

## Sistemas de Entrada e Saída

### Referência:

**Silberschatz, Abraham. Sistemas Operacionais com Java. 7 ed. Rio de Janeiro: Elsevier, 2008.**

Hardware de E/S

Interface de E/S para as aplicações

Sub-sistema de E/S do kernel

Transformação de requisições de E/S em operações do hardware

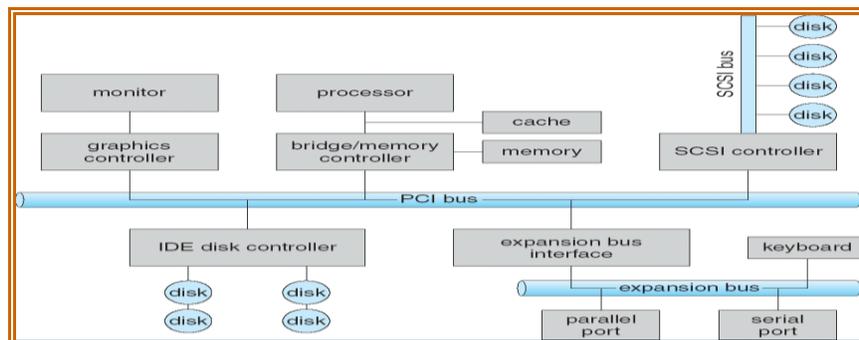
Desempenho

### Hardware de E/S

Existe um variedade muito grande de dispositivos de E/S

- Conceitos comuns:
  - Porta
  - Barramento (*daisy chain* ou acesso direto compartilhado)
  - Controlador (*host adapter*)
- As instruções de E/S controlam os dispositivos
- Os dispositivos têm endereços usados para:
  - Instruções de E/S diretas
  - E/S mapeadas na memória

### Estrutura típica de barramento de PC



## Locais de porta de E/S de dispositivo nos PCs (parcial)

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)

Maneiras de interagir com E/S dependendo do suporte do *hardware*.

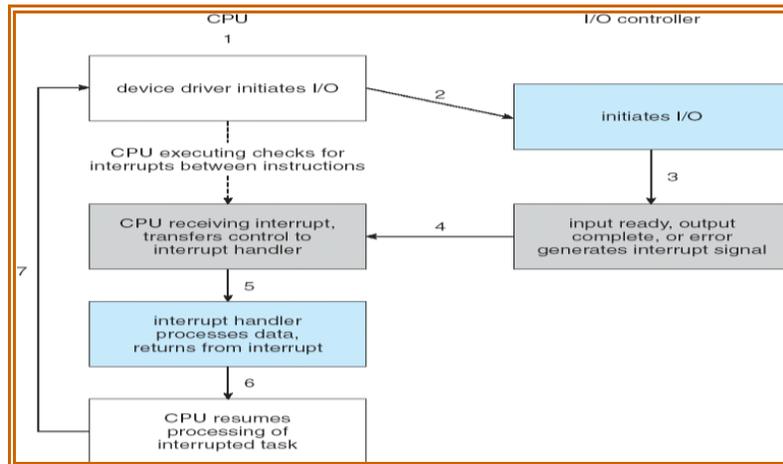
### Polling

- Determina-se o estado de cada dispositivo:
  - pronto para comando
  - ocupado
  - erro
- Ciclo de *espera ocupada* para aguardar uma E/S com o dispositivo.

### Interrupção

- Linha de requisição de interrupção do processador é acionada pelo dispositivo de E/S.
- O *hardware* de manipulação de interrupção recebe as interrupções.
- Mascara novas interrupções - ignora-as ou atrasa o atendimento delas.
  - Baseado na prioridade
  - Algumas interrupções podem não ser mascaráveis
- Usa o vetor de interrupções para selecionar a rotina de manipulação adequada.
- O mecanismo de interrupções também é usado para o tratamento de exceções (interrupções causadas pelo *software*)

## Ciclo de E/S controlado por interrupção



## Tabela de vetor de evento do Intel Pentium

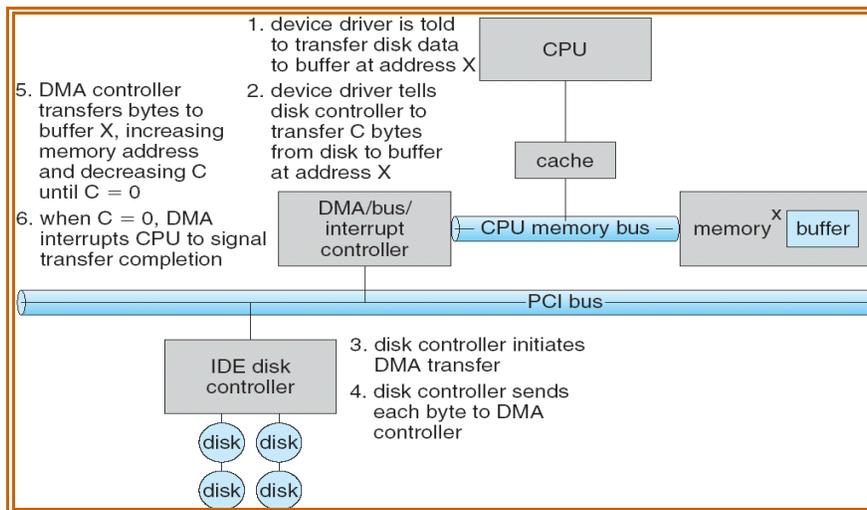
vector number	description
0	divide error
1	debug exception
2	null interrupt
3	breakpoint
4	INTO-detected overflow
5	bound range exception
6	invalid opcode
7	device not available
8	double fault
9	coprocessor segment overrun (reserved)
10	invalid task state segment
11	segment not present
12	stack fault
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

- Usado para evitar *programmed I/O* (E/S programada) para grandes quantidades de dados. *Programmed I/O (PIO)* é o que se faz com dispositivos de E/S que usam apenas interrupção ou *polling*, toda transferência de dados, todos os bytes, precisam passar pelo processador.
- Requer um controlador de DMA.
- Curto circuito o processador transferindo os dados diretamente entre a memória e o dispositivo de E/S.

## Processo de seis etapas para realizar transferência por DMA

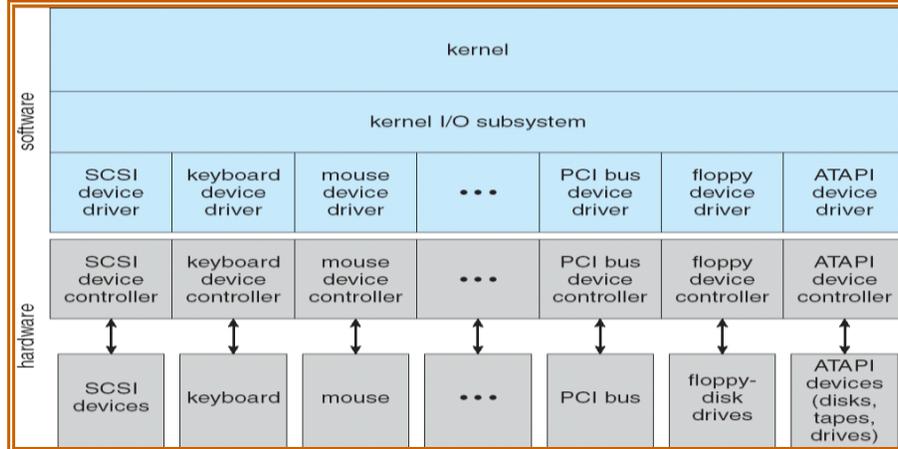
1. O *device driver* recebe um pedido de transferência de dados do disco para o buffer que inicia no endereço X.
2. O *device driver* comanda o controlador de discos para transferir C bytes do disco para o buffer de endereço X.
3. O controlador de disco inicia a transferência de DMA.
4. O controlador de disco manda cada *byte* para o controlador de DMA.
5. O controlador de DMA transfere os bytes para o buffer X, incrementando os endereços de memória e decrementando o valor de C para cada *byte* até C = 0.
6. Quando C = 0, o controlador de DMA interrompe o processador para assinalar o término da transferência.



## Interface de E/S para as aplicações

- Sistema de E/S chama comportamentos de dispositivo em classes genéricas
- Camada de driver de dispositivo esconde diferenças entre controladores de E/S do kernel
- Dispositivos variam em muitas dimensões
  - **Fluxo de caracteres ou bloco**
  - **Acesso seqüencial ou aleatório**
  - **Compartilhável ou dedicado**
  - **Velocidade de operação**
  - **Leitura/escrita, somente de leitura ou somente de escrita**

## Uma estrutura de E/S do kernel



## Características de 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 bloco e caractere

- Dispositivos de bloco incluem unidades de disco
  - Comandos incluem read, write, seek
  - E/S bruta ou acesso ao sistema de arquivos
  - Possível acesso a arquivo mapeado na memória
- Dispositivos de caractere incluem teclado, mouse, porta serial
  - Comandos incluem get, put
  - Bibliotecas dispostas no topo permitem edição de linha

## Dispositivos de bloco e caractere

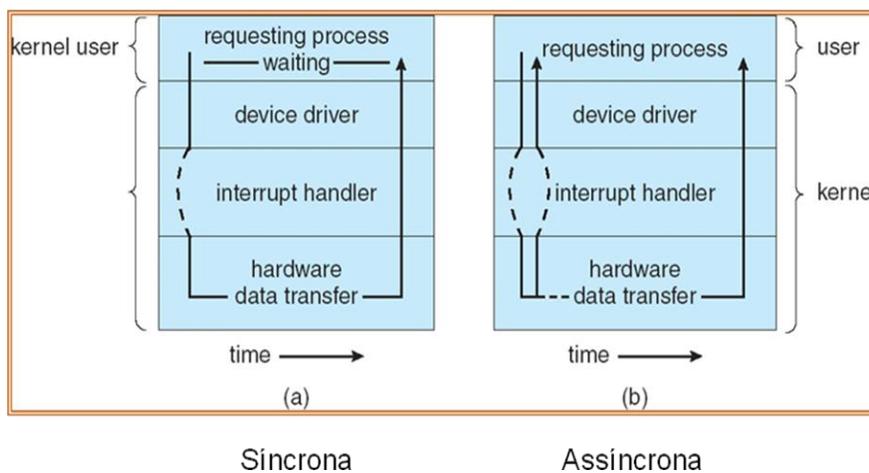
- Dispositivos de bloco incluem unidades de disco
  - Comandos incluem read, write, seek
  - E/S bruta ou acesso ao sistema de arquivos

- Possível acesso a arquivo mapeado na memória
- Dispositivos de caractere incluem teclado, mouse, porta serial
  - Comandos incluem get, put
  - Bibliotecas dispostas no topo permitem edição de linha

### E/S bloqueante e não bloqueante

- **Bloqueante** – processo suspenso até E/S terminar
  - Fácil de usar e entender
  - Insuficiente para algumas necessidades
- **Não bloqueante** – Chamada de E/S retorna o máximo possível
  - Interface de usuário, cópia de dados (E/S em buffer)
  - Implementado via multithreading
  - Retorna rapidamente com contagem de objetos lidos ou gravados
- **Assíncrona** - processo executa enquanto E/S executa
  - Difícil de usar
  - Subsistema de E/S sinaliza processo quando E/S termina

### Dois métodos de E/S



### Subsistema de E/S do kernel

- Escalonamento
  - Alguma ordenação de requisição de E/S via fila por dispositivo
  - Alguns SOs tentam imparcialidade
- Buffering – armazena dados na memória enquanto transfere entre dispositivos
  - Para enfrentar divergência de velocidade de dispositivo
  - Para enfrentar divergência de tamanho de transferência do dispositivo
- Para manter “semântica de cópia

- **Caching** – memória rápida mantendo cópia dos dados
  - Sempre apenas uma cópia
  - Fundamental para o desempenho
- **Spooling** – mantém saída para um dispositivo
  - Se dispositivo pode atender apenas uma requisição por vez
  - Por exemplo, impressão
- **Reserva de dispositivo** – oferece acesso exclusivo a um dispositivo
  - Chamadas do sistema para alocação e desalocação
  - Observa deadlock

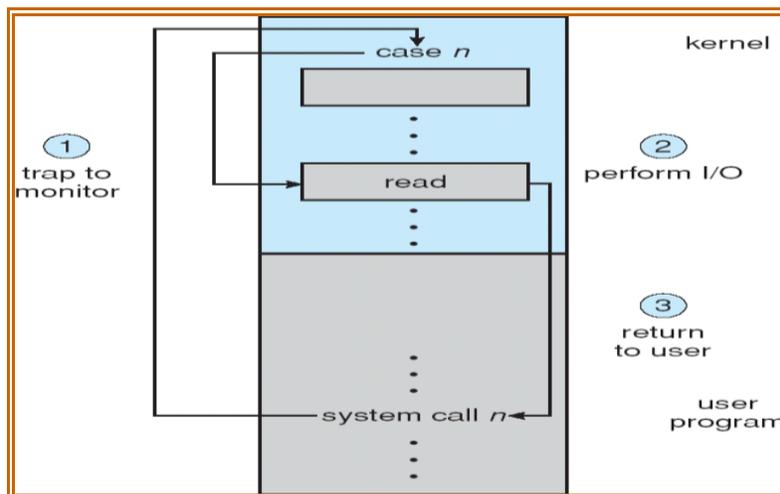
### Tratamento de erros

- O SO pode ser recuperar da leitura de disco, dispositivo não disponível, falhas de gravação transientes
- A maioria retorna um número ou código de erro quando a requisição de E/S falha
- Logs de erro do sistema mantêm relatórios de problemas

### Proteção da E/S

- O processo do usuário pode acidental ou propositalmente tentar atrapalhar a operação do sistema via instruções de E/S ilegais
  - Todas as instruções de E/S definidas como privilegiadas
  - E/S precisa ser realizada via chamadas do sistema
    - Locais de memória mapeada e de porta de E/S também precisam ser protegidos

### Uso de uma chamada do sistema para realizar E/S



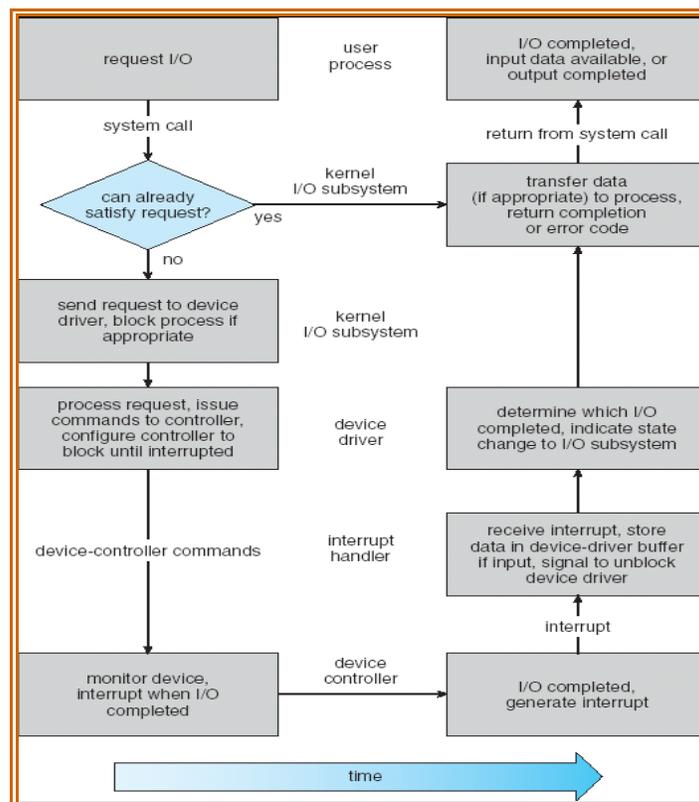
## Estruturas de dados do kernel

- O kernel mantém informações de estado para componentes de E/S, incluindo tabelas de arquivos abertos, conexões de rede, estado de dispositivo de caractere
- Estruturas de dados muito, muito complexas para rastrear buffers, alocação de memória, blocos “sujos”
- Algumas usam métodos orientados a objeto e passagem de mensagens para implementar E/S

## Requisições de E/S para operações de hardware

- Considere a leitura de um arquivo do disco para um processo:
  - Determinar dispositivo mantendo arquivo
  - Traduzir nome para representação de dispositivo
  - Ler dados fisicamente do disco para o buffer
  - Tornar dados disponíveis ao processo solicitante
  - Retornar controle ao processo

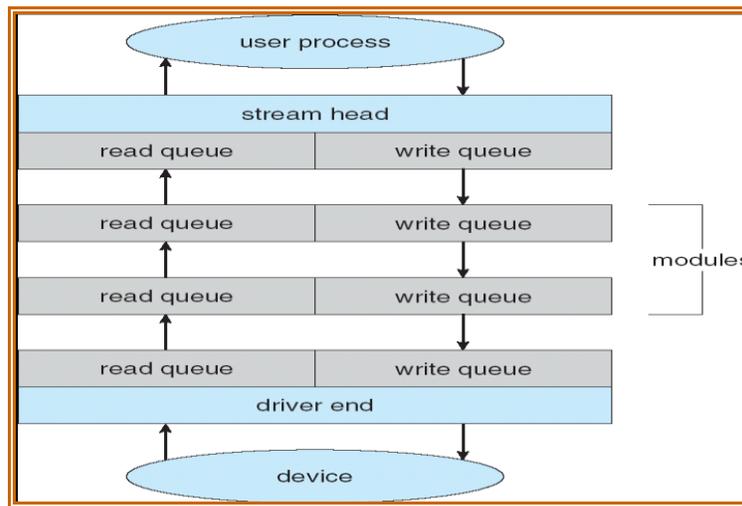
## Ciclo de vida de uma requisição de E/S



## STREAMS

- **STREAM** – um canal de comunicação full-duplex entre um processo em nível do usuário e um dispositivo no Unix System V em diante
- Um STREAM consiste em:
  - Interfaces de cabeça STREAM com o dispositivo
  - zero ou mais módulos STREAM entre elas.
- Cada módulo contém uma **fila de leitura** e uma **fila de gravação**
- Passagem de mensagens é usada para comunicar entre as filas

### A estrutura STREAMS



### Melhorando o desempenho

- Reduzir o número de trocas de contexto
- Reduzir a cópia de dados
- Reduzir interrupções usando grandes transferências, controladores inteligentes, polling
- Usar DMA
- Balancear CPU, memória, barramento e desempenho de E/S para um throughput mais alto

Software e hardware de E/S.

- As chamadas de sistema de E/S encapsulam o comportamento dos dispositivos em classes genéricas.
- A camada de *device drivers* esconde diferenças entre os controladores de E/S do restante do *kernel*.
- Os dispositivos podem variar de diversas maneiras:
  - acesso por fluxo de caracteres ou blocos
  - acesso sequencial ou aleatório
  - compartilhado ou dedicado
  - velocidade para cada operação
  - leitura e escrita, leitura apenas ou escrita apenas.

Dispositivos de E/S mais importantes:

### **Dispositivos a blocos e a caracteres**

- Os dispositivos a blocos incluem os acionadores de discos:
  - os comandos incluem: *read*, *write*, *seek*
  - Acesso de E/S puro (*raw I/O*) ou pelo sistema de arquivos
  - Acesso a arquivo mapeado em memória é possível.

### **Dispositivos de redes**

- Mudam o suficiente entre blocos e caracteres para terem uma interface própria.
- O UNIX e o WindowsNT incluem a interface de *socket*
  - Separa o protocolo de rede da operação de rede
  - incluem a funcionalidade do **select**
- Maneiras de programar e usar variam muito: pipes, FIFOs, streams, queues, mailboxes

### **Relógios (*Clocks*) e temporizadores (*Timers*)**

- Fornece a hora atual, o tempo decorrido, alarme (temporizador)
- Pode ser programável para gerar interrupções (alarmes) periódicos
- A chamada sistema **ioctl** (no UNIX) cobre aspectos incomuns de E/S como relógios e temporizadores.

### **E/S bloqueadora ou não bloqueadora**

- **Bloqueadora** - o processo é suspenso até o término da E/S

- Fácil de usar e de entender
- ineficiente para algumas necessidades
- **Não bloqueadora** - a chamada de sistema de E/S retorna tão rápido quanto for possível
  - Interface de usuário, cópia de dados (E/S bufferizada)
  - Implementada com *multi-threading*
  - Retorna rapidamente com o número de *bytes* lidos ou escritos
- **Assíncrona** - o processo continua rodando enquanto a operação de E/S é executada
  - Difícil de usar
  - O sistema de E/S vai sinaliza ao processo quando a E/S terminar.

### **XII.3 Sub-sistema de E/S do kernel**

- **Cache** (ou bufferização) - a memória rápida guarda uma cópia dos dados
  - Apenas uma cópia
  - Chave para melhorar o desempenho
- **Spooling** (enfileiramento) - segura as saídas para um dispositivo
  - Se o dispositivo só puder atender uma requisição de cada vez
  - Ex., impressão
- **Reserva de dispositivos** - fornece acesso exclusivo a um dispositivo
  - Chamads de sistema para a alocação e desalocação
  - Cuidado com **deadlock**

### **Manipulação de erros**

- O SO pode se recuperar de falhas de leitura de disco, indisponibilidade de dispositivos, transitórios de escrita
- A maioria retorna um código (número inteiro) de erro quando a requisição de E/S falha
- O Sistema de registros (*logs*) de erros guarda uma lista de problemas ocorridos.

### **Estruturas de dados do *kernel***

- O *kernel* guarda as informações de estado dos componentes de E/S, inclusive as tabelas de arquivos abertos, de conexões de redes, o estado de dispositivos a caractéres.
- Mantem muitas estruturas de dados complexas para rastrear *buffers*, alocação de memória, blocos *sujos*.
- Alguns usam métodos de orientação objeto e passagem de mensagens para implementar E/S.

Estruturas do kernel do UNIX para E/S.

## XII.4 Transformação de requisições de E/S em operações do hardware

- Tome por exemplo uma leitura de disco de um processo:
  - Determinação do dispositivo que está com o arquivo
  - Tradução do nome para representação do dispositivo (posição no disco)
  - Leitura física dos dados do disco para o *buffer*
  - Disponibilização dos dados para o processo requerente
  - Retorno do controle para o processo

Ciclo de vida de uma requisição de leitura bloqueadora.

1. Um processo emite uma chamada de sistema `read` bloqueadora para um descritor de arquivo de um arquivo que foi aberto anteriormente com um `open`.
2. O código da chamada de sistema no kernel verifica a correção dos parâmetros. No caso de uma entrada, se os dados já estiverem disponíveis na *cache*, eles serão retornados para o processo e a requisição de E/S termina.
3. Se os dados não estiverem na *cache*, uma operação de E/S física precisa ser realizada, portanto o processo é removido da fila de execução e colocado na fila de espera pelo disco e o pedido de E/S é escalonado (agendado). O subsistema de E/S envia a requisição para o *device driver*. Dependendo do SO, o pedido é enviado através de uma chamada de rotina ou através de uma mensagem interna do kernel.
4. O *device driver* aloca espaço de *buffer* do kernel para receber os dados e escalona a operação de E/S. O *device driver* envia comandos para a controladora de dispositivos escrevendo nos registradores de controle do controlador do dispositivo.
5. O controlador de dispositivos opera o *hardware* do dispositivo para realizar a transferência de dados.
6. O *device driver* pode efetuar *polling* para consultar o *status* e buscar os dados, ou pode configurar uma transferência DMA para a memória do *kernel*. Assume-se que a transferência é gerenciada pelo controlador de DMA que gera uma interrupção quando a transferência terminar.
7. A rotina de tratamento de interrupção apropriada recebe a interrupção através do vetor de interrupções, armazena quaisquer dados necessários, sinaliza o *device driver* e retorna da interrupção.
8. O kernel transfere os dados ou códigos de retorno para o espaço de endereçamento do processo solicitante e move o processo da fila de espera para a fila de processos prontos.
9. O processo foi desbloqueado quando foi movido para a fila de prontos. Quando o escalonador atribuir o processo para o processador, ele retoma a execução na conclusão da chamada de sistema.

## XII.5 Desempenho

- O sistema de E/S é um dos principais fatores para o desempenho de um computador
  - precisa do processador para a execução do código do *device driver* e do código de E/S do *kernel*
  - são necessárias comutações de contexto devido às interrupções
  - cópia de dados
  - o tráfego de dados de rede é particularmente *stressante*

Comunicação entre computadores.

### Melhoramentos de desempenho

- Redução do número de comutações de contexto
- Redução da quantidade de cópia de dados
- Redução das interrupções através de transferências maiores, controladores mais inteligentes e *polling* (se a espera ocupada for mínima)
- Uso de DMA
- BALanceamento do desempenho do processador, da memória, dos barramentos e das E/S para obter o maior *throughput* possível (maior taxa de processos/tarefas executadas possível).

---

[Paulino Ng<Paulino\\_Ng@pucsp.br>](mailto:Paulino_Ng@pucsp.br)

Last modified: Thu Nov 07 17:09:19 E. South America Daylight Time 2002