

Sistemas Operacionais

Prof. Dr. Helder Oliveira

Plano de Aula

- Princípios do hardware de E/S
- Princípios do software de E/S

Entrada/ Saída no SO

- Dispositivos de entrada e saída são controlados pelos Sistema operacional.
- Sistema operacional deve emitir comandos para os dispositivos, interceptar interrupções e lidar com erros.
- SO deve fornecer uma interface entre os dispositivos e o resto do sistema que seja simples e fácil de usar.
- A interface deve ser a mesma para todos os dispositivos (independentemente do dispositivo).
- O código de E/S representa uma fração significativa do sistema operacional total.

Princípios do hardware de E/S

- Visão do hardware:
 - Engenheiro elétrico.
 - Chips, cabos, motores, suprimento de energia e todos os outros componentes físicos que compreendem o hardware.
 - Programadores
 - Interface do software.
 - Comandos aceitos pelo hardware.
 - Funções que realiza.
 - Erros.

Dispositivos de E/S

- Podem ser classificadas:
 - **Tipo de conexão**
 - **Tipo de transparência dos dados**
 - **Tipo de compartilhamento de conexões.**

Dispositivos de E/S

Tipo de conexão

- Refere-se a natureza da conexão entre:
 - Módulo de E/S e
 - Periférico.
- Serial
 - Única linha de conexão.
 - Mais barata e mais lenta que a paralela
 - Relativamente confiável.
 - Usada em dispositivos mais baratos e lentos.
- Paralela
 - Varias linhas de conexão.
 - Mais complexa.
 - Mais cara e mais rápida.
 - Altamente confiável.
 - Usada em dispositivos mais velozes.

Dispositivos de E/S

Tipo de transparência dos dados

- Divididos de modo geral em duas categorias:
 - **Dispositivos de blocos**
 - Armazena informações em blocos de tamanho fixo, cada um com seu próprio endereço.
 - Cada bloco pode ser lido ou escrito independentemente de todos os outros.
 - Discos rígidos, discos Blu-ray e pendrives são dispositivos de bloco comuns.
 - **Dispositivos de caractere.**
 - Envia ou aceita um fluxo de caracteres, desconsiderando qualquer estrutura de bloco.
 - Não é endereçável e não tem qualquer operação de busca.
 - Impressoras, interfaces de rede, mouses, ...

Dispositivos de E/S

Tipo de transparência dos dados

- Esquema de classificação não é perfeito.
- Alguns dispositivos não se enquadram nessas duas categorias.
- Ex: Relógios
 - Relógios não são endereçáveis por blocos.
 - Relógios não aceitam ou geram fluxos de caracteres.
- Ex: Telas
- Mesmo assim, o modelo de dispositivos de blocos e de caractere é suficientemente geral para ser usado como uma base.

Dispositivos de E/S

FIGURA 5.1 Algumas taxas de dados típicas de dispositivos, placas de redes e barramentos.

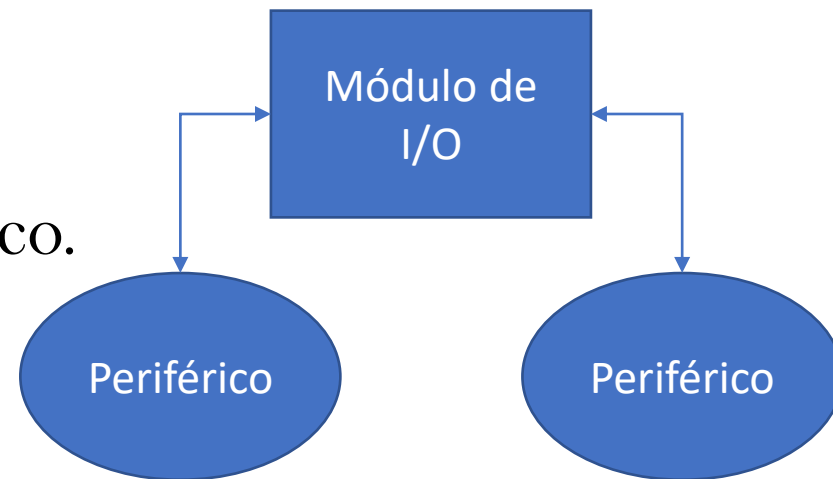
Dispositivo	Taxa de dados
Teclado	10 bytes/s
Mouse	100 bytes/s
Modem 56 K	7 KB/s
Scanner em 300 dpi	1 MB/s
Filmadora <i>camcorder</i> digital	3,5 MB/s
Disco Blu-ray 4x	18 MB/s
Wireless 802.11n	37,5 MB/s
USB 2.0	60 MB/s
FireWire 800	100 MB/s
Gigabit Ethernet	125 MB/s
Drive de disco SATA 3	600 MB/s
USB 3.0	625 MB/s
Barramento SCSI Ultra 5	640 MB/s
Barramento de faixa única PCIe 3.0	985 MB/s
Barramento Thunderbolt2	2,5 GB/s
Rede SONET OC-768	5 GB/s

Dispositivos de E/S

Tipo de compartilhamento de conexões.

- **Ponto a Ponto**

- É a conexão mais simples.
- Linha dedicada para a ligação.
- Entre o módulo de E/S e periférico.
- Oferece maior confiabilidade.
- Paralelismo.

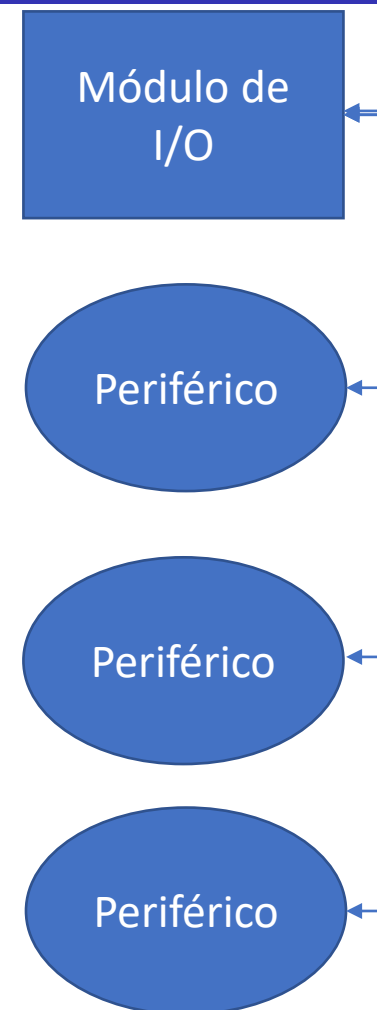


Dispositivos de E/S

Tipo de compartilhamento de conexões.

- **Multiponto**

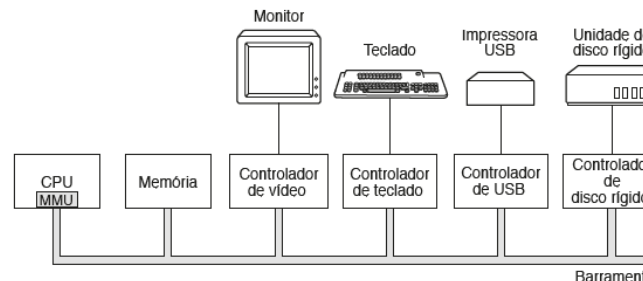
- Compartilham um conjunto de linhas.
- Entre diversos periféricos.
- Maior escalabilidade.
- Não permite paralelismo.
- Usada para armazenamento.
- Interface de rede: crossbar.



Controladores de dispositivos

- Unidades de E/S consistem, em geral, de um componente mecânico e um componente eletrônico.
- O componente eletrônico é chamado de **controlador do dispositivo** ou **adaptador**.
- Em computadores pessoais, ele muitas vezes assume a forma de um chip na placa-mãe ou um cartão de circuito impresso.
- O componente mecânico é o dispositivo em si.

FIGURA 1.6 Alguns dos componentes de um computador pessoal simples.



Controladores de dispositivos

- Muitos controladores podem lidar com dois, quatro ou mesmo oito dispositivos idênticos.
- Se a interface for padrão:
 - Empresas podem produzir controladores ou dispositivos que se enquadrem àquela interface.
- Existem interfaces de baixo nível e alto nível.
 - Baixo nível: entre a controladora e o dispositivo.
 - Alto nível: entre o SO e a controladora.

Controladores de dispositivos

- Exemplo com uma controladora de disco:
- Recebe um fluxo de bits com:
 - Preâmbulo
 - Os bits do setor.
 - Checksum (Error Correcting Code - ECC)
- Dispositivos:
 - Monta os bytes em bloco.
 - Coloca-se em um buffer interno.
 - Após verificar a checksum.
 - Copia o bloco da RAM.

E/S mapeada na memória

- Controladores possuem registradores que são usados para comunicar-se com a CPU.
- Escrevendo no registrador o SO pode realizar comandos.
- Lendo no registrador o SO pode descobrir o estado do dispositivo.
- Além dos registradores de controle, muitos dispositivos têm um buffer de dados a partir do qual o sistema operacional pode ler e escrever.
- Como CPU se comunica com os registradores de controle e também com os buffers de dados do dispositivo?

E/S mapeada na memória

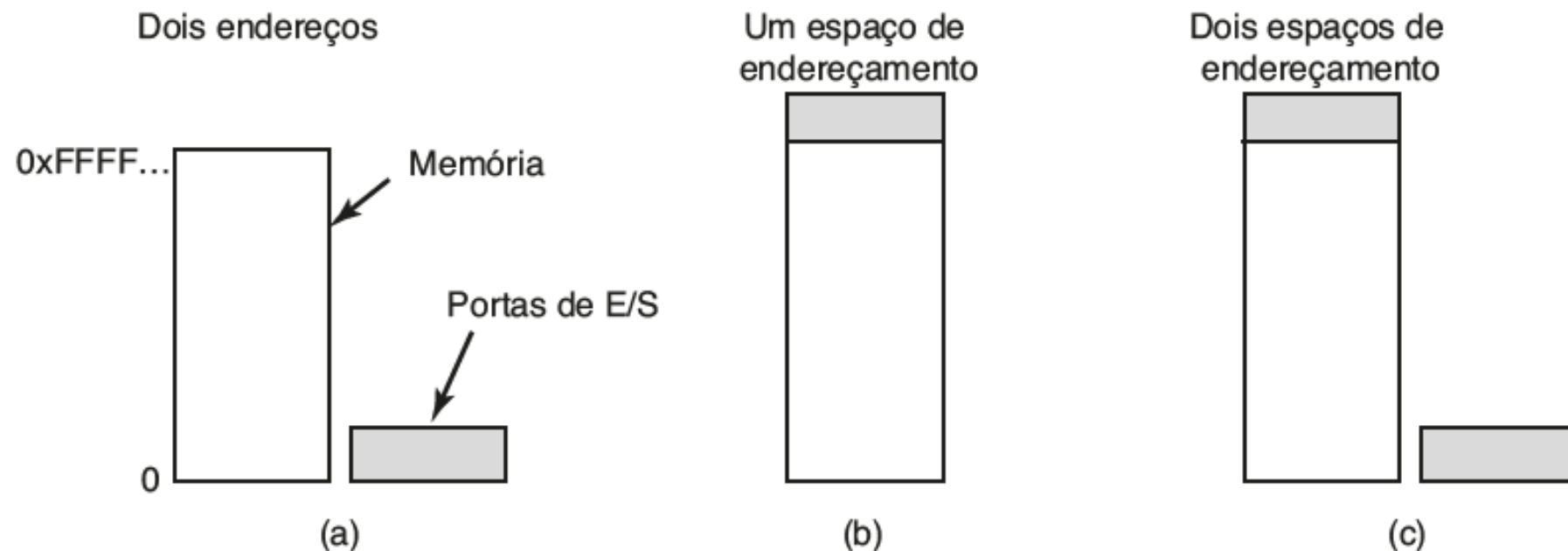
Duas abordagens:

1. Para cada registrador de controle é designado um número de porta de E/S.
 - O conjunto de todas as portas de E/S formam o **espaço de E/S**.
 - O espaço de E/S é protegido de maneira que apenas o SO consiga acessá-lo.
 - Utiliza instruções especiais **IN REG,PORT**, e **OUT PORT,REG**
2. Mapear todos os registradores de controle no espaço da memória.
 - Para cada registrador de controle é designado um endereço de memória único para o qual nenhuma memória é designada.
 - Esse sistema é chamado de E/S mapeada na memória.

E/S mapeada na memória

- Duas abordagens:

FIGURA 5.2 (a) Espaços de memória e E/S separados. (b) E/S mapeada na memória. (c) Híbrido.



E/S mapeada na memória

Como esses esquemas realmente funcionam na prática?

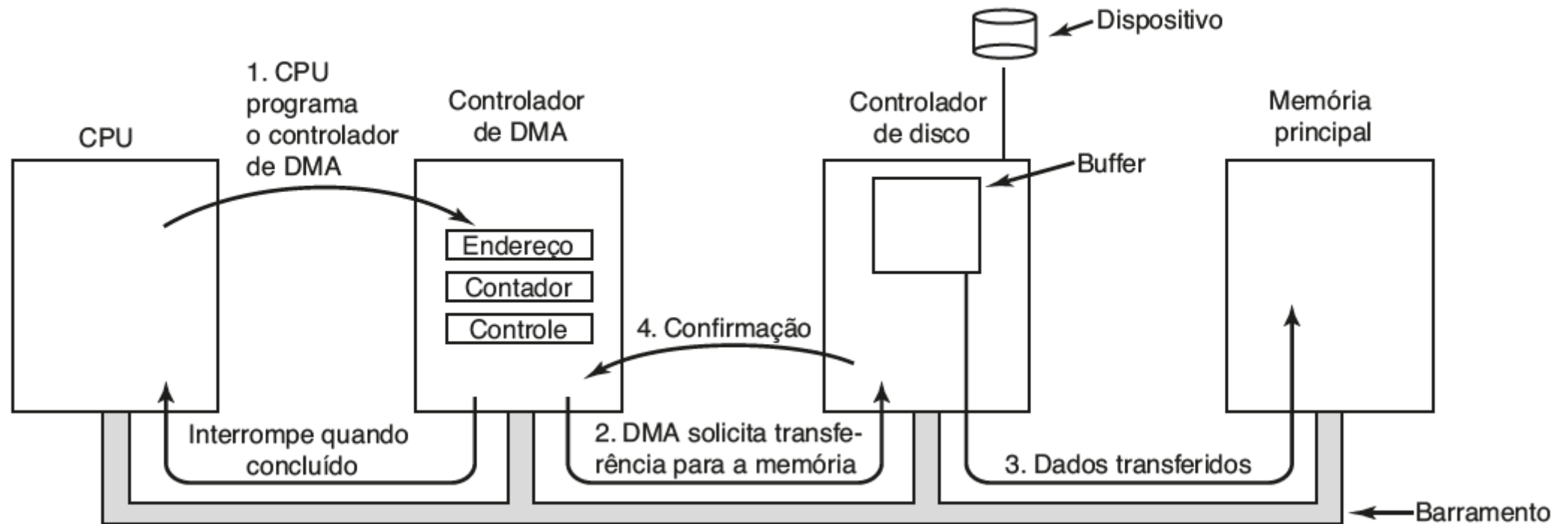
- Quando a CPU quer ler uma palavra,
 - Ela coloca o endereço de que precisa nas linhas de endereçamento do barramento e então emite um sinal READ sobre uma linha de controle do barramento.
 - Uma segunda linha de sinal é usada para dizer se o espaço de E/S ou o espaço de memória é necessário.
 - Se for o espaço de E/S:
 - O dispositivo de E/S responde ao pedido.
 - Se houver apenas espaço de memória.
 - Cada módulo de memória e cada dispositivo de E/S comparam as linhas de endereços com a faixa de endereços que elas servem. Se o endereço cair na sua faixa, ela responde ao pedido.

Acesso direto à memória

- CPU precisa endereçar os controladores dos dispositivos para poder trocar dados com eles.
- A CPU pode requisitar dados de um controlador de E/S um byte de cada vez, mas fazê-lo desperdiça o tempo da CPU.
- Por isso utiliza o **acesso direto da memória**.

Acesso direto à memória

FIGURA 5.4 Operação de transferência utilizando DMA.



Acesso direto à memória

- Muitos barramentos podem operar em dois modos:
 - **Modo uma palavra de cada vez**
 - O controlador de DMA solicita a transferência de uma palavra e consegue. Se a CPU também quiser o barramento, ela tem de esperar.
 - **Roubo de ciclo**: o controlador do dispositivo entra furtivamente e rouba um ciclo de barramento ocasional da CPU.
 - **Modo bloco.**
 - O controlador de DMA diz para o dispositivo para adquirir o barramento, emitir uma série de transferências, então libera o barramento.
 - **Modo de surto** (burst): emite uma serie de transferências.
 - Mais eficiente, mas pode bloquear a CPU e outros dispositivos por um tempo..

Princípios do software de E/S

- Um conceito fundamental no projeto de software de E/S é conhecido como **independência de dispositivo**.
- O que isso significa é que devemos ser capazes de escrever programas que podem acessar qualquer dispositivo de E/S sem ter de especificá-lo antecipadamente.
- Por exemplo, um programa que lê um arquivo como entrada deve ser capaz de ler um arquivo em um disco rígido, um DVD ou em um pen-drive sem ter de ser modificado para cada dispositivo diferente.

Princípios do software de E/S

- Um objetivo muito relacionado com a independência do dispositivo é a **nomeação uniforme**. O nome de um arquivo ou um dispositivo deve simplesmente ser uma cadeia de caracteres ou um número inteiro e não depender do dispositivo de maneira alguma.
- **EXEMPLO:** Um pen-drive pode ser montado em cima do diretório `/usr/ast/backup` de maneira que, ao copiar um arquivo para `/usr/ast/backup/Monday`, você copia o arquivo para o pen-drive. Assim, todos os arquivos e dispositivos são endereçados da mesma maneira: por um nome de caminho.

Princípios do software de E/S

- Outra questão importante para o software de E/S é o tratamento de erros. Em geral, erros devem ser tratados o mais próximo possível do hardware.
- Se o controlador descobre um erro de leitura, ele deve tentar corrigi-lo se puder. Se ele não puder, então o driver do dispositivo deverá lidar com ele, talvez simplesmente tentando ler o bloco novamente.
- Muitos erros são transitórios, como erros de leitura causados por grãos de poeira no cabeçote de leitura, e muitas vezes desaparecerão se a operação for repetida. Apenas se as camadas mais baixas não forem capazes de lidar com o problema as camadas superiores devem ser informadas a respeito.
- Em muitos casos, a recuperação de erros pode ser feita de modo transparente em um nível baixo sem que os níveis superiores sequer tomem conhecimento do erro.

Princípios do software de E/S

- Ainda outra questão importante é a das transferências **síncronas** (bloqueantes) versus **assíncronas** (orientadas à interrupção). A maioria das E/S físicas são assíncronas — a CPU inicializa a transferência e vai fazer outra coisa até a chegada da interrupção. Programas do usuário são muito mais fáceis de escrever se as operações de E/S forem bloqueantes — após uma chamada de sistema read, o programa é automaticamente suspenso até que os dados estejam disponíveis no buffer.
- Fica a cargo do sistema operacional fazer operações que são realmente orientadas à interrupção parecerem bloqueantes para os programas do usuário. No entanto, algumas aplicações de muito alto desempenho precisam controlar todos os detalhes da E/S, então alguns sistemas operacionais disponibilizam a E/S assíncrona para si.

Princípios do software de E/S

- Outra questão para o software de E/S é a utilização de **buffer**. Muitas vezes, dados provenientes de um dispositivo não podem ser armazenados diretamente em seu destino final. Por exemplo, quando um pacote chega da rede, o sistema operacional não sabe onde armazená-lo definitivamente até que o tenha colocado em algum lugar para examiná-lo.
- Também, alguns dispositivos têm severas restrições de tempo real (por exemplo, dispositivos de áudio digitais), portanto os dados devem ser colocados antecipadamente em um buffer de saída para separar a taxa na qual o buffer é preenchido da taxa na qual ele é esvaziado, a fim de evitar seu completo esvaziamento. A utilização do buffer envolve consideráveis operações de cópia e muitas vezes tem um impacto importante sobre o desempenho de E/S.

Princípios do software de E/S

- O conceito final que mencionaremos aqui é o de dispositivos **compartilhados** versus **dedicados**. Alguns dispositivos de E/S, como discos, podem ser usados por muitos usuários ao mesmo tempo. Nenhum problema é causado por múltiplos usuários terem arquivos abertos no mesmo disco ao mesmo tempo.
- Outros dispositivos, como impressoras, têm de ser dedicados a um único usuário até ele ter concluído sua operação. Então outro usuário pode ter a impressora. Ter dois ou mais usuários escrevendo caracteres de maneira aleatória e intercalada na mesma página definitivamente não funcionará. Introduzir dispositivos dedicados (não compartilhados) também introduz uma série de problemas, como os impasses. Novamente, o sistema operacional deve ser capaz de lidar com ambos os dispositivos — compartilhados e dedicados — de uma maneira que evite problemas.

Princípios do software de E/S

- A entrada/saída é um tópico importante, mas muitas vezes negligenciado. Uma fração substancial de qualquer sistema operacional diz respeito à E/S. A E/S pode ser conseguida de três maneiras.
 - 1º: há a **E/S programada**, na qual a CPU principal envia ou recebe cada byte ou palavra e aguarda em um laço estreito esperando até que possa receber ou enviar o próximo byte ou palavra.
 - 2º: há a **E/S orientada à interrupção**, na qual a CPU inicia uma transferência de E/S para um caractere ou palavra e vai fazer outra coisa até a interrupção chegar sinalizando a conclusão da E/S.
 - 3º: há o **DMA**, no qual um chip separado gerencia a transferência completa de um bloco de dados, gerando uma interrupção somente quando o bloco inteiro foi transferido.

Leitura

- SISTEMAS OPERACIONAIS MODERNO 4^a edição
 - Capítulo 5 – Entrada/saída

Dúvidas?