



Laboratório de Pesquisa em Redes e Multimídia

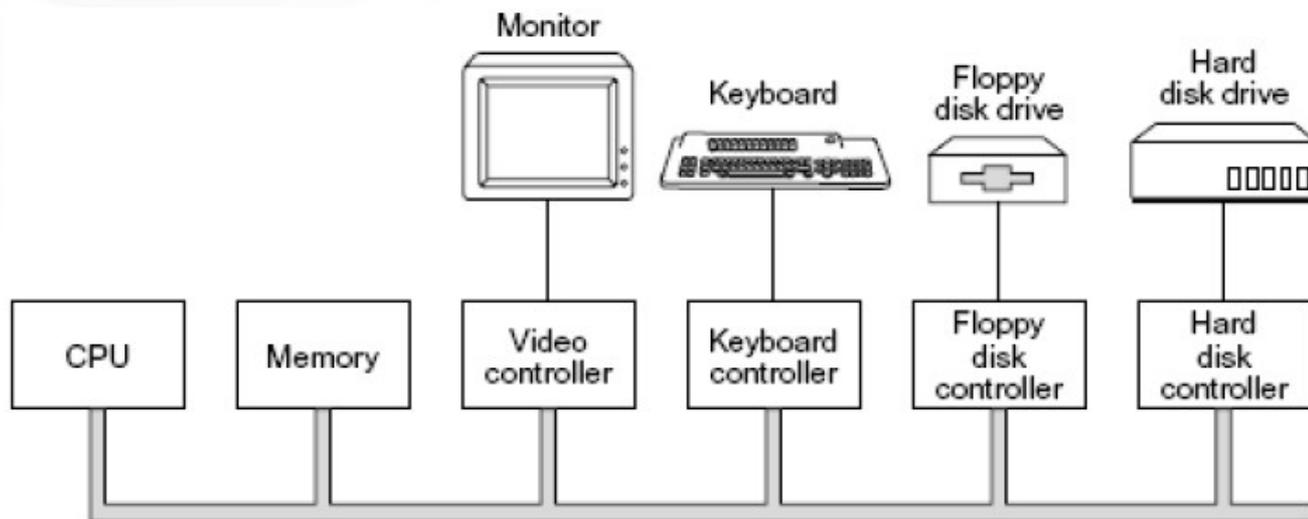
# Gerência de Entrada e Saída



Universidade Federal do Espírito Santo  
Departamento de Informática

## Dispositivos de Entrada e Saída (1)

- Constituídos de 2 partes:
  - Mecânica
  - Eletrônica - Controladora ou Adaptadora
- Controladora
  - Placa ligada a um slot livre, ou inserida diretamente na placa-mãe



## Dispositivos de Entrada e Saída (2)

- Controladora (cont.)
  - Implementa um conjunto de funções básicas para o dispositivo
  - Também tratar o acesso do dispositivo ao barramento
  - Tarefas típicas
    - Converter fluxo serial de bits em bloco de bytes
    - Correção de erros
    - Tornar o bloco disponível para ser copiado para a memória principal
  - Possui registradores usados para comunicar com o SO
    - Seqüência e/ou valores armazenados nestes registradores determina a operação sendo realizada
  - Tipicamente tem memória interna (buffer)
    - Compatibilizar velocidades

## Princípios Básicos de Hardware

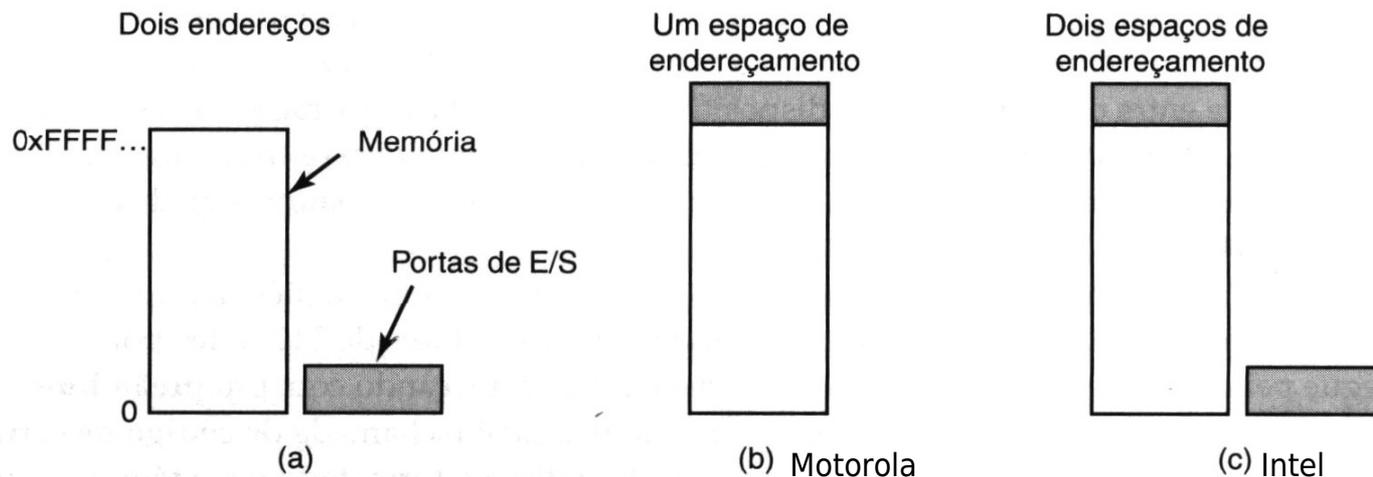
- Tipos de conexão e transferência de dados
  - Serial versus paralela
  - Barramentos
- Técnicas para realização de E/S
  - E/S programada, E/S orientada a interrupções, acesso direto a memória
- Mapeamento de endereços
  - Em espaço de E/S e em espaço em memória
- Controladoras

## Tipos de interfaces (dados)

- Define o modo como os dados são transferidos:
  - Serial - bits transmitidos seqüencialmente em uma linha de dados
  - Paralela - vários bits transmitidos simultaneamente (múltiplo de 8)
- Barramentos:
  - Efetua a ligação com o dispositivo de E/S
    - Porta Serial, Porta Paralela, USB, PS/2
    - PCI, AGP, PCI-E, SCSI, IDE, SATA,

# Mapeamento de Endereços

- Mapeamento em espaço de entrada e saída
  - Instruções especiais da CPU para E/S
    - Opcodes separados (IN reg, [end16], OUT [end16], reg)
  - Cada registrador está associado a uma porta de E/S
- Mapeamento em espaço de memória
  - Associa um espaço de endereços de memória aos registradores
  - Leitura/Escrita neste espaço realizam a operação sobre os registradores



## Técnicas para realização de E/S (1)

- Determina a forma de interação entre dispositivos e processador
- Três técnicas usadas:
  - E/S Programada
  - Interrupção
  - Acesso Direto à Memória.

## Técnicas para realização de E/S (2)

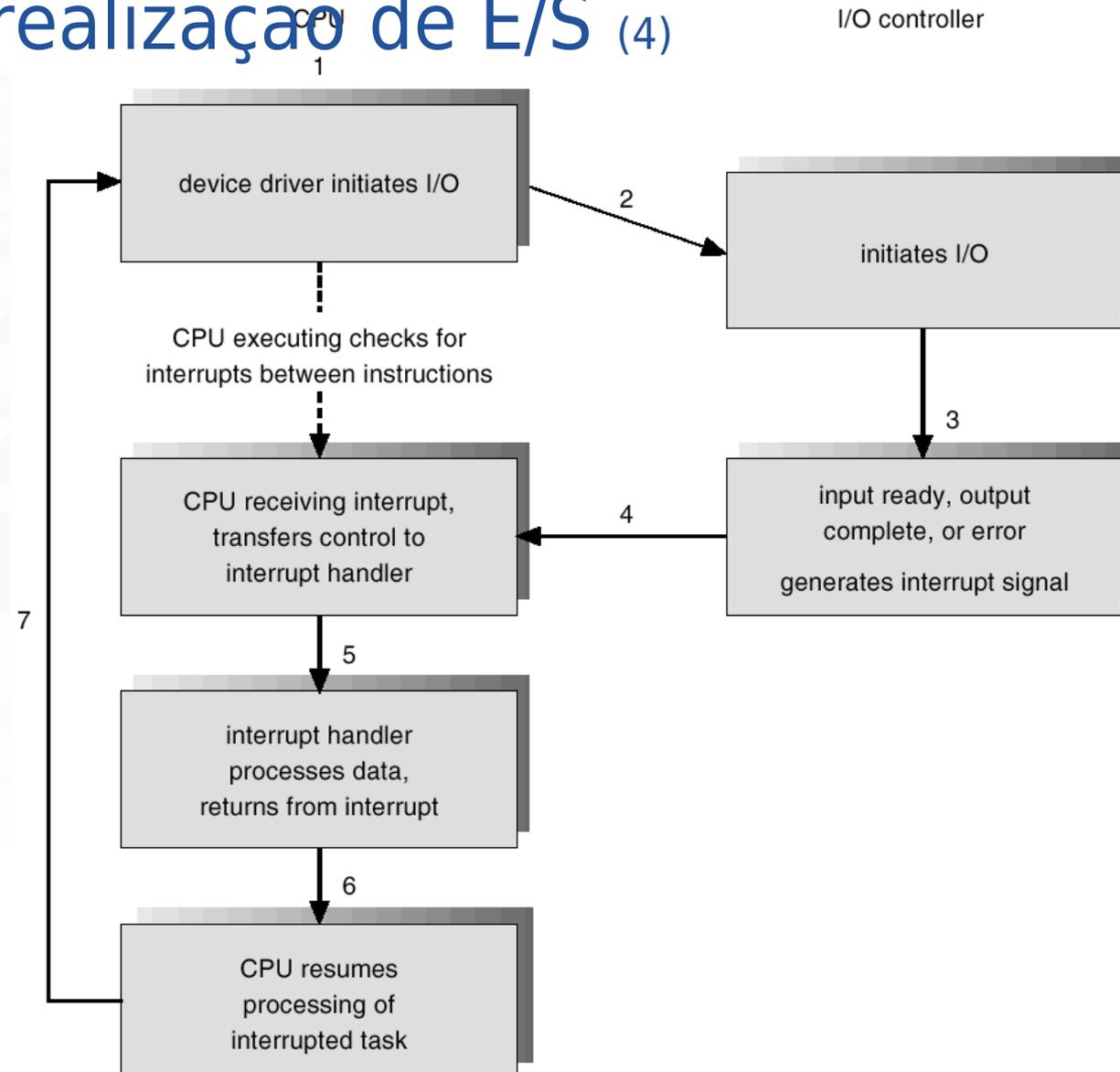
- E/S Programada
  - Interação com o dispositivo é responsabilidade do programador
  - Ciclo de funcionamento
    - Envio do comando ao dispositivo
    - Espera pela realização do comando
  - A CPU deve aguardar o término da operação
    - Controladora atualiza bits nos seus registradores de status
  - Processador espera término da operação
  - Desvantagem
    - Desperdício do processador (verificando estado da operação)
    - Solução: Inserir operações entre operações (*polling*) ... mas em que frequência?

## Técnicas para realização de E/S (3)

- E/S Orientada a Interrupção
  - O processador é apenas responsável por iniciar uma operação de E/S
  - O dispositivo interrompe a CPU quando necessário ou quando a operação de E/S terminar
  - Requer hardware especial:
    - Controlador de interrupções:
      - Identifica o dispositivo que gerou a interrupção
      - Define a prioridade das interrupções
      - Define quais serão atendidas (mascaramento)

# Técnicas para realização de E/S (4)

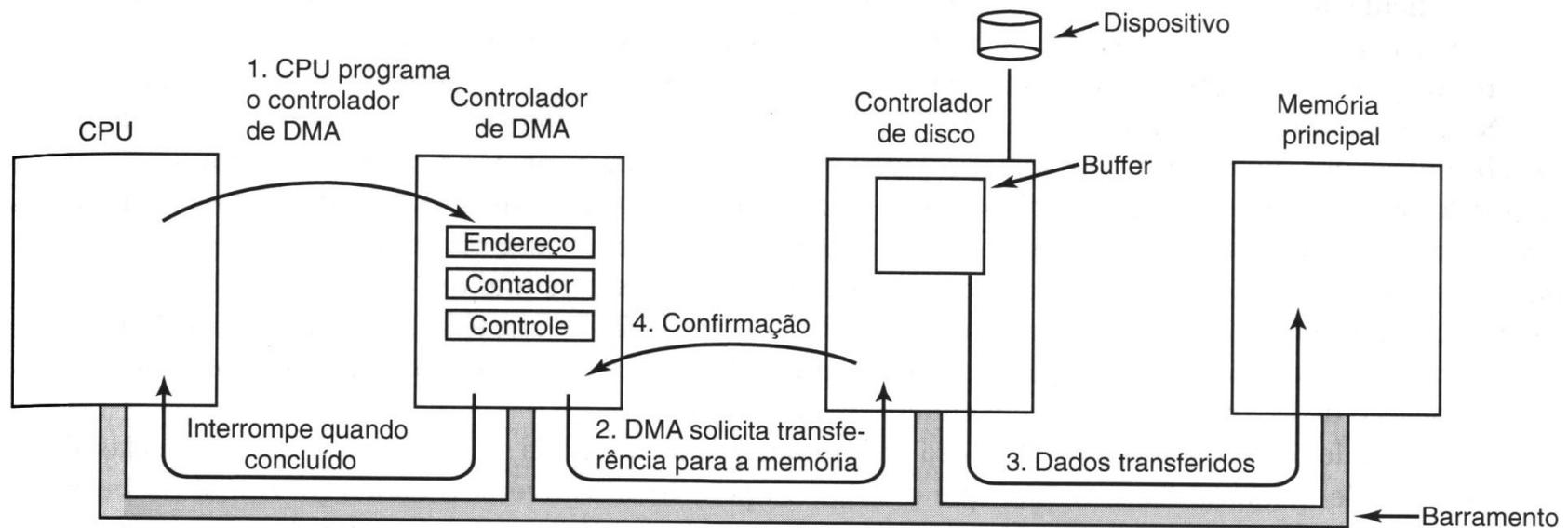
- E/S Orientada a Interrupção (cont.)



## Técnicas para realização de E/S (5)

- Acesso direto à memória (DMA)
  - Em grandes quantidades de dados, gasta-se tempo do processador transferindo dados entre a memória e o dispositivo
- Controlador de DMA:
  - Realiza a transferência sem o processador
  - O processador fornece a posição inicial, quantidade, origem e destino ao controlador de DMA
  - O processador é liberado para outras tarefas
  - O controlador compartilha o barramento e realiza a transferência
  - Terminando a transferência, o controlador gera uma interrupção

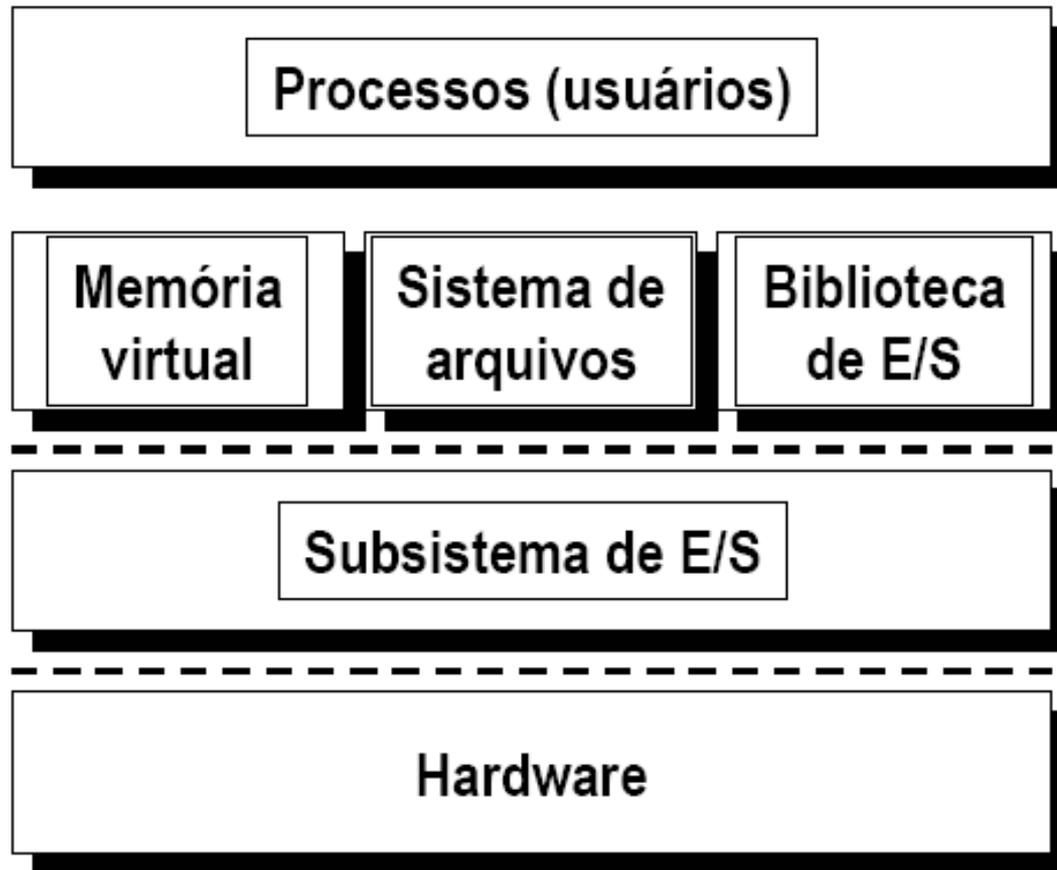
# Técnicas para realização de E/S (5)



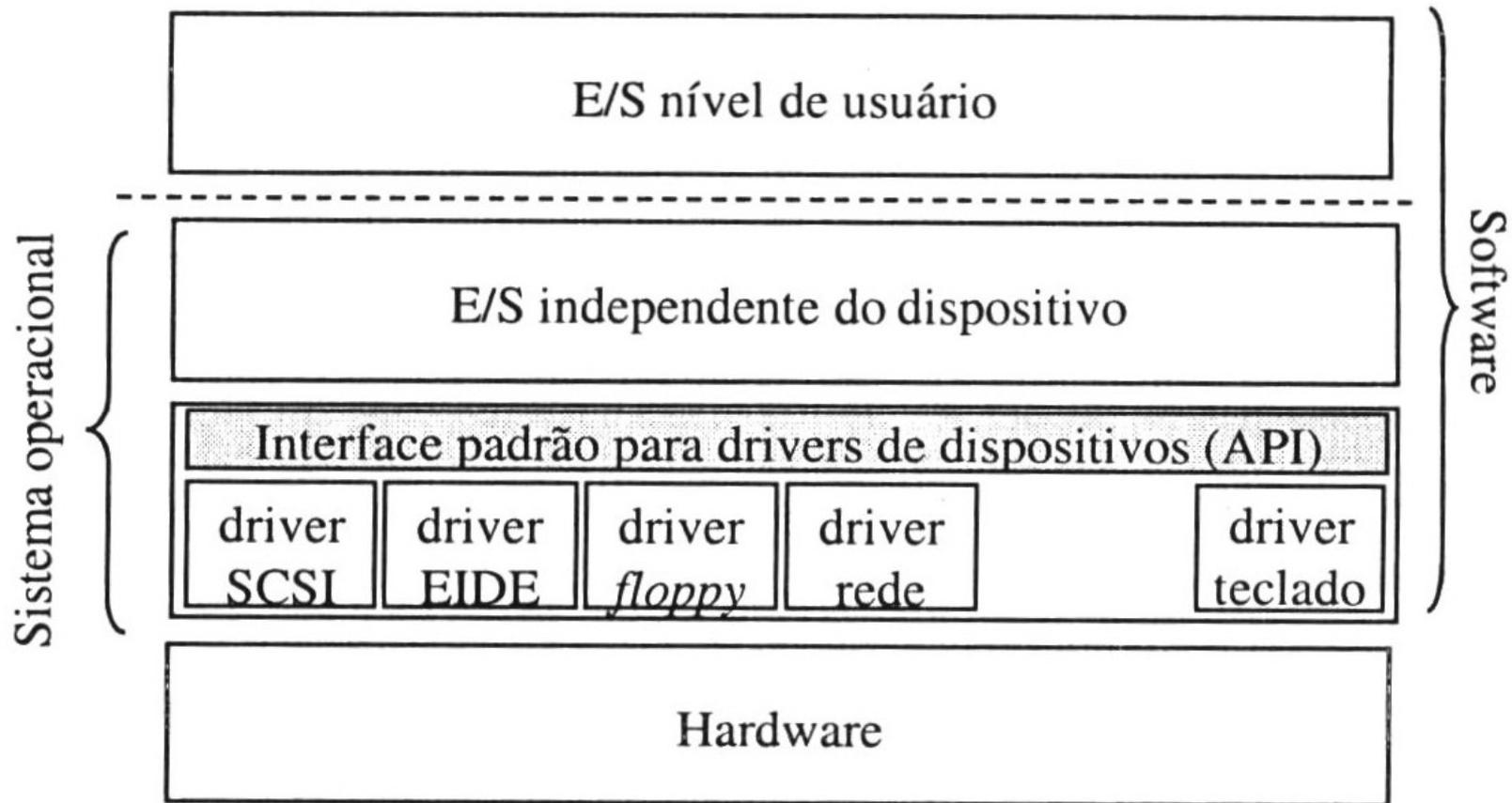
# Subsistema (Software) de Entrada e Saída (1)

- Objetivos
  - Fornecer uma interface “amigável” para utilização dos dispositivos de E/S
    - Uniformizar o tratamento dos dispositivos
    - “Esconder” detalhes de mais “baixo nível”
  - Permitir a inclusão de novos dispositivos
  - Facilitar a correção de erros gerados pelo dispositivo
  - Explorar eficientemente os dispositivos de E/S
    - Desempenho
  - Compartilhamento dos dispositivos de E/S
  - Alocação dos dispositivos
  - Escalonamento de requisições

# Subsistema (Software) de Entrada e Saída (2)



# Subsistema (Software) de Entrada e Saída (3)



## Subsistema (Software) de Entrada e Saída (4)

- Software de E/S de usuário:
  - Realiza chamadas de alto nível ao software de E/S independente de dispositivo:
  - Wrappers para chamadas de sistema
    - Abrir/fechar arquivo, ler/escrever dados, etc.

## Subsistema (Software) de Entrada e Saída (5)

- Interface do subsistema de E/S
  - Dispositivos “abstratos” de E/S
  - Cada um representa uma classe de dispositivos de E/S
- Dispositivos “abstratos”
  - Orientado a bloco (block device)
  - Orientado a caractere (stream, character device)
  - Rede

## Subsistema (Software) de Entrada e Saída (6)

- Interface do subsistema de E/S (cont.)
  - Dispositivos Orientado a bloco
    - Organiza dados em blocos de tamanho fixo
    - Acessa diretamente um bloco de dados
    - Blocos são identificados por endereços (número do bloco)
      - ex: blocos de disco
    - Operações típicas:
      - `open( )`, `read( )`, `write( )` e `close( )`
      - Disponibilizadas aos usuários via sistema de arquivos

## Subsistema (Software) de Entrada e Saída (7)

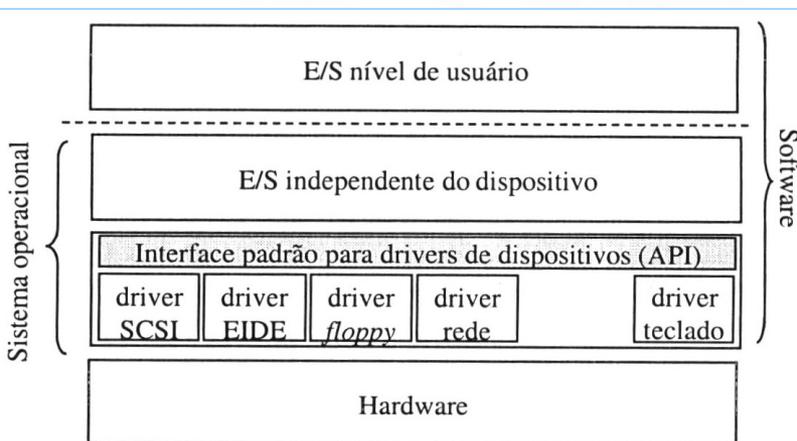
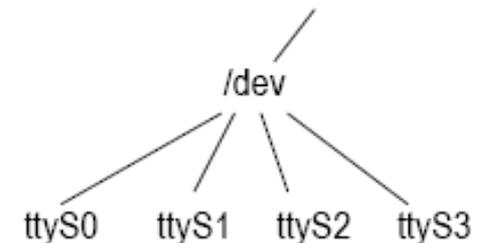
- Interface do subsistema de E/S (cont.)
  - Dispositivos Orientado a caractere
    - Controla dispositivos que produzem ou consomem conjunto de dados de tamanho arbitrário
    - Operações típicas:
      - `put( )` e `get( )`
      - ex: teclado, vídeo, mouse, impressora, etc...
    - Operações específicas (inversão de cores, bip, inicialização, etc) são fornecidas por uma função genérica
      - `io_control( )`

# Subsistema (Software) de Entrada e Saída (8)

- Interface do subsistema de E/S (cont.)
  - Dispositivos Orientado a rede
    - Controle e acesso a dispositivos que estão fisicamente instalados em outros equipamentos
    - Necessário estabelecimento de conexões
    - Interface típica são sockets
    - Operações típicas:
      - `open( )`, `close( )`, `create( )`
    - Suporte a serviços:
      - orientado a conexão: `connect( )`, `accept( )`, `read( )`, `write( )`
      - sem conexão: `send( )` e `recv( )`

## Subsistema (Software) de Entrada e Saída (9)

- Software de E/S independente de dispositivo
  - Implementa funções gerais comuns a todos os dispositivos
  - Atribuição uniforme do nome independente do dispositivo
    - O UNIX é um exemplo clássico:
      - Nome do dispositivo é um string



## Subsistema (Software) de Entrada e Saída (10)

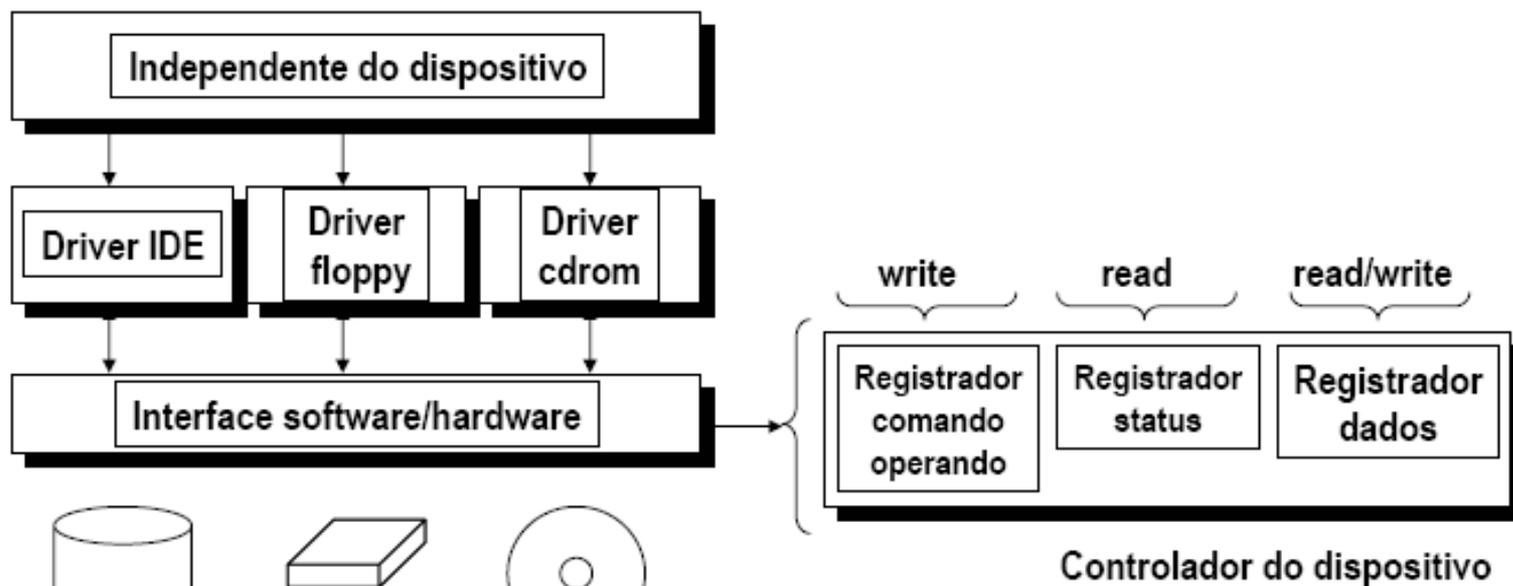
- Software de E/S independente de dispositivo
  - Implementa os serviços de:
    - **Escalonamento** de E/S: reordena as requisições para melhorar desempenho
    - **Identificação** (denominação) : nome lógico no SO
    - **Bufferização**: área de memória para armazenamento temporário de dados. Ajusta velocidade entre as camadas
    - **Cache** de dados: área de memória para armazenar dados usados com mais frequência, melhorando desempenho
    - Direitos de Acesso
    - Tratamento de erros

# Subsistema (Software) de Entrada e Saída (11)

- Software de E/S independente de dispositivo (cont.)
  - Buffer:
    - Área de armazenamento temporário de dados
    - Por que usar?
      - Ajustar velocidades entre produtores e consumidores
        - e.g. transmissão de dados via conexão de baixa velocidade
      - Ajustar unidades de transferência de dados de tamanho diferentes
        - e.g. pilha de protocolo de redes
  - Cache
    - Importante: *buffer* não é sinônimo de cache
      - Embora as caches sejam um tipo de buffer
    - O objetivo de uso de cache é desempenho

# Subsistema (Software) de Entrada e Saída (12)

- Drivers de dispositivo
  - Receber comandos da camada superior (“independente de dispositivo”) e interagir com os dispositivos
  - Implementa a interface padrão como sequência de acesso aos registradores



## Subsistema (Software) de Entrada e Saída (12)

- Drivers de dispositivo (cont.)
  - Em geral, devem rodar no modo kernel
  - Desenvolvidos pelo fabricante do dispositivo
  - Dispositivos mais comuns podem ter o driver incluído no Sistema Operacional
  - Vantagens
    - Isolar o código específico a um dispositivo em um módulo aparte
      - Fabricantes ão precisam mexer no kernel
    - Facilidade de adicionar novos drivers

## Entrada e Saída no Unix

- O Unix mapeia os dispositivos de entrada e saída em arquivos especiais
- Dispositivos localizados no diretório /dev
- No caso de discos, um número inteiro concatenado ao dispositivo indica a partição
  - /dev/sda1 - Primeira partição do dispositivo mestre do barramento sata 1
- O suporte a drives SCSI é usado (com pequenas adaptações) para dar suporte a outros tipos de dispositivos
  - um HD SATA e um pendrive, instalados na mesma máquina
    - o HD será visto como "/dev/sda" (pois é inicializado primeiro, logo no início do boot) e o pendrive como "/dev/sdb"
  -

## Referências

- Jonathan Corbet, Alessandro Rubini, and Greg Kroah-Hartman “LINUX DEVICE DRIVERS”, 3rd Edition, editora O'REILLY
  - <http://lwn.net/images/pdf/LDD3/>