

Sistemas Operacionais

Entrada e Saída

Introdução

- Uma das principais funções de um sistema operacional é controlar os dispositivos de entrada e saída;
- O sistema operacional deve disponibilizar uma interface entre os dispositivos e o resto do sistema que seja simples e fácil de usar;
- O sistema operacional não se preocupa com o funcionamento interno e com a forma que é construído o hardware de entrada e saída, mas com a programação e utilização deste hardware.

Introdução

Interfaces padronizadas X variedade de dispositivos.

Device	Data rate
Keyboard	10 bytes/sec
Mouse	100 bytes/sec
56K modem	7 KB/sec
Telephone channel	8 KB/sec
Dual ISDN lines	16 KB/sec
Laser printer	100 KB/sec
Scanner	400 KB/sec
Classic Ethernet	1.25 MB/sec
USB (Universal Serial Bus)	1.5 MB/sec
Digital camcorder	4 MB/sec
IDE disk	5 MB/sec
40x CD-ROM	6 MB/sec
Fast Ethernet	12.5 MB/sec
ISA bus	16.7 MB/sec
EIDE (ATA-2) disk	16.7 MB/sec
FireWire (IEEE 1394)	50 MB/sec
XGA Monitor	60 MB/sec
SONET OC-12 network	78 MB/sec
SCSI Ultra 2 disk	80 MB/sec
Gigabit Ethernet	125 MB/sec
Ultrium tape	320 MB/sec
PCI bus	528 MB/sec
Sun Gigaplane XB backplane	20 GB/sec

3

Tipos de Dispositivos

- Existem basicamente dois tipos de dispositivos de entrada/saída
 - Dispositivos de Bloco
 - Armazenam informações em blocos de tamanho fixo, cada um com seu próprio endereço;
 - Possibilitam a leitura ou escrita de cada bloco independentemente.
 - Dispositivos de Caractere
 - Aceitam como entrada ou geram como saída seqüências de caractere sem nenhuma estrutura.
- Alguns dispositivos não se encaixam nesta classificação.

4

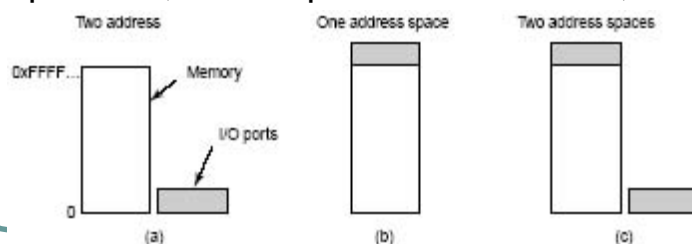
Controlador de Dispositivos

- **Componente Eletrônico:**
 - Controlador ou adaptador;
- **Componente mecânico:**
 - Dispositivo em si.
- Cada controlador tem alguns registradores que são utilizados para a comunicação com o processador;
- Nos SO com entrada/saída mapeada em memória, estes registradores ocupam posições no espaço de endereçamento da memória principal
- O SO executa as operações de entrada/saída escrevendo comandos nos registradores dos dispositivos.

5

E/S Mapeada em Memória

- **O SO manipula o dispositivo através de registradores e memória:**
 - comandar o dispositivo, desligar ou executar tarefas, verificar o estado, etc.
- **3 tipos: Endereçamento de memória e E/S separados; E/S mapeada na memória; e Híbrido.**



6

E/S Mapeada em Memória

- *O mapeamento de memória facilita a escrita, porém é vulnerável a modificações acidentais.*
- Um dispositivo se comunica com o sistema de computador enviando sinais por cabo ou ar.
- Se um ou mais dispositivos usarem um conjunto comum de fios condutores, a conexão é chamada de barramento.
- Um barramento dedicado entre a memória e a CPU pode ser implementado para otimizar o desempenho da memória.

7

DMA

- O controlador lê um bloco e armazena no buffer interno;
- Após a verificação de erros, o controlador solicita o processador que atualiza a memória.
- ➡ Uso ineficiente do processador.
- **Alternativa:** processador especial de controle de acesso direto.
- O DMA inicia a transferência direta para memória, emitindo pelo barramento uma requisição de leitura para o controlador.

8

Objetivos do Software de E/S

- Independência de dispositivos
 - Os programas devem estar aptos a acessar qualquer dispositivo. Ex.: ler arquivo (HD, CD-ROM, disquete etc.)
- Nomeação uniforme
 - Dispositivos referenciados de maneira simples. Ex.: montar.
- Tratamento de erros
 - Deve ser tratado preferencialmente em nível mais baixo.
- Tipo de transferência
 - Síncrona (bloqueante) e assíncrona (irq).
- Utilização de buffer
 - Armazenamento temporário.
- Dispositivos Compartilhados X Dedicados.

Deadlock

9

Maneiras de Realizar E/S

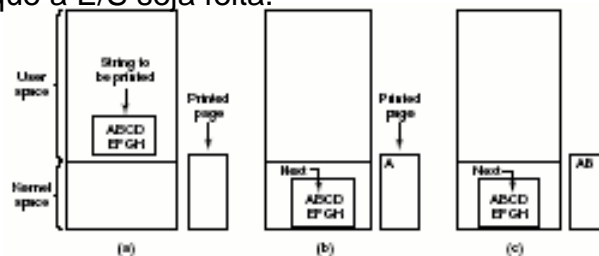
- E/S Programada
 - método mais simples (a CPU faz todo o trabalho).
 - Ex.: Impressão de documento...
1. usuário monta a cadeia (espaço do usuário/buffer);
 2. processo do usuário requisita a impressora;
 3. se a impressora existe, o processo faz uma chamada;
 4. SO copia o buffer no kernel;
 5. SO verifica se impressora está disponível;
 6. se disponível, SO copia o primeiro caracter para o registrador de dados da impressora;
 7. se existe mais impressão, volta para o passo 5.

10

Maneiras de Realizar E/S

● E/S Programada

- Após a saída de um caracter, a CPU verifica continuamente se o dispositivo está pronto para aceitar outro.
- Comportamento denominado *polling* ou *busy waiting*.
- A E/S programada segura a CPU o tempo todo até que a E/S seja feita.



11

Maneiras de Realizar E/S

● E/S Orientada à Interrupção

- Considera que o tempo que a CPU fica no laço ocioso é suficiente para chaveamento.
- O caractere é copiado tão logo a impressora esteja apta a recebê-lo.
- Nesse momento, a CPU chama o escalonador.
- Processo que solicitou a impressão é bloqueado até que a cadeia seja impressa.
- Quando a impressora imprimiu um caractere e está pronta para receber outra, ela gera uma interrupção.
- O SO salva o contexto do processo continua a impressão.

12

Maneiras de Realizar E/S

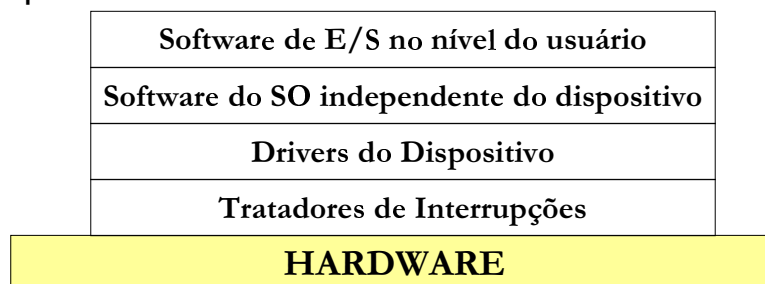
● E/S usando DMA

- Uso de interrupções leva ao desperdício de CPU por conta da interrupção a cada caractere.
 - Solução: usar a DMA.
 - A DMA alimenta a impressora sem o uso da CPU.
 - Variação da E/S programada (executada pela DMA).
 - Reduz o número de interrupções (a cada buffer impresso e não a cada caractere impresso).
 - Entretanto, a DMA é mais lenta que CPU.
- ➔ Se CPU ociosa ou velocidade máxima do dispositivo subutilizada, então as outras alternativas são mais indicadas.

13

Camadas do Software de E/S

- Geralmente organizado em quatro camadas.
- Funções e interfaces bem definidas.
- Funcionalidade e interfaces variam de sistema para sistema.



14

Camadas do Software de E/S

- Implementação estruturada em camadas
 - Baixo nível escondem características do hardware das camadas superiores.
 - Velocidade de operação;
 - Unidade de transferência;
 - Representação de dados;
 - Tipos das Operações.
 - Camadas divididas em dois grupos:
 - 1.º grupo** - Visualiza os diversos tipos de dispositivos de um modo único;
 - 2.º grupo** - Específico para cada dispositivo.

15

Camadas do Software de E/S

- Rotina de interrupção
 - Bloqueia o driver até que a E/S termine e uma interrupção ocorra.
- Drivers
 - Natureza dos comandos varia de dispositivo para dispositivo. Ex.: mouse x disco.
 - Cada dispositivo precisa de um código específico → *driver de dispositivo*
 - fornecido pelo fabricante juntamente com o dispositivo para diferentes SO.

16

Camadas do Software de E/S

- Cada driver trata um tipo de dispositivo.
 - Drivers carregados dinamicamente.
 - Iniciar o dispositivo, executar requisição de leitura/gravação etc.
- Funções:
 - comunicação com o dispositivo de E/S no nível do hardware e especificação das características físicas do dispositivo.
 - Recebem comandos gerais e traduzem para comandos específicos a serem executados pelos controladores.

17

Camadas do Software de E/S

- Independência de dispositivo
 - Deve prover uma interface uniforme para drivers de dispositivo
 - fazer todos os dispositivos parecerem similares.
 - Uso de buffer para liberar dispositivo e diminuir interrupções.
 - Relatar ao processo um código de erro.
 - Alocar/liberar dispositivos dedicados(falha/fila)
 - Considerar que dispositivos possuem tamanho de bloco diferentes entre si.

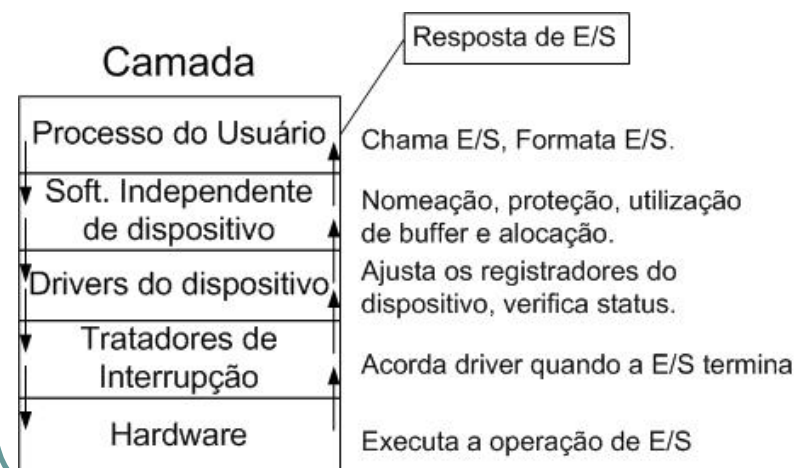
18

Camadas do Software de E/S

- Espaço do usuário
 - funções e procedimentos contidos no programa do usuário.
- **EXEMPLO: ler um bloco de arquivo.**
 - O SO é requisitado para atender a chamada;
 - O software independente de disp. busca o buffer;
 - Caso não esteja, o driver é chamado para executar uma requisição ao hardware;
 - O processo é bloqueado até a E/S terminar;
 - Quando o disco termina é gerado uma interrupção;
 - O tratador de interrupção atende ao disp. e acorda processo que estava dormindo.

19

Resumindo as Camadas do Soft



20

Discos

- Tempo de E/S determinado por: *seek*, *latência* e *taxa de transferência*.
- maior influência \Rightarrow Posicionamento.
- Possibilidade de seeks sobrepostos:
 - Controladora pode fazer seeks simultâneos em unidades distintas;
 - Entretanto leitura/gravação não podem ser realizadas simultaneamente.
- Grande possibilidade de quando o braço estiver sendo posicionado para atender alguma requisição, outras requisições podem ser geradas por outros processos.

21

Discos

- Drivers mantêm uma tabela
 - Requisições pendentes para cada cilindro.
- Algoritmos de escalonamento de braços de disco:
- FCFS (First Come, First Serve)
 - Atende as requisições na ordem que foram recebidas.
 - Ex.: atendimento de requisição de posicionamento para o cilindro 11.
 - Novas requisições: 1, 36, 16, 34, 9 e 12. \Rightarrow tabela.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

22

Discos

- Ao finalizar o posicionamento (11), o driver pode escolher qual será a próxima requisição a ser atendida.
- FCFS: Próxima requisição \Rightarrow cilindro 1 (10 cilindros percorridos); \Rightarrow cilindro 36 (35 cilindros percorridos); \Rightarrow cilindro 16 (20 cilindros percorridos); \Rightarrow cilindro 34 (18 cilindros percorridos); \Rightarrow cilindro 9 (25 cilindros percorridos); \Rightarrow cilindro 12 (3 cilindros percorridos).
- Σ cilindros percorridos: 111 cilindros.
- Alternativas:
 - Posicionamento mais curto primeiro:
 - SSF (Shortest Seek First)

23

Discos

- Atender à requisição mais próxima da posição atual da cabeça de leitura/gravação.
- SSF: Próxima requisição \Rightarrow cilindro 12 (1 cilindro percorrido); \Rightarrow cilindro 9 (3 cilindros percorridos); \Rightarrow cilindro 16 (7 cilindros percorridos); \Rightarrow cilindro 1 (15 cilindros percorridos); \Rightarrow cilindro 34 (33 cilindros percorridos); \Rightarrow cilindro 36 (2 cilindros percorridos).
- Σ cilindros percorridos: 61 cilindros.
- Problema: novas requisições podem chegar.
- Ex.: posição atual \Rightarrow cilindro 16
- Novas requisições: cilindros 8 e 13 (prioridade sobre o cilindro 1)

24

Discos

- ⇒ Tendência do braço permanecer no meio do disco.
- Objetivos menor tempo de resposta e justiça conflitantes.
- Algoritmo do Elevador
 - Mover em uma direção até não haver mais requisições pendentes naquele sentido.
 - Manutenção de um bit: **sobe/desce**.
 - Ex.: bit = sobe
 - Ordem: 12 (1); 16 (4); 34 (18); 36 (2); 9 (27); 1 (8).
 - Σ cilindros percorridos: 60 cilindros.
- Tentativa de aumentar desempenho:
 - Cache de controladora/RAM Disk.

25

Tratamento de erros

- Discos são dispositivos sujeitos a grande variedade de erros:
 - Erro de programação: programador solicita a controladora setor inexistente → verificar parâmetros / abortar requisição: reportar erro.
 - Erro transiente: partícula de sujeira entre cabeça e superfície → repetição da operação.
 - Erro permanente: bloco danificado → marcar como ruim. Uso de setores reservas (transparente).
 - Erro de seek: problemas mecânicos no braço → correção automática: *recalibrate*.
 - Erro de controladora: comandos não são aceitos → uso de bit para resetar controladora.

26

Desempenho e segurança

- **Armazenamento estável**
 - Uso de par de discos idênticos com os blocos correspondentes trabalhando juntos para formar um bloco livre de erro.
 - Para atingir o objetivo, 3 operações são definidas:
 - Escritas estáveis: escrever bloco no disco 1 e em seguida ler o mesmo dado. Idem no disco 2.
 - Leituras estáveis: ler bloco do disco 1. Sempre será possível ao menos **uma** leitura estável.
 - Recuperação de falhas: após falha do sistema, um programa de recuperação varre par de blocos.
 - Variação do **espelhamento** (mirroring/shadowing).

27

Relógios

- **Clocks, relógios, temporizadores ou timers.**
 - Essenciais para sistemas multiprogramáveis.
 - **Dois tipos básicos:**
 - Ligados a uma linha de voltagem: mais simples → uma interrupção a cada ciclo.
 - Formado por 3 componentes: *cristal, contador e registrador de apoio* → cristal gera sinal periódico de alta precisão. *Contador Regressivo (0)*: IRQ é gerada.
 - Modos de operação (cópia do valor RA ao contador):
 - *One-shot mode*: chegando em zero, contador pára;
 - Modo onda quadrada: tiques do relógio.
 - Relógio de segurança mantido por bateria.

28

Terminais

- Permitem a comunicação entre o usuário e o sistema.
 - Standalone: uso de interfaces seriais (computadores de grande porte);
 - Vídeos de computadores pessoais com GUI;
 - Terminais de Rede.
- Terminal Serial
 - Padrão RS-232 (conector com 9/25 pinos)
 - 3 pinos o tornam funcional.
 - Emulação de terminais: Executar grande gama de programas/ fornecer interface poderosa.
 - Baseado em texto (25 x 80).

29

Terminais

- Uso de chip para conversão (bit \leftrightarrow caractere).
 - UART (Transmissores e Receptores Assíncronos Universais)
- Coletar a entrada do teclado e transferir aos programas (seqüência de códigos ASCII)
 - Baseada em caracteres: *RAW MODE*;
 - Baseada em linhas: *COOKED MODE*;
 - ECO, CR, LF.
- Saída escrita em Buffer
 - Caractere enviado a cada pedido de interrupção.

30

Terminais Gráficos

- Computadores Pessoais usam GUI
- Quatro elementos essenciais: **WIMP**.
- Hardware de E/S
 - Teclado: uso de registradores de E/S → **Driver** converte em código ASCII.
 - Mouse: mensagem para o computador formada por Δx , Δy e botões.
 - Monitor: Gráfico de Varredura; Memória especial armazena bytes/bits; Controlador de Vídeo (retira caracteres ou bits da RAM de vídeo e gera sinais de vídeo para a saída); Cores geradas por palhetas ou combinação RGB.
- Software de E/S: Driver (conversão através de keymap) e Software de Saída (APIWin32, X System, Xfree).

31

Terminais de Rede

- Usado para conectar um usuário remoto ao computador por meio de uma rede.
- Dois paradigmas:
- Terminais com grande poder computacional
 - Protocolos de janelas X, Xlib e Widgets.
- Terminais simples e baratos
 - Timesharing (clientes leves).
- Considerações sobre sistemas centralizados e descentralizados.

32