

Entrada e Saída (E/S)

Input/Output (I/O)

Sistemas Operacionais

Prof. Pedro Ramos
pramos.costar@gmail.com

ÚLTIMA AULA

Armazenamento em massa, programação de disco

- Minimização do tempo de busca e da **latência de rotação**
- Programação de cabeçotes de disco
 - FCFS, SSTF, SCAN, C-SCAN
- Tipos de armazenamento em massa
 - HDDs, SSDs, armazenamento terciário, RAID

ENTRADA E SAÍDA

- Como o **hardware** de E/S influencia o sistema operacional?
- Quais **serviços** de E/S o sistema operacional oferece?
- Como o sistema operacional **implementa** esses serviços?
- Como o sistema operacional pode **melhorar** o desempenho de E/S?

Arquitetura de sistemas de E/S

- Principais componentes

- **Barramento do sistema:** permite que o dispositivo se comunique com a CPU, normalmente compartilhado por vários dispositivos.

- Uma porta de dispositivo que normalmente consiste em 4 registradores:

- 0 **status** indica um dispositivo ocupado, dados prontos ou condição de erro

- **Controle**: comando para executar

- **Entrada** de dados: dados sendo enviados do dispositivo para a CPU

- **Saída** de dados: dados sendo enviados da CPU para o dispositivo

Arquitetura de sistemas de E/S

– **Controlador:** recebe comandos do barramento do sistema, converte-os em ações do dispositivo e lê/grava dados no barramento do sistema.

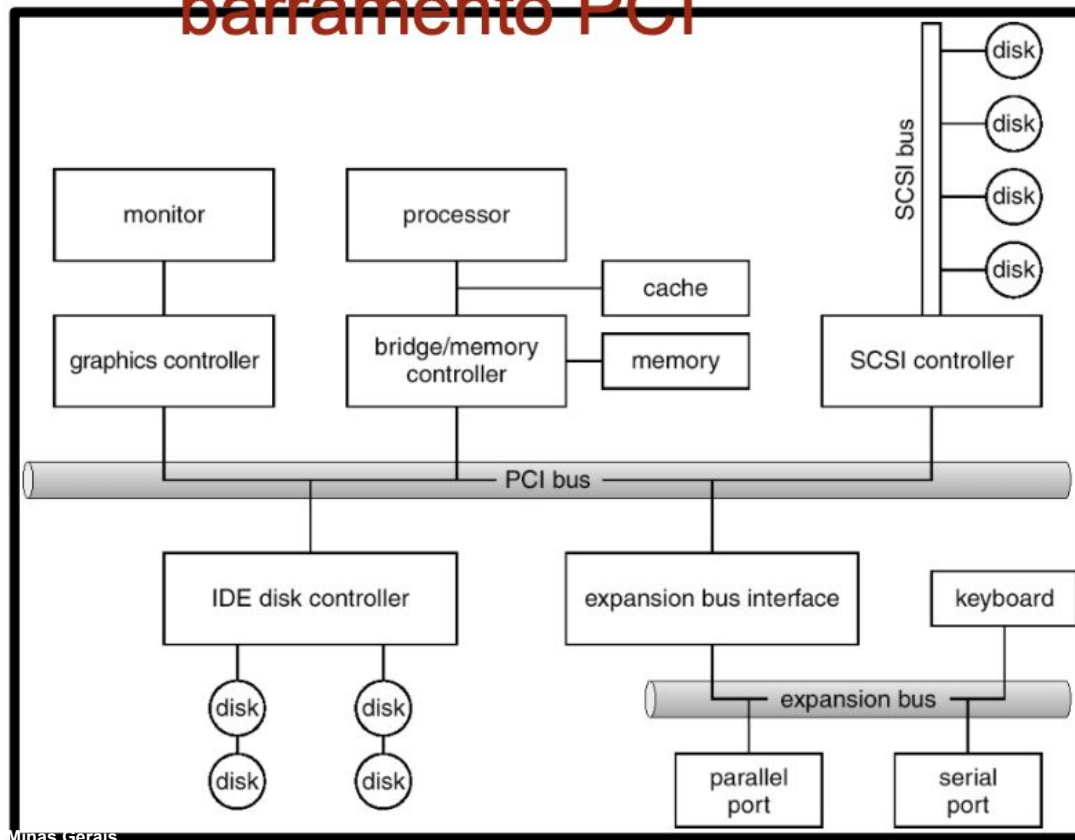
– **O próprio dispositivo**

- **Dispositivos tradicionais:** unidade de disco, impressora, teclado, modem, mouse, monitor

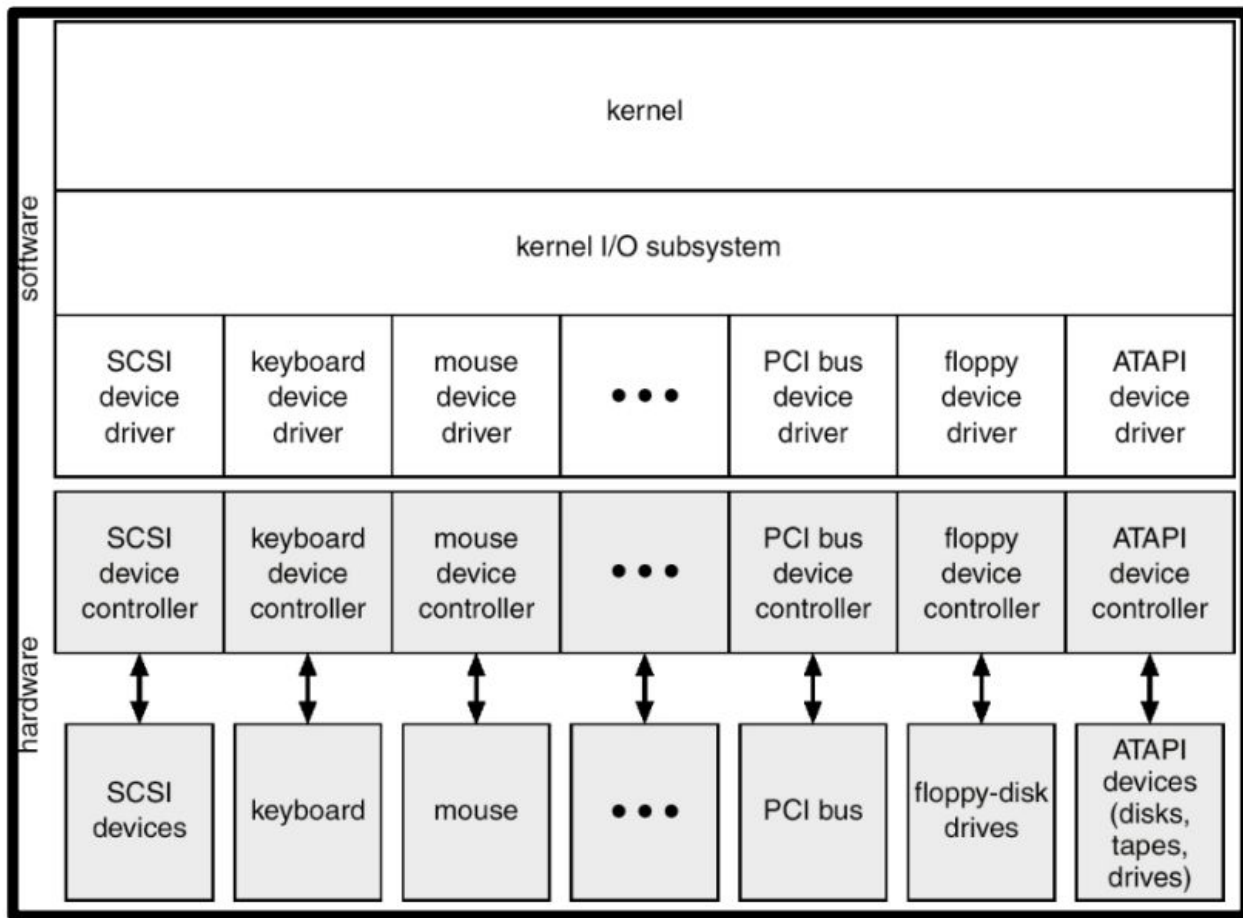
- **Dispositivos não tradicionais:** joystick, atuadores de robôs, superfícies de voo de um avião, sistema de injeção de combustível de um carro, ...

Estrutura do barramento PCI

PCI: INTERCONEXÃO DE COMPONENTES PERIFÉRICOS



Minas Gerais



I/O address range (hexadecimal)	device
000-00F	DMA controller
020-021	interrupt controller
040-043	timer
200-20F	game controller
2F8-2FF	serial port (secondary)
320-32F	hard-disk controller
378-37F	parallel port
3D0-3DF	graphics controller
3F0-3F7	diskette-drive controller
3F8-3FF	serial port (primary)

nao confundir
com portas
de rede!

Serviços de E/S fornecidos pelo sistema operacional

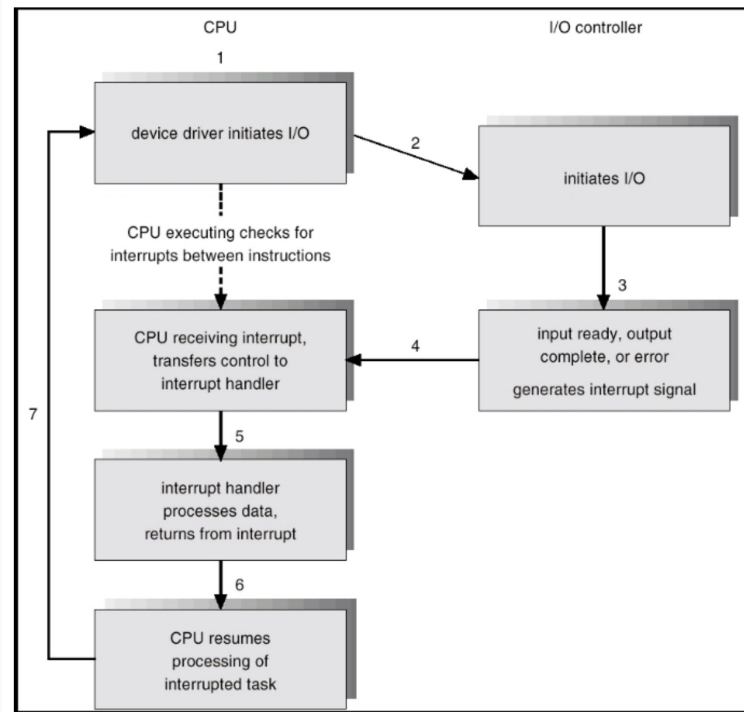
- Nomeação de arquivos e dispositivos. (No Unix, os dispositivos aparecem como arquivos no diretório `/dev`)
- **Controle de acesso**.
- Operações apropriadas para os arquivos e dispositivos.
- **Alocação** de dispositivos.
- **Buffering** e **cache** para permitir uma comunicação eficiente com os dispositivos.
- Escalonamento de E/S.
- Tratamento de erros e recuperação de falhas associadas aos dispositivos (novas tentativas de comando, por exemplo).
- **Drivers** de dispositivo para implementar **comportamentos específicos** do dispositivo

Comunicação usando Polling

- A CPU fica em **espera** até que o status esteja ocioso.
- A CPU define o registrador de **comando** e o **data-out** se for uma operação de saída.
- A **CPU** define o status como **pronto** para comando
=> O **controlador** define o status como **ocupado**.
- O controlador lê o registrador de comando e executa o comando, colocando um valor no **data-in** se for um comando de entrada.
- Se a operação for bem-sucedida, o controlador mudará o status para **inativo**.
- A CPU observa a mudança para o modo inativo e ***lê os dados se for uma operação de entrada.***
- Boa opção se os dados precisarem ser tratados prontamente, como em um **modem** ou **teclado**.
- O que acontece se o dispositivo for lento em comparação com a CPU?

Comunicação usando interrupções

- Em vez de usar a espera ocupada, o dispositivo pode interromper a CPU quando concluir uma operação de E/S.
- Em uma interrupção de E/S:
 - Determine **qual dispositivo** causou a interrupção.
 - Se o **último comando** foi uma **operação de entrada**, recupere os **dados** do registrador do dispositivo.
 - **Iniciar a próxima** operação para esse dispositivo.



Vetores de eventos Intel x86

vector number	description
0	divide error
1	debug exception
2	null interrupt
3	breakpoint
4	INTO-detected overflow
5	bound range exception
6	invalid opcode
7	device not available
8	double fault
9	coprocessor segment overrun (reserved)
10	invalid task state segment
11	segment not present
12	stack fault
13	general protection
14	page fault
15	(Intel reserved, do not use)
16	floating-point error
17	alignment check
18	machine check
19-31	(Intel reserved, do not use)
32-255	maskable interrupts

Acesso direto à memória

- Para dispositivos que transferem grandes volumes de dados de uma só vez (como um bloco de disco), é caro fazer com que a CPU recupere um byte de cada vez.

- **Solução: Acesso direto à memória (DMA)**

- Use um controlador de DMA sofisticado que possa **gravar diretamente na memória**. Em vez de registradores de entrada e saída de dados, ele tem um registrador de endereço.

- A **CPU informa** ao DMA os locais da **origem** e do **destino** da transferência.

- O controlador de **DMA opera** o **barramento** e **interrompe** a **CPU** quando toda a transferência é concluída, em vez de quando cada byte está pronto.

- O controlador de **DMA** e a **CPU competem** pelo **barramento** de memória, tornando a CPU um pouco mais lenta, mas ainda assim proporcionando um desempenho melhor do que se a CPU tivesse que fazer a transferência sozinha.

Visão do programador de aplicativos dos dispositivos de E/S

- O sistema operacional fornece uma interface de alto nível para os dispositivos, simplificando muito o trabalho do programador.
 - São fornecidas interfaces padrão para dispositivos relacionados.
 - As dependências de dispositivos são encapsuladas em drivers de dispositivos.
 - Novos dispositivos podem ser compatíveis com o fornecimento de um novo driver de dispositivo.

Visão do programador de aplicativos dos dispositivos de E/S

- **Características do dispositivo:**

- Unidade de transferência: **caractere** ou **bloco**
- Método de acesso: acesso **sequencial** ou **aleatório**
- Temporização: **síncrona** ou **assíncrona**
 - A maioria dos dispositivos é assíncrona, enquanto as chamadas de sistema de E/S são síncronas. => O sistema operacional implementa E/S de bloqueio
- **Compartilhável** ou **dedicado**
- **Velocidade**
- Operações: **Entrada**, **saída** ou **ambas**

Visão do programador de aplicativos dos dispositivos de E/S

aspect	variation	example
data-transfer mode	character block	terminal disk
access method	sequential random	modem CD-ROM
transfer schedule	synchronous asynchronous	tape keyboard
sharing	dedicated sharable	tape keyboard
device speed	latency seek time transfer rate delay between operations	
I/O direction	read only write only read&write	CD-ROM graphics controller disk

Dispositivos de blocos e caracteres

- Os dispositivos de bloco incluem unidades de disco:
 - Os comandos incluem leitura, gravação, busca
 - E/S bruta ou acesso ao sistema de arquivos
 - Possibilidade de acesso a arquivos com mapeamento de memória
- Os dispositivos de caracteres incluem teclados, mouses, portas seriais
 - Os comandos incluem get, put
 - Bibliotecas em camadas na parte superior para maior funcionalidade (por exemplo, edição de linha, backspace)

Buffer de E/S

Os dispositivos de E/S normalmente **contêm uma pequena memória integrada na qual podem armazenar dados temporariamente** antes de transferí-los de/para a CPU.

- Um **buffer de disco** armazena um bloco quando ele é lido do disco.
- Ele é transferido pelo barramento pelo controlador de DMA para um **buffer na memória física**.
- O controlador de DMA interrompe a CPU quando a transferência é concluída.

Por que o buffer no lado do sistema operacional?

- Para lidar com as **incompatibilidades** de **velocidade** entre os **dispositivos**.

- Exemplo: Receber arquivo em uma rede (lento) e armazenar em disco (mais rápido)

- Para **lidar** com dispositivos que têm **diferentes tamanhos** de transferência de dados.

- Exemplo: o ftp traz o arquivo pela rede, um pacote de cada vez. O armazenamento em disco ocorre em um bloco de cada vez.

- Para **minimizar o tempo** em que um processo de usuário fica **bloqueado** em uma gravação.

- *Gravações => copiam dados para um buffer do kernel e retornam o controle para o programa do usuário. A gravação do buffer do kernel no disco é feita posteriormente.*

CACHE

- Melhore o desempenho do disco reduzindo o número de acessos ao disco.

- Ideia: manter os blocos de disco usados recentemente na memória principal após a conclusão da chamada de E/S que os trouxe para a memória.

- Exemplo: Read (diskAddress)

- If (block in memory)

- return value from memory

- Else

- return ReadSector(diskAddress)

- Exemplo: Write (diskAddress)

- If (bloco na memória)

- Atualizar o valor na memória

- Else

- Alocar espaço na memória, ler o bloco do disco e atualizar o valor na memória

CACHE

– Exemplo: Write (diskAddress)

If (bloco na memória)

 Atualizar o valor na memória

Else

 Alocar espaço na memória, ler o bloco do disco e atualizar o valor na memória

• O que deve acontecer quando gravamos em um cache?

– **política de write-through** (gravação em todos os níveis de memória que contêm o bloco, inclusive no disco). Alta confiabilidade.

– **política de write-back** (gravar somente na memória mais rápida que contém o bloco, gravar nas memórias mais lentas e no disco algum tempo depois). Mais rápido.

JUNTANDO TUDO - UMA CHAMADA TÍPICA PARA READ()

1. O processo do usuário **solicita uma leitura** de um dispositivo.
2. O **sistema operacional verifica se os dados estão em um buffer**. Se não estiverem,
 - a) O sistema operacional **informa ao driver do dispositivo para executar a entrada**.
 - b) O **driver** do dispositivo **informa ao controlador de DMA** o que fazer e bloqueia a si mesmo.
 - c) O controlador de **DMA transfere os dados** para o **buffer** do kernel quando todos tiverem sido recuperados do dispositivo.
 - d) O controlador de **DMA interrompe a CPU** quando a transferência é concluída.
3. O **sistema operacional transfere os dados** para o **processo** do usuário e coloca o processo na fila de prontidão.
4. **Quando o processo** obtém a CPU, ele **inicia a execução seguindo a chamada do sistema**.

Resumo

- A E/S é cara por vários motivos:

- Dispositivos lentos e links de comunicação lentos
- Contenção de vários processos.
- Normalmente, a E/S é suportada por meio de chamadas de sistema e tratamento de interrupções, que são lentas.

- Abordagens para melhorar o desempenho:

- Reduzir a cópia de dados por meio do armazenamento em **cache** na memória
- Reduzir a **frequência** de **interrupções** usando grandes transferências de dados
- **Descarregue** a computação da **CPU** principal usando controladores de **DMA**.
- **Aumente** o **número** de **dispositivos** para reduzir a contenção de um único dispositivo e, assim, melhorar a utilização da CPU.
- **Aumentar** a **memória física** para reduzir o tempo de **paginação** e, assim, melhorar a utilização da CPU.

PERGUNTAS?

REFERÊNCIAS

- **TANENBAUM, Andrew.** Sistemas operacionais modernos.
- **SILBERSCHATZ, Abraham et al.** Fundamentos de sistemas operacionais: princípios básicos.
- **MACHADO, Francis; MAIA, Luiz Paulo.** Arquitetura de Sistemas Operacionais.
- **CARISSIMI, Alexandre et al.** Sistemas operacionais.