

## SSC 140 - SISTEMAS OPERACIONAIS I

Turmas A e B

### Aula 18 – Gerenciamento de Dispositivos de Entrada/Saída (E/S)

Profa. Sarita Mazzini Bruschi

Slides de autoria de  
Luciana A. F. Martimiano baseados no livro  
*Sistemas Operacionais Modernos* de A. Tanenbaum

## Dispositivos de Entrada e Saída

### SO pode atuar de duas maneiras diferentes:

- Como máquina estendida (*top-down*) – tornar uma tarefa de baixo nível mais fácil de ser realizada pelo usuário;
- Como gerenciador de recursos (*bottom-up*) – gerenciar os dispositivos que compõem o computador;

2

## Dispositivos de Entrada e Saída

### Funções específicas:

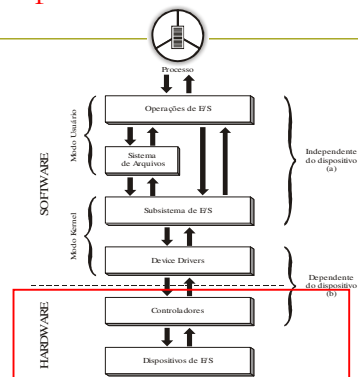
- Enviar sinais para os dispositivos;
- Atender interrupções;
- Gerenciar comandos aceitos e funcionalidades (serviços prestados);
- Tratar possíveis erros;
- Prover interface entre os dispositivos e o sistema;

### Princípios:

- Hardware;
- Software;

3

## Dispositivos de Entrada e Saída



4

## Dispositivos de E/S

### Princípios de Hardware

### Podem ser divididos em duas categorias:

- Dispositivos baseados em bloco: informação é armazenada em blocos de tamanho fixo, cada um com um endereço próprio;
  - Tamanho varia entre 512 bytes e 32.768 bytes;
  - Permitem leitura e escrita independentemente de outros dispositivos;
  - Permitem operações de busca;
  - Ex.: discos rígidos;

5

## Dispositivos de E/S

### Princípios de Hardware

- Dispositivos baseados em caracter: aceita uma sequência de caracteres, sem se importar com a estrutura de blocos; informação não é endereçável e não possuem operações de busca;
  - Ex.: impressoras, interfaces de rede (placas de rede); placas de som;

6

## Dispositivos de E/S

### Princípios de Hardware

- ❑ Classificação não é perfeita, pois alguns dispositivos não se encaixam em nenhuma das duas categorias:
  - *Clocks*: provocam interrupções em intervalos definidos;
- ❑ Classificação auxilia na obtenção de independência ao dispositivo:
  - Parte dependente está a cargo dos *drivers* → software que controla o acionamento dos dispositivos;

7

## Dispositivos de E/S

### Princípios de Hardware

- ❑ Dispositivos de E/S possuem basicamente dois componentes:
  - Mecânico → o dispositivo propriamente dito;
  - Eletrônico → controladores ou adaptadores (placas);
- ❑ O dispositivo (periférico) e a controladora se comunicam por meio de uma interface:
  - Serial ou paralela;
  - Barramentos: IDE, ISA, SCSI, AGP, USB, PCI, etc.

8

## Dispositivos de E/S

### Princípios de Hardware

- ❑ Cada controladora possui um conjunto de registradores de controle, que são utilizados na comunicação com a CPU;
- ❑ Além dos registradores, alguns dispositivos possuem um *buffer* de dados:
  - Ex.: placa de vídeo; algumas impressoras;
- ❑ SO gerencia, utilizando os *drivers*, os dispositivos de E/S escrevendo/lendo nos/dos registradores/*buffers*;
  - Comunicação em baixo nível – instruções em *Assembler*;
  - Enviar comandos para os dispositivos;
  - Saber o estado dos dispositivos;

9

## Dispositivos de E/S

### Princípios de Hardware

- ❑ Como a CPU se comunica com esses registradores de controle?
  - **Porta**: cada registrador de controle possui um número de porta (ou porto) de E/S de 8 ou 16 bits;
    - ❑ Instrução em *Assembler* para acessar os registradores;
    - ❑ Espaço de endereçamento diferente para a memória e para os dispositivos de E/S;
    - ❑ *Mainframes* IBM utilizavam esse método;
    - ❑ SOs atuais fazem uso dessa estratégia para a maioria dos dispositivos;

10

## Dispositivos de E/S

### Princípios de Hardware

- ❑ Comunicação com os registradores de controle:
  - **Memory-mapped (mapeada na memória)**: mapear os registradores de controle em espaços de memória;
    - ❑ Cada registrador possui um único endereço de memória;
    - ❑ Em geral, os endereços estão no topo da memória protegidos em endereços não utilizados por processos;
    - ❑ Uso de linguagem de alto nível, já que registradores são apenas variáveis na memória;
    - ❑ SOs utilizam essa estratégia para os dispositivos de vídeo;

11

## Dispositivos de E/S

### Princípios de Hardware

- ❑ Comunicação com os registradores de controle:
  - **Estratégia híbrida**:
    - ❑ Registradores → Porta;
    - ❑ *Buffers* → Memória;
      - Exemplo: Pentium - endereços de 640k a 1M para os *buffers* e as portas de E/S de 0 a 64k;

12

## Dispositivos de E/S

### Princípios de Hardware

- ❑ Como funciona a comunicação da CPU com os dispositivos?
  - Quando a CPU deseja ler uma palavra, ela coloca o endereço que ela está desejando no barramento de endereço e manda um comando **READ** no barramento de controle;
  - Essa comunicação pode ser controlada pela própria CPU ou pela DMA;

13

## Dispositivos de E/S

### Princípios de Hardware

- ❑ **DMA (Direct Access Memory)** → acesso direto à memória:
  - Presente principalmente em dispositivos baseados em bloco → discos;
    - ❑ Controladora integrada à controladora dos discos;
  - Pode estar na placa-mãe e servir vários dispositivos → controladora de DMA independente do dispositivo;
  - DMA tem acesso ao barramento do sistema independentemente da CPU;

14

## Dispositivos de E/S

### Princípios de Hardware

- ❑ DMA contém vários registradores que podem ser lidos e escritos pela CPU:
  - Registrador de endereço de memória;
  - Registrador contador de bytes;
  - Registrador (es) de controle;
    - ❑ Porta de E/S em uso;
    - ❑ Tipo da transferência (leitura ou escrita);
    - ❑ Unidade de transferência (byte ou palavra);
    - ❑ Número de bytes a ser transferido;

15

## Dispositivos de E/S

### Princípios de Hardware

- ❑ Sem DMA: Leitura de um bloco de dados em um disco:
  - Controladora do dispositivo lê bloco (bit a bit) a partir do endereço fornecido pela CPU;
  - Dados são armazenados no *buffer* da controladora do dispositivo;
  - Controladora do dispositivo checa consistência dos dados;
  - Controladora do dispositivo gera interrupção;
  - SO lê (em um *loop*) os dados do *buffer* da controladora do dispositivo e armazena no endereço de memória fornecido pela CPU;

16

## Dispositivos de E/S

### Princípios de Hardware

- ❑ Com DMA: Leitura de um bloco de dados em um disco: CPU controla
  - 1. Além do endereço a ser lido, a CPU fornece à controladora de DMA duas outras informações: endereço na RAM para onde transferir os dados e o número de bytes a ser transferido;
  - 2. Controladora de DMA envia dados para a controladora do dispositivo;
  - Controladora do dispositivo lê o bloco de dados e o armazena em seu *buffer*, verificando consistência;
  - 3. Controladora do dispositivo copia os dados para RAM no endereço especificado na DMA (modo direto);

17

## Dispositivos de E/S

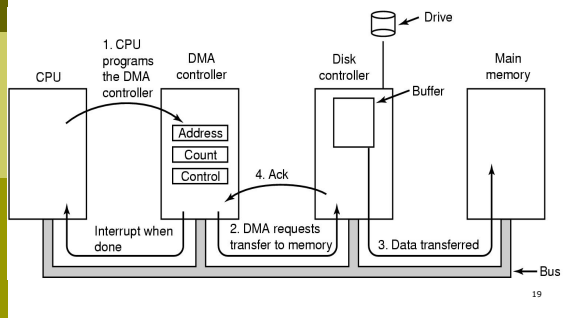
### Princípios de Hardware

- 4. Após confirmação de leitura, a controladora de DMA **incrementa o endereço de memória** na DMA e **decrementa o contador** da DMA com o número de bytes transferidos;
- Repete os passos de 2 a 4 até o contador da DMA chegar em 0. Assim que o contador chegar em zero (0), a controladora de DMA gera uma interrupção avisando a CPU;
- Quando o SO inicia o atendimento à interrupção, o bloco de dados já está na RAM;

18

## Dispositivos de E/S

### Princípios de Hardware



## Dispositivos de E/S

### Princípios de Hardware

- A DMA pode tratar múltiplas transferências simultaneamente:
  - Possuir vários conjuntos de registradores;
  - Decidir quais requisições devem ser atendidas → escalonamento (*Round-Robin* ou prioridades, por exemplo);

20

## Dispositivos de E/S

### Princípios de Hardware

- Por que a DMA não escreve diretamente na RAM?
  - Permite realizar consistência dos dados antes de iniciar alguma transferência;
  - Dados (bits) são transferidos do disco a uma taxa constante, independentemente da controladora estar pronta ou não;
  - Acesso à memória depende de acesso ao barramento, que pode estar ocupado com outra tarefa;
  - Com o *buffer*, o barramento é usado apenas quando a DMA opera;

21

## Dispositivos de E/S

### Princípios de Hardware

- Interrupções de E/S (*interrupt-driven I/O*):
  - Sinais de interrupção são enviados (através dos barramentos) pelos dispositivos ao processador;
  - Após uma interrupção, o **controlador de interrupções** decide o que fazer;
    - Envia para CPU;
    - Ignora no momento → dispositivos geram sinais de interrupção até serem atendidos;

22

## Dispositivos de E/S

### Princípios de Hardware

- **Controlador de Interrupções:**
  - Está presente na placa-mãe;
  - Possui vários manipuladores de interrupção;
  - Diferentes tipos de interrupções → IRQs (*Interrupt ReQuest*);
- **Manipuladores de interrupção:**
  - Gerenciam interrupções realizadas pelos dispositivos de E/S;
  - Bloqueiam *driver* até dispositivo terminar a tarefa;

23

## Dispositivos de E/S

### Tratando Interrupções

- Sinal (linha) de interrupção é amostrado dentro de cada ciclo de instrução do processador;
- Se sinal ativo → salva contexto e atende a interrupção;

24

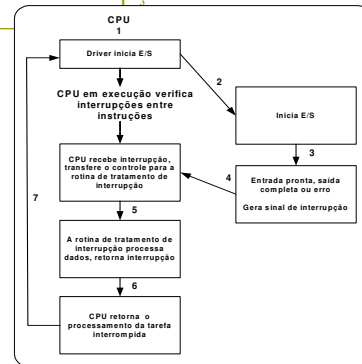
## Dispositivos de E/S Tratando Interrupções

### □ Ciclo de instrução com interrupção: CPU

- Busca; Decodificação e Execução
  - Verifica se existe interrupção
  - Se não → busca próxima instrução,...
  - Se existe interrupção pendente:
    - Suspende a execução do programa;
    - Salva contexto;
    - Atualiza PC (*Program Counter*) → apontar para ISR (rotina de atendimento de interrupção);
    - Executa interrupção;
    - Recarrega contexto e continua processo interrompido;

25

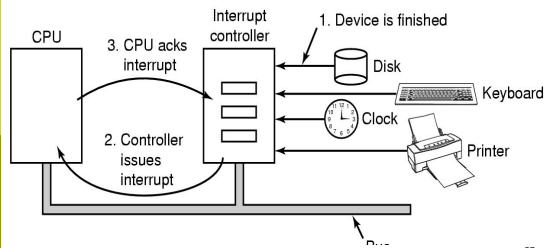
## Dispositivos de E/S Tratando Interrupções



26

## Dispositivos de E/S Princípios de Hardware

### □ Como uma interrupção ocorre:



27

## Dispositivos de E/S Tabela de Interrupções (exemplo)

IRQ	Uso padrão	Outras utilizações
00	Timer do sistema	Nenhum
01	Teclado	Nenhum
02	IRQs 8 a 15	Modem, placa de vídeo, porta serial (3, 4), IRQ 9
03	Porta serial 2	Modem, placa de som, placa de rede
04	Porta serial 1	Modem, placa de som, placa de rede
05	Placa de som (codec)	LPT2, COM 3 e 4, Modem, placa de rede, HDC
06	FDC	Placa aceleradora de fita
07	Porta paralela 1	COM 3 e 4, Modem, placa de som, placa de rede
08	Relógio de tempo real	Nenhum
09	Placa de som (midi)	Placa de rede, SCSI, PCI
10	Nenhum	Placa de rede, placa de som, SCSI, PCI, IDE 2
11	Placa de vídeo VGA	Placa de rede, placa de som, SCSI, PCI, IDE 3
12	Mouse P/S2	Placa de rede, placa de som, SCSI, PCI, IDE 3, VGA
13	FPU (Float Point Unit)	Nenhum
14	IDE primária	Adaptador SCSI
15	IDE secundária	Placa de Rede, adaptador SCSI

28

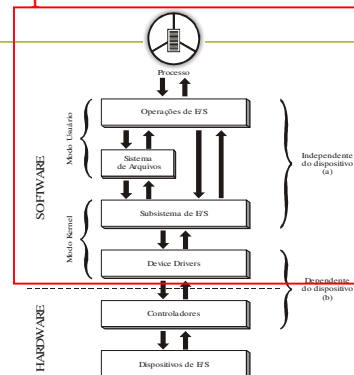
## Dispositivos de E/S Princípios de Software

### □ Organizar o software como uma série de camadas facilita a independência dos dispositivos:

- Camadas mais baixas apresentam detalhes de hardware:
  - Drivers e manipuladores de interrupção;
- Camadas mais altas apresentam interface para o usuário:
  - Aplicações de Usuário;
  - Chamadas de Sistemas;
  - Software Independente de E/S ou Subsistema de Kernel de E/S;

29

## Dispositivos de Entrada e Saída



30

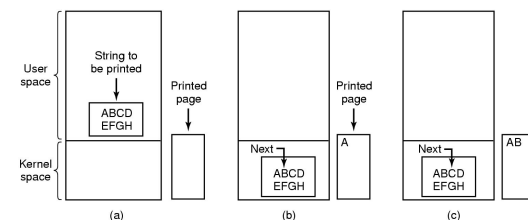
## Dispositivos de E/S Princípios de Software

- Maneiras de realizar E/S:
  - E/S programada;
    - Mais usada em sistemas embarcados/embutidos;
  - E/S orientada à interrupção;
  - E/S com uso da DMA;

31

## Dispositivos de E/S Princípios de Software

- E/S programada: passos para impressão de uma cadeia de caracteres (laço até que toda a cadeia tenha sido impressa);



## Dispositivos de E/S Princípios de Software

- E/S programada:
  - Desvantagem:
    - CPU é ocupada o tempo todo até que a E/S seja feita;
    - CPU continuamente verifica se o dispositivo está pronto para aceitar outro carácter → espera ocupada;

33

## Dispositivos de E/S Princípios de Software

- E/S orientada à interrupção:
  - No caso da impressão, a impressora não armazena os caracteres;
  - Quando a impressora está pronta para receber outros caracteres, gera uma interrupção;
  - Processo é bloqueado;

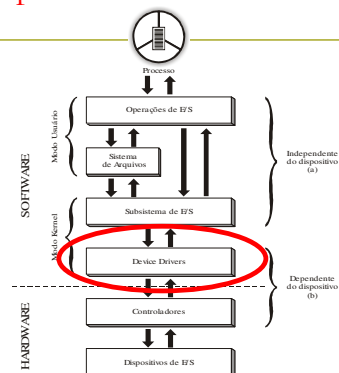
34

## Dispositivos de E/S Princípios de Software

- E/S com uso da DMA:
  - DMA executa E/S programada → controladora de DMA faz todo o trabalho ao invés da CPU;
    - Redução do número de interrupções;
  - Desvantagem:
    - DMA é mais lenta que a CPU;

35

## Dispositivos de Entrada e Saída



36

## Dispositivos de E/S

### Princípios de Software - Camadas

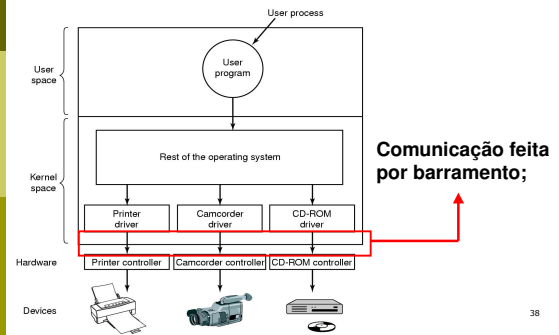
#### ❑ Drivers:

- São gerenciados pelo *kernel* do SO;
- Contêm todo o código dependente do dispositivo;
- Controlam o funcionamento dos dispositivos por meio de sequência de comandos escritos/lidos nos/dos registradores da controladora;
- Dispositivos diferentes possuem *drivers* diferentes;
  - ❑ Classes de dispositivos podem ter o mesmo *driver*;
- São dinamicamente carregados;
- *Drivers* defeituosos podem causar problemas no *kernel* do SO;

37

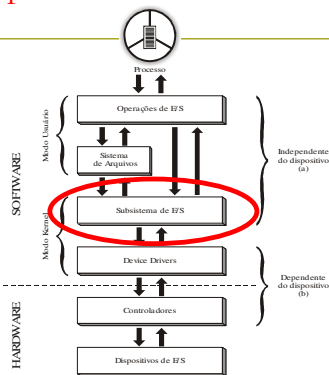
## Dispositivos de E/S

### Princípios de Software - Camadas



38

## Dispositivos de Entrada e Saída



39

## Dispositivos de E/S

### Princípios de Software - Camadas

#### ❑ Software de E/S no nível Usuário:

- Bibliotecas de E/S são utilizadas pelos programas dos usuários
  - ❑ Chamadas ao sistema (*system calls*);

40

## Dispositivos de E/S

### Princípios de Software - Camadas

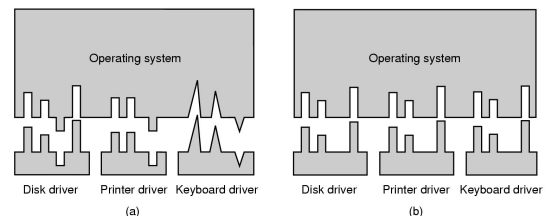
#### ❑ Software Independente de E/S:

- Realizar as funções comuns a qualquer dispositivos;
- Prover uma interface uniforme para os drivers dos dispositivos
  - ❑ Número de procedimentos que o restante do SO pode utilizar para fazer o driver trabalhar para ele;
- Fazer o escalonamento de E/S;
- Atribuir um nome lógico a partir do qual o dispositivo é identificado;
  - ❑ Ex.: UNIX → (/dev)
- Prover *buffering*: ajuste entre a velocidade e a quantidade de dados transferidos;
- *Cache* de dados: armazenar na memória um conjunto de dados freqüentemente acessados;

41

## Dispositivos de E/S

### Princípios de Software - Camadas



(a) Sem padrão de interface

(b) Com padrão de interface (uniforme)

42

## Dispositivos de E/S

### Princípios de Software - Camadas

#### ■ Software Independente de E/S:

- Reportar erros e proteger os dispositivos contra acessos indevidos :
  - Programação: Ex.: tentar efetuar leitura de um dispositivo de saída (impressora, vídeo);
  - E/S: Ex.: tentar imprimir em uma impressora desligada ou sem papel;
  - Memória: escrita em endereços inválidos;
- Gerenciar alocação, uso e liberação dos dispositivos → acessos concorrentes;

43

## Dispositivos de E/S

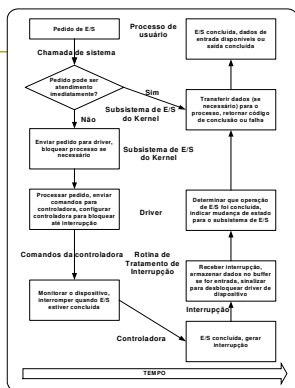
### Princípios de Software

#### ■ Software Independente de E/S:

- Transferência de dados:
  - Síncrona (bloqueante): requer bloqueio até que os dados estejam prontos para transferência;
  - Assíncrona (não-bloqueante): transferências acionadas por interrupções; mais comuns;
- Tipos de dispositivos:
  - Compartilháveis: podem ser utilizados por vários usuários ao mesmo tempo; Exemplo: disco rígido;
  - Dedicados: podem ser utilizados por apenas um usuário de cada vez; Exemplo: impressora, unidade de fita;

44

## Dispositivos de E/S - Ciclo de E/S



45

## Dispositivos de E/S - Ciclo de E/S

### Seqüência da Figura anterior

- Um processo emite uma chamada de sistema bloqueante (por exemplo: `read`) para um arquivo que já esteve aberto (`open`);
- O código da chamada de sistema verifica os parâmetros. Se os parâmetros estiverem corretos e o arquivo já estiver no *buffer* (*cache*), os dados retornam ao processo e a E/S é concluída;
- Se os parâmetros estiverem corretos, mas o arquivo não estiver no *buffer*, a E/S precisa ser realizada;
  - E/S é escalonada;
  - Subsistema envia pedido para o *driver*;

46

## Dispositivos de E/S - Ciclo de E/S

### Seqüência da Figura anterior

- *Driver* aloca espaço de *buffer*, escalona E/S e envia comando para a controladora do dispositivo escrevendo nos seus registradores de controle;
  - *Driver* pode usar a DMA;
- A controladora do dispositivo opera o hardware, ou seja, o dispositivo propriamente dito;
- Após a conclusão da E/S, uma interrupção é gerada;
- A rotina de tratamento de interrupções apropriada recebe a interrupção via vetor de interrupção, armazena os dados, sinaliza o *driver* e retorna da interrupção;

47

## Dispositivos de E/S - Ciclo de E/S

### Seqüência da Figura anterior

- *Driver* recebe o sinal, determina qual pedido de E/S foi concluído, determina o status e sinaliza que o pedido está concluído;
- Kernel transfere dados ou códigos de retorno para o espaço de endereçamento do processo que requisitou a E/S e move o processo da fila de bloqueados para a fila de prontos;
- Quando o escalonador escalona o processo para a CPU, ele retoma a execução na conclusão da chamada ao sistema.

48