLB é acelerar a tradução de endereços virtuais em físicos, diminuindo o desperdício de tempo feito pelo acesso a Tabela de Páginas (PTE).

Uma TLB mantém um conjunto de descritores das páginas acedidas recentemente, esta tática resulta porque cada processo tende a usar mais exaustivamente um pequeno conjunto de páginas.

Esta tabela é implementada como uma memória associativa interna à UGM e gerida por hardware. A pesquisa na TLB é lançada em paralelo com o acesso à PTE, se o descritor for encontrado na TLB, é interrompido o acesso à PTE.

Se não encontrado acede à PTE e o descritor é introduzido na TLB.

Quando uma página é descartada da memória principal, é também libertada a entrada correspondente na TLB.

F3.8. O que entende por um algoritmo de substituição de páginas com uma política local e com uma política global ?

Quando ocorre uma Falta de Página temos de escolher uma página que será descartada da memória.

Colocam-se duas hipóteses: escolhemos uma página do processo que causou a falha (opção LOCAL); ou optamos pela página (de todos os processos) mais antiga em memória (opção GLOBAL).

Tendo em conta a abordagem podemos concluir que a solução global funcionará melhor.

Capítulo 4 – Sistema de Ficheiros

F4.1. O que entende por um ficheiro? (4.1)

Um ficheiro é uma coleção de dados persistentes, geralmente relacionados, identificados por um nome.

Um ficheiro é composto por informação e metainformação (descrição), sendo o correspondente eletrónico dos “antigos” ficheiros usados nos escritórios.

F4.2. Descreva sucintamente os dois modos possíveis de acesso a um ficheiro. (4.1)

Modo acesso sequencial – ficheiro é lido de forma ordenada. Para localizar uma parte é necessário lê-lo desde início, logo elemento mais distante demora mais tempo a ser acedido.

Exemplo Descritivo: Rolo de Pergaminho

Exemplo Aplicativo: Fita Magnética

Modo acesso aleatório – possível localizar uma posição com base numa chave. Um elemento distante leva o mesmo tempo a ser acessado do que um elemento próximo.

Exemplo Descritivo: Livro que pode ser aberto aleatoriamente

Exemplo Aplicativo: Discos Rígidos atuais

F4.3. Explique o significado de diretoria de trabalho ou corrente. (4.1)

Cada ficheiro é definido pelo nome absoluto (da raiz ao diretório atual – Ex: “/usr/amss/UAb/pfolio.txt”).

O caminho pode ser longo, tornando o nome absoluto demasiado longo. Para ultrapassar esta questão surgiu o conceito *directoria de trabalho* -> passando os ficheiros a poderem ser designados pelo nome relativo (Ex: “/UAb/pfolio.txt”). O efeito do nome relativo ou absoluto é o mesmo, mas o relativo em determinados momentos poderá ser mais conveniente.

Para um sistema de ficheiros baseado em i-nodes, explique no que consiste e como é implementado um “hard link”. (4.2)

O mesmo ficheiro i-node é referenciado por vários caminhos, possivelmente em várias diretórias.

Existe um contador que contabiliza o número de referencias e os dados associados só serão libertados quando o contador vai a 0.

F4.4. O que é uma ligação simbólica? (4.2)

Existem dois tipos de ligação *Hard Link* e *Soft Link* (simbólica).

Uma ligação simbólica é feita através de um ficheiro link com caminho para o ficheiro original.

Através do caminho acede-se à entrada da diretoria do ficheiro original (com os atributos e localização).

Apesar do dono do ficheiro ter um apontador para o respetivo i-node, se o excluir a ligação simbólica aponta para um caminho que não levará a lugar nenhum.

F4.5. Num sistema de ficheiros que utilize I-nodes, onde se encontra a informação referente aos atributos dos ficheiros? [4.2]

No sistema que utiliza i-nodes, é associado a cada ficheiro uma estrutura de dados que contém os metadados do ficheiro. O i-node contém todas as características do exceto o nome que fura na diretoria onde o i-node é referenciado.

F4.6. Qual é a principal desvantagem da implementação de um sistema de ficheiros baseado numa Tabela de Alocação de Ficheiros (FAT) ? Dê um exemplo numérico. [4.3.2]

O principal problema é a elevada dimensão da tabela de alocação quando os discos são de grande dimensão.

Exemplo: Disco de 1TB e blocos de 2KB necessita FAT com 500 milhões entradas, requerendo 2GB memória.

F4.7. Escolher a dimensão do bloco que serve como unidade de alocação de memória num disco, significa um compromisso entre dois objetivos a optimizar: rapidez de acesso e desperdício de memória. Explique porquê [4.4.1]

Blocos Grande Dimensão – temos maior rapidez de acesso por termos um menor número de blocos. A lista de blocos será menor e a sua busca, leitura e escrita mais rápida. Se os ficheiros forem pequenos iremos desperdiçar espaço.

Blocos Pequena Dimensão – temos menor desperdício de memória, pois mesmo os ficheiros mais pequenos são subdivididos e distribuídos pelos blocos. Por contra partida aumentamos o tempo de acesso, a lista é maior e os métodos a aplicar serão mais demorados.

O compromisso entre os dois cenários, em que maximizamos a rapidez de acesso, ao mesmo tempo que diminuímos desperdício de memória, está atualmente nos 2Kb por bloco.

F4.8. O que entende por um disco fragmentado ? Nessa situação, o que acontece quando é criado um novo ficheiro cuja dimensão ocupe vários blocos ?

Com a utilização do disco (criar, acrescentar, eliminar) este fica preenchido por ficheiros e espaços livres, cujos blocos estão dispersos pelo disco.

Quando é criado um novo ficheiro que ocupe vários blocos, estes não serão contíguos mas dispersos, o que faz com que tenhamos menor desempenho nas operações de leitura/escrita.

Num sistema de ficheiros que utilize I-nodes, onde se encontra a informação referente aos atributos dos ficheiros ? [6.1.1]

O i-node é uma estrutura de dados que entre outras informações contém os atributos do ficheiro.

Capítulo 5 – Entradas/Saídas (I/O)

F5.1. De um ponto de vista genérico, quais são as principais responsabilidades do SO relativamente a dispositivos de entrada/saída (I/O) ?

O SO controla os dispositivos periféricos. É responsável por enviar comandos, receber e enviar dados, detetar as interrupções (servem para indicar ao SO que o dispositivo está pronto para receber/enviar.) e tratamento de erros (por exemplo durante a transferência interrompida de dados).

O SO é responsável por providenciar uma interface simples e uniforme para o maior número de dispositivos.

F5.2. A maioria dos dispositivos de I/O pode ser classificada em duas categorias de dispositivos. Indique quais e dê um exemplo de um dispositivo para cada uma.

Dispositivos de blocos – Guardam informação em blocos de dimensão fixa, cada um com o seu endereço – Ex: Discos Rígidos e CDROM's.

Dispositivos de carácter – aceitam um conjunto contínuo de bytes – Ex: Teclados e Ratos...

F5.3. Como se processa a comunicação entre o SO e um controlador de dispositivo, quer a nível de comandos, quer a nível de dados ?

Duas Formas -> Portos de I/O e Memória

Portos – Atribuído um número (Porto I/O) a cada registo dos controladores. O acesso é realizado através da instrução em linguagem de baixo nível. Espaço I/O e espaço memória separados.

Memória – cada registo é mapeado para uma posição e o acesso tratasse como um acesso à memória.

F5.4. Qual o nome do circuito de controlo que permite a comunicação directa entre um dispositivo de I/O e a memória RAM ?

O nome do circuito é DMA (Direct Memory Access).

DMA é um circuito auxiliar que a CPU pode usar para efetuar transferência de dados entre a RAM e um dispositivo, ficando livre para outros processamentos.

F5.5. Indique e justifique a principal vantagem e desvantagem do método de entrada/saída designado por I/O programado.

Neste método a CPU realiza todo o trabalho de IO.

Vantagem – Facilidade de compreensão e implementação.

Desvantagem – Faz com que processador fique muito tempo em espera.

Explique em que consiste um esquema híbrido de comunicação entre um processador e um circuito controlador de dispositivo. (5.1.3)

Utilização de pontos de Entrada/Saída para cesso aos registo de controlo do dispositivo e de endereço de memória para troca de dados.

No âmbito do software de I/O, explique o conceito de “independência do dispositivo”. (5.1.3)

Independência do dispositivo como a operação de diferentes de diferentes dispositivos com a mesma interface de programação.

Em que consiste a técnica de spooling ? (5.3.4)

Spooling consiste em armazenamento de trabalhos em disco para que assim que o sistema estiver disponível para processamento, este poder iniciar o próximo trabalho o mais rápido possível.

A técnica é também usada para aceder a dispositivos de I/O.

Explique os conceitos de recursos preemptíveis e não preemptíveis. Dê um exemplo de cada. (6.1.1)

Recurso Preemptível – Aquele que pode em qualquer altura ser retirado a um processo e atribuído a outro – Exemplo memória RAM

Recurso não Preemptível, não pode, sendo exemplo uma impressora.

Descreva as quatro condições que têm que se verificar para que possa ocorrer uma situação de impasse (deadlock) de recursos. (6.2.1)

1. Atribuição de recursos em exclusão mutua
2. Processo pode reter um recurso e esperar por outro
3. Não preempção de recursos
4. Espera Circular

Explique como o sistema operativo gere os processos quando um processo efectua um pedido de entrada/saída (I/O) de um dispositivo que opera por interrupção.

O processador envia/recebe dados o periférico, mas depois pode dedicar-se a outro processo.

Quando periférico pronto a continuar interrompe processador.

Após interrupção, processador envia/recebe dados...e assim sucessivamente até que operação I/O esteja concluída.