#### Sumário

- Introdução
- 2 Princípios de hardware de E/S
- 3 Princípios do software de E/S
- 4 Camadas do software de E/S
- Discos
- 6 E/S no Linux

lon	&	Raf	fael	R	. OI	oell	neir	0						2/72
	4		Þ	4	ð	•	4	Ē	•	4	ŧ	•	₽	200

Introdução
Princípios de hardware de E/S
Princípios do software de E/S
Camadas do software de E/S
Discos
E/S no Linux

Introdução

Princípios de hardware de E/S

Princípios do software de E/S

Camadas do software de E/S

Discos

E/S no Linux

Sistemas Operacionais

Dispositivos de E/S Controladores de dispositivos E/S mapeada em memória DMA Interrupções revisitadas

Maurício A. Pill

#### Introdução

 $\bullet$  O controle dos dispositivos de E/S é uma das principais funções do SO

Sistemas Operacionais:

Gerência de Entrada/Saída

Prof. Maurício Aronne Pillon

Prof. Rafael Obelheiro

UDESC/CCT - Departamento de Ciência da Computação {mauricio.pillon,rafael.obelheiro}@udesc.br

Joinville, novembro de 2014

- O SO deve oferecer ao usuário uma interface simples e fácil de usar, ocultando na medida do possível as particularidades de dispositivos específicos
- O gerenciamento de E/S responde por uma fração significativa do código em um SO
  - variedade de dispositivos que necessitam de suporte

#### Sumário

- Introdução
- 2 Princípios de hardware de E/S
- 3 Princípios do software de E/S
- Camadas do software de E/S
- Discos
- 6 E/S no Linux

Dispositivos de E/S
Controladores de dispositivos
E/S mapeada em memória
DMA
Interrupções revisitadas

Introdução
Princípios de hardware de E/S
Princípios do software de E/S
Camadas do software de E/S
Discos
E/S no Linux

Velocidade dos dispositivos

Dispositivos de E/S Controladores de dispositivos E/S mapeada em memória DMA Interrupções revisitadas

#### Dispositivos de E/S

- Lembrete: o ponto de vista é o da programação do dispositivo, não da sua construção física
- Existem duas grandes classes de dispositivos
- **Orientados a blocos**: transferem dados em unidades de tamanho fixo (blocos)
  - blocos variam de 512 bytes a 32 KB (tipicamente)
  - endereçamento também é por bloco (ler o bloco n)
- Orientados a caracteres: transferem dados em sequências (fluxos) de caracteres
  - não podem ser endereçados
- Nem todos os dispositivos se encaixam nessas classes
  - um relógio só gera interrupções a intervalos periódicos

Discos

E/S no Linux

(4日) (部) (主) (主) (主) (つ)	◆□▶ ◆□▶ ◆□▶ ◆□▶ □□ →	200
--------------------------	----------------------	-----

As velocidades dos dispositivos variam bastante

Dispositivo	Taxa de dados
Teclado	10 bytes/s
Mouse	100 bytes/s
Modem 56 K	7 KB/s
Canal telefônico	8 KB/s
Linhas ISDN dual	16 KB/s
Impressora a laser	100 KB/s
Scanner	400 KB/s
Ethernet clássica	1,25 MB/s
USB (universal serial bus — barramento serial universal)	1,5 MB/s
Câmara de vídeo digital	4 MB/s
Disco IDE	5 MB/s
CD-ROM 40x	6 MB/s
Ethernet rápida	12,5 MB/s
Barramento ISA	16,7 MB/s
Disco EIDE (ATA-2)	16,7 MB/s
FireWire (IEEE 1394)	50 MB/s
Monitor XGA	60 MB/s
Rede SONET OC-12	78 MB/s
Disco SCSI Ultra 2	80 MB/s
Ethernet Gigabit	125 MB/s
Dispositivo de Fita Ultrium	320 MB/s
Barramento PCI	528 MB/s
Barramento da Sun Gigaplane XB	20 GB/s

6/72

Sistemas Operacionais
Introdução
Princípios de hardware de E/S
Princípios do software de E/S
Camadas do software de E/S

Dispositivos de E/S

Controladores de dispositivos E/S mapeada em memória

Maurício A. Pillon & Rafael R. Obelheiro

DMA Interrupções revisitadas Sistemas Operacionais
Introdução
Princípios de hardware de E/S
Princípios do software de E/S
Camadas do software de E/S
Discos

Dispositivos de E/S Controladores de dispositivos E/S mapeada em memória DMA Interrupções revisitadas

Maurício A. Pillon & Rafael R. Obelheiro

#### Controladores de dispositivos

- Dispositivos tipicamente têm dois grandes componentes
  - eletrônico: programação
  - mecânico: dispositivo propriamente dito
- O componente eletrônico é o controlador
  - o mesmo controlador pode tratar vários dispositivos idênticos
- Tarefas do controlador
  - leitura e escrita serial de bits
  - conversão entre fluxos de bits e blocos fixos
  - verificação e correção de erros
  - tornar um bloco ou caracter disponível para ser copiado para a memória

### E/S mapeada em memória (1)

- Os controladores de dispositivos precisam ser endereçados de alguma maneira
  - registradores de dados, status e controle
- Existem duas formas básicas de endereçamento de E/S
- Portas de E/S: endereços separados para E/S

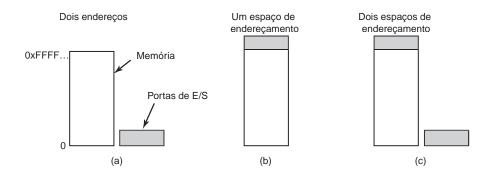
Sistemas Operacionais

- exigem instruções específicas de máquina (IN, OUT)
- E/S mapeada em memória: endereços de E/S são os mesmos enderecos de memória
  - ocupam faixas distintas no espaço de endereçamento
  - utilizam as mesmas instruções de máquina que referenciam a memória

Dispositivos de E/S Controladores de dispositivos E/S mapeada em memória DMA Interrupções revisitadas Introdução
Princípios de hardware de E/S
Princípios do software de E/S
Camadas do software de E/S
Discos

Dispositivos de E/S Controladores de dispositivos E/S mapeada em memória DMA Interrupções revisitadas

### E/S mapeada em memória (2)



(a) Espaços de memória e E/S separados

Princ

- (b) E/S mapeada em memória
- (c) Endereçamento híbrido (memória para dados, portas para controle/status)

### E/S mapeada em memória (3)

Vantagens de E/S mapeada em memória:

- Registradores do controlador são posições de memória e podem ser referenciados como variáveis em C
  - dispensa o uso de Assembly para escrever drivers
- Dispensa o uso de mecanismos especiais de proteção para impedir o acesso de processos de usuário aos dispositivos
  - mecanismos de proteção de memória são suficientes

emas Operacionais

hardware de E/S

lo software de E/S

o software de E/S

Discos

- Qualquer instrução que referencie memória pode ser usada para manipular os registradores do controlador
  - torna a programação mais flexível e (possivelmente) eficiente

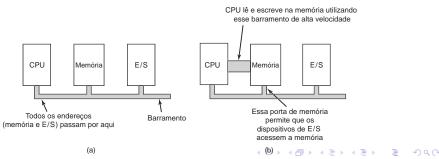
Sistemas Operacionais	Maurício A. Pillon & Rafael R. Obelheiro	9/72	Siste
Introdução cípios de hardware de E/S icípios do software de E/S madas do software de E/S Discos	Dispositivos de E/S Controladores de dispositivos E/S mapeada em memória DMA		<b>Princípios de</b> Princípios d Camadas d

◆ロト 4周ト 4 章 ト 4 章 ト 章 めなべ

## E/S mapeada em memória (4) Acesso direto à memória (DMA)

Desvantagens de E/S mapeada em memória:

- Complica o gerenciamento do cache de memória
  - o cache deve ser desabilitado para endereços referentes aos registradores de dispositivos
- Adiciona complexidade quando existem múltiplos barramentos de memória
  - todos os endereços têm de ser inspecionados pela memória e pelos controladores de dispositivo



- Controladores de dispositivos trabalham com unidades de blocos ou bytes
  - o controlador precisa ser continuamente atendido pelo SO para realizar uma transferência de dados
  - o desperdício de CPU é muito grande
- Uma solução alternativa é o uso de acesso direto à memória (DMA, direct memory access)
  - as transferências de dados são feitas diretamente entre controlador e memória, sem passar pela CPU
  - requer um controlador específico (controlador de DMA)
  - o controlador de DMA tem acesso direto ao barramento

イロト (個) (国) (国) (国)

Maurício A. Pillon & Rafael R. Obelheiro

Dispositivos de E/S

DMA

Controladores de dispositivos

E/S mapeada em memória

Interrupções revisitadas

Dispositivos de E/S Controladores de dispositivos E/S mapeada em memória Interrupções revisitadas

Princípios de hardware de E/S Princípios do software de E/S Camadas do software de E/S Discos

Dispositivos de E/S Controladores de dispositivos E/S mapeada em memória DΜΔ Interrupções revisitadas

#### Leitura de dados sem DMA

- 1. O controlador efetua a leitura do bloco de dados (serialmente, bit a bit)
- 2. O controlador faz a verificação e correção de erros
- 3. O controlador causa uma interrupção
- 4. O SO lê o bloco do buffer do controlador, um byte ou bloco por vez
- 5. O SO armazena os dados lidos na memória
- 6. Repete até que a leitura esteja concluída

#### Leitura de dados com DMA

- 1. A CPU programa o controlador de DMA, indicando
  - quais os blocos que devem ser lidos
  - o endereco inicial na memória
  - o total de bytes/blocos a transferir
- 2. O controlador de DMA solicita a leitura do próximo bloco ao controlador de disco
- 3. O controlador de disco interrompe o controlador de DMA quando tiver lido um bloco
- 4. O controlador de DMA comanda a transferência dos dados do buffer do controlador de disco para a memória
- 5. Se houver mais dados a ler, volta ao passo 2; senão, interrompe a CPU sinalizando o final da transferência



Princípios de hardware de E/S Princípios do software de E/S Camadas do software de E/S Discos

Dispositivos de F/S Controladores de dispositivos E/S mapeada em memória Interrupções revisitadas

Sistemas Operacionais Princípios de hardware de E/S Princípios do software de E/S Camadas do software de E/S Discos

Dispositivos de F/S Controladores de dispositivos E/S mapeada em memória DΜΔ Interrupções revisitadas

Maurício A. Pillon & Rafael R. Obelheiro

4 日 7 4 周 7 4 3 7 4 3 7 9 9

#### Transferência de dados com DMA

#### → Dispositivo 1. CPU programa Controlador Controlador Memória o controlador CPU de DMA de DMA de disco principal Endereço Contador Controle 4. Confirmação 2. DMA solicita transfe-Interrompe quando concluído rência para a memória 3. Dados transferidos Barramento

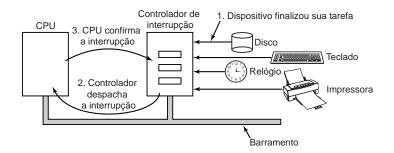
## Interrupções revisitadas (1)

#### Funcionamento de interrupções:

- Quando um dispositivo de E/S finaliza seu trabalho, ele gera uma interrupção
- Isso é feito através do envio de um sinal pela linha de barramento à qual o dispositivo está associado
- O sinal é detectado pelo chip controlador de interrupções localizado na placa-mãe
- O sinal de interrupção faz com que a CPU pare de executar a tarefa atual e vá tratar a interrupção

4□ → 4同 → 4 ■ → ■ 900

### Interrupções revisitadas (2)



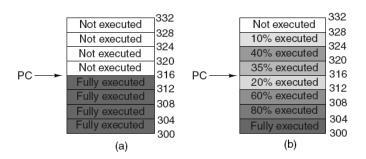
- A confirmação (3) sinaliza ao controlador que a interrupção foi atendida
  - o controlador de interrupção avisa ao dispositivo e pode liberar outra interrupção pendente

### Interrupções precisas e imprecisas (1)

- Em arquiteturas pipelined e superescalares, diversas instruções podem estar sendo executadas quando uma interrupção é gerada
  - qual o estado da CPU quando ocorre a interrupção?
- Interrupções precisas deixam o sistema em um estado bem definido
  - 1. O PC é salvo em um lugar conhecido
  - 2. Todas as instruções antes do PC foram completamente executadas
  - 3. Nenhum instrução depois do PC foi executada
  - 4. O estado de execução da instrução apontada pelo PC é conhecido
- Interrupções imprecisas não atendem a esses requisitos
- Interrupções precisas complicam o hardware e simplificam o SO
  - abordagem adotada pela maioria das arquiteturas atuais

	←□ → ←□ → ← □ → ← □	▶ \( \begin{align*}		←□ → ←□ → ← = → ← = →	
Sistemas Operacionais	Maurício A. Pillon & Rafael R. Obelheiro	17/72	Sistemas Operacionais	Maurício A. Pillon & Rafael R. Obelheiro	18/72
Introdução  Princípios de hardware de E/S  Princípios do software de E/S  Camadas do software de E/S  Discos  E/S no Linux	Dispositivos de E/S Controladores de dispositivos E/S mapeada em memória DMA Interrupções revisitadas		Introdução Princípios de hardware de E/S Princípios do software de E/S Camadas do software de E/S Discos E/S no Linux	Objetivos do software de E/S Técnicas para realizar E/S	

### Interrupções precisas e imprecisas (2)



- (a) interrupção precisa
- (b) interrupção imprecisa

### Sumário

- Introdução
- Princípios de hardware de E/S
- Princípios do software de E/S
- Camadas do software de E/S
- Discos
- 6 E/S no Linux

イロト イ伊ト イラト イラト ラ りなら

Objetivos do software de E/S Técnicas para realizar E/S

Princípios de hardware de E/S Princípios do software de E/S

Objetivos do software de E/S (2)

Objetivos do software de E/S Técnicas para realizar E/S

### Objetivos do software de E/S (1)

- Independência do dispositivo: programas de usuário devem funcionar com quaisquer dispositivos, sem se preocupar com as diferenças entre eles
  - o SO é responsável por tratar as diferencas
- Nomeação uniforme: todos os dispositivos devem ser nomeados da mesma maneira, sem distinções
  - exemplo: montagem de sistemas de arquivos no UNIX
- Tratamento de erros: muitos erros de E/S podem ser tratados em baixo nível, sem chegar até o usuário
  - erros transientes
  - ideal é tratar o mais próximo possível do hardware

- Transferência síncrona vs assíncrona: SO precisa implementar chamadas de sistema síncronas (bloqueantes) usando operações de E/S assíncronas
  - operações assíncronas: o chamador não espera pelo resultado, ele solicita a operação e mais tarde recupera o resultado
- **Buffers**: operações de E/S exigem que o SO gerencie buffers de armazenamento temporário de dados
  - transferências de dados entre buffers podem ter um impacto significativo no desempenho

	<ul><li>←□ → ←□ → ←□ → □</li><li>←□ → ←□ → ←□ → □</li></ul>	20		◆□▶ ◆圖▶ ◆團▶ ◆團▶	1	200
Sistemas Operacionais	Maurício A. Pillon & Rafael R. Obelheiro 21	1/72	Sistemas Operacionais	Maurício A. Pillon & Rafael R. Obelheiro		22/72
Introdução cípios de hardware de E/S cípios do software de E/S nadas do software de E/S Discos E/S no Linux	Objetivos do software de E/S Técnicas para realizar E/S		Introdução Princípios de hardware de E/S Princípios do software de E/S Camadas do software de E/S Discos E/S no Linux	Objetivos do software de E/S Técnicas para realizar E/S		

### Técnicas para realizar E/S

Princí

- E/S programada
- E/S orientada a interrupções
- E/S usando DMA

# E/S programada (1)

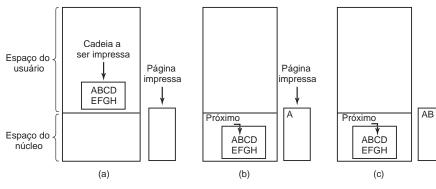
- A CPU faz todo o trabalho
  - forma mais simples
- Exemplo: impressão de uma string
  - SO copia string para um buffer

Sistemas Operacionais

- dados do buffer são enviados, caractere a caractere, para a impressora (usando E/S mapeada em memória, p.ex.)
- Espera ocupada ou polling
- CPU fica muito tempo ociosa
  - solução adequada para sistemas embarcados ou se E/S é muito rápida (e ociosidade é pequena)

if (count == 0) {

### E/S programada (2)



copy\_from\_user(buffer, p, cont); for  $(i=0; i < count; i++) {$ while (\*printer\_status\_reg !=READY); \*printer\_data\_register = p[i]; return\_to\_user();

/\* p é o buffer do núcleo \*/ /\* executa o laço para cada caractere \*/ /\* executa o laço até PRONTO \*/

/\* envia um caractere para a saída \*/

4 D > 4 A > 4 B > 4 B > B

### E/S orientada a interrupções (1)

- Em vez da CPU esperar pelo dispositivo, ela inicia a operação de E/S e espera por uma interrupção para continuar
  - enquanto isso, outros processos podem ser executados

```
copy from user(buffer, p, count);
enable interrupts();
while (*printer_status_reg != READY);
*printer_data_register = p[0];
scheduler();
```

unblock user(): } else { \*printer\_data\_register = p[i]; count = count - 1; i = i + 1; acknowledge interrupt();

return from interrupt(); (b)

(a) início da operação

(b) tratamento da interrupção

4 日 1 4 周 2 4 3 4 3 4 3 4 3 4

Sistemas Operacionais Princípios de hardware de E/S Princípios do software de E/S Camadas do software de E/S Discos

Objetivos do software de E/S Técnicas para realizar E/S

Maurício A. Pillon & Rafael R. Obelheiro

Maurício A. Pillon & Rafael R. Obelheiro

Sistemas Operacionais Princípios de hardware de E/S Princípios do software de E/S Camadas do software de E/S Discos E/S no Linux

Objetivos do software de E/S Técnicas para realizar E/S

Maurício A. Pillon & Rafael R. Obelheiro

### E/S orientada a interrupções (2)

- Mais eficiente que E/S programada
- CPU ainda é interrompida para cada caractere impresso

### E/S usando DMA

(a)

- CPU programa o controlador de DMA e é interrompida somente no final da operação
  - por exemplo, depois de todo o buffer impresso
- Em geral é mais eficiente
  - CPU é menos interrompida e fica livre para outros processos
  - controlador de DMA pode ser mais lento que a CPU

```
copy_from_user(buffer, p, count);
set_up_DMA_controller();
scheduler();
```

return\_from\_interrupt();

unblock user():

(a)

(b)

4 日 F 4 周 F 4 国 F 4 国 F 5

acknowledge\_interrupt();

25/72

Tratadores de interrupções Drivers de dispositivos E/S independente de dispositivo Software de E/S no espaço de usuário Introdução
Princípios de hardware de E/S
Princípios do software de E/S
Camadas do software de E/S
Discos
E/S no Linux

Tratadores de interrupções Drivers de dispositivos E/S independente de dispositivo Software de E/S no espaço de usuário

#### Sumário

- Introdução
- 2 Princípios de hardware de E/S
- 3 Princípios do software de E/S
- 4 Camadas do software de E/S
- Discos
- 6 E/S no Linux

#### Camadas do software de E/S

- Software de E/S é organizado em quatro camadas
  - cada camada tem uma função bem definida

Software de E/S no nível do usuário

Software do sistema operacional independente do dispositivo

Drivers do dispositivo

Tratadores de interrupções

Hardware

◆□▶ ◆□▶ ◆□▶ ◆□▶ □ ** ** ** ** ** ** ** ** ** ** ** ** *	C
---	---

 4 □ ▶ 4 ∃ ▶ 4 ∃ ▶ 3 €
 ★ ○ ○

 Maurício A. Pillon & Rafael R. Obelheiro
 30/72

Sistemas Operacionais
Introdução
Princípios de hardware de E/S
Princípios do software de E/S
Camadas do software de E/S
Camadas do software de E/S

Tratadores de interrupções

Drivers de dispositivos E/S independente de dispositivo Software de E/S no espaço de usuário

Maurício A. Pillon & Rafael R. Obelheiro

Sistemas Operacionais

Introdução
Princípios de hardware de E/S
Princípios do software de E/S
Camadas do software de E/S
Discos
E/S no Linux

Tratadores de interrupções Drivers de dispositivos E/S independente de dispositivo Software de E/S no espaço de usuário

#### Tratadores de interrupções

- As interrupções devem ser escondidas o máximo possível
  - uma forma de fazer isso é bloqueando o driver que iniciou uma operação de E/S até que uma interrupção notifique que a E/S foi completada
- Rotina de tratamento de interrupção cumpre sua tarefa
  - e então desbloqueia o driver que a chamou
- Tratamento de interrupções em software envolve
  - trocar de contexto (processo corrente → tratador)
  - tratar a interrupção
  - escolher um novo processo para executar (escalonador)
  - ullet trocar de contexto (tratador o novo processo)

### Drivers de dispositivos (1)

- Cada controlador possui registradores para comandos e status
- O número de registradores varia de dispositivo para dispositivo
- Cada dispositivo de E/S precisa de um código específico de tratamento (driver do dispositivo)
- Em geral, os drivers são escritos pelo fabricante

Sistemas Operacionais

- questões de confiabilidade/segurança
- É desejável que um driver trate de uma classe de dispositivos

Tratadores de interrupções Drivers de dispositivos E/S independente de dispositivo Software de E/S no espaço de usuário Introdução
Princípios de hardware de E/S
Princípios do software de E/S
Camadas do software de E/S
Discos
E/S no Linux

Drivers de dispositivos (3)

Tratadores de interrupções

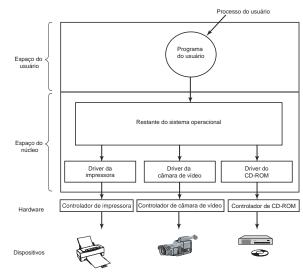
Drivers de dispositivos

E/S independente de dispositivo

Software de E/S no espaço de usuário

#### Drivers de dispositivos (2)

- Duas grandes categorias:
  - dispositivos de bloco: disco, fita, ...
  - dispositivos de caractere: mouse, teclado, impressora, ...
- Em geral, os sistemas operacionais definem uma interface padrão para cada categoria
- Algumas funções de um driver:
  - tratar requisições abstratas de leitura ou gravação independente do dispositivo
  - inicializar o dispositivo
  - tratar necessidade de energia
  - tratar eventos



comunicação entre drivers e controladores é feita pelo barramento

**₽** 99€

34/72

Sistemas Operacionais

Introdução
Princípios de hardware de E/S
Princípios do software de E/S
Camadas do software de E/S
Discos

Maurício A. Pillon & Rafael R. Obelheiro

Tratadores de interrupções

4 D > 4 A > 4 B > 4 B > B

Drivers de dispositivos E/S independente de dispositivo

Software de E/S no espaço de usuário

Princípios de hardware de E/S
Princípios do software de E/S
Camadas do software de E/S
Discos
E/S no Linux

Tratadores de interrupções Drivers de dispositivos **E/S independente de dispositivo** Software de E/S no espaço de usuário

Maurício A. Pillon & Rafael R. Obelheiro

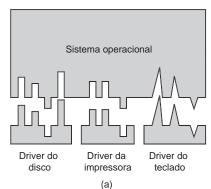
#### E/S independente de dispositivo

#### Funções do software de E/S independente de dispositivo:

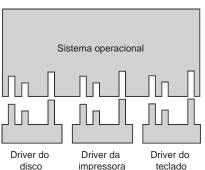
- 1. Interface uniforme para os drivers de dispositivos
- 2. Uso de buffers de E/S
- 3. Relatório de erros
- 4. Alocação e liberação de dispositivos dedicados
- 5. Tamanho de bloco independente de dispositivo

### Interface uniforme para os drivers de dispositivos

Sistemas Operacionais







- (a) Sem uma interface padrão do driver
- (b) Com uma interface padrão do driver

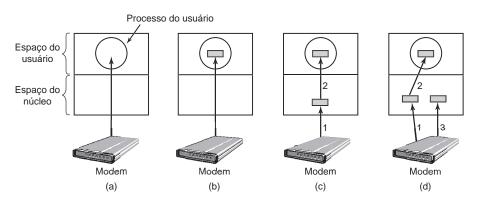
4□ → 4同 → 4 ■ → ■ 900

Tratadores de interrupções Drivers de dispositivos E/S independente de dispositivo Software de E/S no espaço de usuário

Introdução Princípios de hardware de E/S Princípios do software de E/S Camadas do software de E/S Discos E/S no Linux

Tratadores de interrupções Drivers de dispositivos E/S independente de dispositivo Software de E/S no espaço de usuário

#### Uso de buffers de E/S

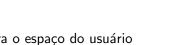


- (a) Entrada sem uso de buffer
- (b) Uso de buffer no espaço do usuário
- (c) Uso de buffer no núcleo seguido de cópia para o espaço do usuário

Sistemas Operacionais Princípios de hardware de E/S Princípios do software de E/S Camadas do software de E/S

E/S no Linux

(d) Uso de buffer duplo no núcleo



◆□▶ ◆□▶ ◆■▶ ◆■ り९◎

#### Relatórios de erros

- Erros de programação
  - escrita em um dispositivo de entrada (teclado)
  - fornecimento de endereços inválidos
  - atitude a ser tomada: retornar o código (tipo) de erro ocorrido ao processo envolvido
- Erros reais de E/S
  - tentativa de escrita em um bloco danificado
  - podem ser irrecuperáveis
    - leitura do MBR do disco de boot, por exemplo
  - atitude deve ser decidida pelo driver

Maurício A. Pillon & Rafael R. Obelheiro	37/72	Sistemas Operacionais	Maurício A. Pillon & Rafael R. Obelheiro 38/7	2
		Introdução		
Tratadores de interrupções		Princípios de hardware de E/S	Tratadores de interrupções	
Drivers de dispositivos		Princípios do software de E/S	Drivers de dispositivos	
E/S independente de dispositivo		Camadas do software de E/S	E/S independente de dispositivo	
Software de E/S no espaço de usuário		Discos	Software de E/S no espaço de usuário	
		= 10		

#### Alocação/liberação de dispositivos

- Alguns dispositivos, tais como gravadores de CD/DVD, podem ser usados por apenas um único processo por vez
- Associar chamadas de sistema diretamente ao dispositivo (open)
  - caso o dispositivo não possa ser alocado, a chamada pode falhar ou o chamador ser bloqueado
    - decisão de projeto

## Software de E/S no espaço de usuário

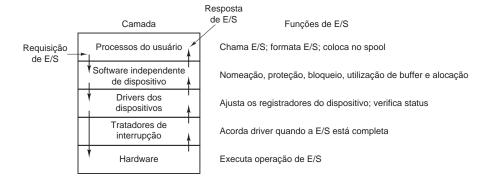
- Chamadas de sistema e funções de biblioteca que permitem ao usuário fazer E/S
  - read, write, printf, scanf, ...
- Spooling

◆□▶ ◆□▶ ◆□▶ ◆□▶ □ のQで

Tratadores de interrupções Drivers de dispositivos E/S independente de dispositivo Software de E/S no espaço de usuário Introdução
Princípios de hardware de E/S
Princípios do software de E/S
Camadas do software de E/S
Discos
E/S no Linux

Organização dos discos Desempenho do disco Escalonamento de disco

#### Camadas do sistema de E/S: resumo



Sumário

- 1 Introdução
- 2 Princípios de hardware de E/S
- 3 Princípios do software de E/S
- 4 Camadas do software de E/S

Organização dos discos (1)

- Discos
- 6 E/S no Linux

Sistemas Operacionais Maurício A. Pillon & Rafael R. Obelheiro 41/72 Sistemas Operacionais Maurício A. Pillon & Rafael R. Obelheiro	
	42/72
Introdução Princípios de hardware de E/S Princípios do software de E/S Princípios do software de E/S Camadas do software de E/S Oiscos E/S no Linux  Introdução Princípios de hardware de E/S Princípios do software de E/S Princípios do software de E/S Princípios do software de E/S Camadas do software de E/S Escalonamento de disco  Organização dos discos Desempenho do disco Discos E/S no Linux  Introdução Princípios de hardware de E/S Princípios do software de E/S Princípios do softwa	

### Discos magnéticos

- O disco é um dos dispositivos de E/S mais importantes
- Serve não apenas como memória secundária (de massa) como também oferece suporte a outras funções do SO
  - swapping
  - memória virtual
- Consideraremos aqui discos magnéticos (hard disks)
  - princípios de funcionamento similares aos de outros dispositivos como CD e DVD

• Um disco é formado por um conjunto de pratos

- superfícies cobertas por material ferromagnético
- A superfície do disco é magnetizada para armazenar os bits desejados
  - cabeça (cabeçote) de leitura e escrita converte bits de/para padrões magnéticos registrados na superfície do disco
  - há uma ou duas cabeças por superfície
- Cada disco é organizado em trilhas concêntricas
- Cada trilha é dividida em **setores** de tamanho fixo

Sistemas Operacionais

- trilhas e setores são definidos por formatação física pelo fabricante
- Um cilindro representa uma trilha em todas as superfícies

4D > 4A > 4B > 4B > B 990

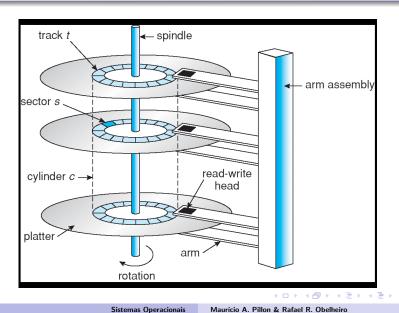
4□ > 4同 > 4 = > 4 = > ■ 900

Organização dos discos Desempenho do disco Escalonamento de disco

Introdução Princípios de hardware de E/S Princípios do software de E/S Camadas do software de E/S E/S no Linux

Organização dos discos Desempenho do disco Escalonamento de disco

### Organização dos discos (2)



#### Formato dos setores

- Cada setor possui o seguinte formato
  - preâmbulo: marca o início do setor
  - dados: geralmente em blocos de 512 bytes
  - ECC: um código de correção de erros (Reed-Solomon)
    - permite corrigir erros de leitura em alguns bits de dados

Preâmbulo	Dados	ECC	l
-----------	-------	-----	---

< □ > < 🗇 >	$\leftarrow \equiv +$	∢ ≣ →	=	200
-------------	-----------------------	-------	---	-----

Princípios de hardware de E/S Princípios do software de E/S Camadas do software de E/S Discos E/S no Linux

Organização dos discos Desempenho do disco Escalonamento de disco

Sistemas Operacionais Princípios de hardware de E/S Princípios do software de E/S Camadas do software de E/S Discos E/S no Linux

Organização dos discos Desempenho do disco Escalonamento de disco

Maurício A. Pillon & Rafael R. Obelheiro

#### Exemplos de parâmetros de disco

Parâmetro	Disco flexível IBM 360 KB	Disco rígido WD 18300
Número de cilindros	40	10 601
Trilhas por cilindro	2	12
Setores por trilha	9	281 (avg)
Setores por disco	720	35 742 000
Bytes por setor	512	512
Capacidade do disco	360 KB	18,3 GB
Tempo de posicionamento (cilindros adjacentes)	6 ms	0,8 ms
Tempo de posicionamento (caso médio)	77 ms	6,9 ms
Tempo de rotação	200 ms	8,33 ms
Tempo de pára/inicia do motor	250 ms	20 s
Tempo de transferência para um setor	22 ms	17 us

- tempo de posicionamento diminuiu 7 vezes
- taxa de transferência aumentou 1300 vezes
- capacidade aumentou 50 mil vezes

#### Acesso a dados

- Para acessar dados, é preciso localizar o bloco desejado
  - superfície, trilha, setor
- Existem dois métodos básicos de endereçamento
  - CHS (Cylinder, Head, Sector): usando as componentes
  - LBA (Linear Block Addressing): usando um no. de bloco
- O método mais usado atualmente é LBA
  - controlador é responsável por converter um número de bloco lógico nos parâmetros físicos adequados
  - permite mascarar detalhes como
    - variações no número de setores por trilha: zonas
    - setores defeituosos

4 ロ ト 4 周 ト 4 里 ト 4 里 ト 9 9 0

45/72

#### Desempenho do disco

- Para ler ou escrever dados é necessário posicionar corretamente a cabeça de leitura e gravação
- O tempo da operação é dado por três componentes
  - tempo de posicionamento (seek): tempo necessário para levar o braço até a trilha desejada
  - latência rotacional: tempo necessário para que o setor desejado passe sob o cabeçote
  - tempo de transferência: tempo necessário para transferir efetivamente os dados
- Tempo de posicionamento domina

$$t_{acesso} = t_{seek} + t_{lat} + t_{transf}$$

(	Como	de	terr	nınar	os	componente	es!	(T	)

- O tempo de seek é uma característica do disco, e tipicamente é fornecido pelo fabricante
  - muitas vezes se trabalha com um valor médio
- A latência rotacional depende da velocidade de rotação e de qual setor está passando embaixo do cabeçote quando o braço chega à trilha desejada
  - no melhor caso o setor desejado é o próximo

Sistemas Operacionais

- no pior caso, o setor desejado acabou de passar
- na média, a latência é dada pelo tempo de meia rotação

Maurício A. Pillo

Organização dos discos

Desempenho do disco

∢ □	▶ 4 🗗	<b>▶</b> ∢	≣ →	< ≣ →	₽	200

on &	& Rafael	R. Obelhei	о			50/72
		4 00 1 4	= *	1 = 1	= `	r) q (*

Organização dos discos Desempenho do disco Escalonamento de disco

Maurício A. Pillon & Rafael R. Obelheiro

Introdução
ps discos
Princípios de hardware de E/S
po disco
de disco
Camadas do software de E/S
Discos
E/S no Linux

49/72

### Como determinar os componentes? (2)

• O tempo de transferência depende da velocidade de rotação e do tamanho e densidade dos setores

$$t_{transf} = \frac{b}{r \cdot N}$$

- b é a quantidade de bytes em um setor
- r é a velocidade de rotação (rps)
- N é a quantidade de bytes em uma trilha

### Tempo médio de acesso

• O tempo médio de acesso é dado então por

$$t_{acesso} = t_{seek} + \frac{1}{2r} + \frac{b}{r \cdot N}$$

### Cálculo do tempo médio (1)

 Qual o tempo médio necessário para ler um setor de um disco com os seguintes parâmetros?

• tempo de seek médio: 10 ms

• velocidade de rotação: 10.000 rpm

bytes/setor: 512bytes/trilha: 32.768

$$t_{acesso} = 0.01 + \frac{1}{2 \times \frac{10000}{60}} + \frac{512}{\frac{10000}{60} \times 32768} = 13,09 \, \text{ms}$$

#### Cálculo do tempo médio (2)

- Qual o tempo médio necessário para ler cinco setores consecutivos do disco usado como exemplo? (slide 47)
- Parâmetros:

tempo de seek médio: 6,9 ms
tempo de rotação: 8,33 ms

• bytes/setor: 512

• setores/trilha: 281

$$t_{acesso} = 6.9 + rac{8.33}{2} + 8.33 imes rac{5 imes 512}{281 imes 512} = 11.21 \, ext{ms}$$

- 1	Maurício A. Pillon	&	Rafa	el F	. Ob	elhei	ro						54/72
		4		4		• •	Ξ	•	•	₽ >	-	· •	) d (4

Sistemas Operacionais
Introdução
Princípios de hardware de E/S
Princípios do software de E/S
Camadas do software de E/S
Discos
E/S no Linux

Maurício A. Pillon & Rafael R. Obelheiro

4□ → 4同 → 4 ■ → ■ 900

4□ → 4周 → 4 = → 4 = → 9 Q ○

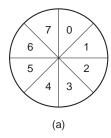
Organização dos discos Desempenho do disco Escalonamento de disco Sistemas Operacionais
Introdução
Princípios de hardware de E/S
Princípios do software de E/S
Camadas do software de E/S
Discos
E/S no Linux

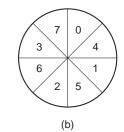
Organização dos discos Desempenho do disco Escalonamento de disco

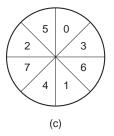
### Entrelaçamento (1)

- Alguns discos n\u00e3o conseguem ler setores adjacentes de forma consecutiva
  - devido à transferência do bloco de dados do buffer do controlador para a memória, o próximo setor passa sob o cabeçote antes que a leitura possa iniciar
  - isso implica uma rotação adicional para cada setor
- Uma solução é introduzir setores entre setores logicamente adjacentes
  - dá tempo para que o próximo setor seja lido

### Entrelaçamento (2)







- (a) sem entrelaçamento
- (b) entrelaçamento simples
- (c) entrelaçamento duplo

#### Escalonamento de disco

- Como o tempo de posicionamento domina o tempo de acesso, reduções nesse tempo melhoram o desempenho dos acessos a disco
- Embora o tempo de posicionamento seja uma característica do disco, é possível mudar a ordem em que as requisições de leitura e escrita são atendidas para minimizar os deslocamentos do braço
- Essa é a função dos algoritmos de escalonamento de disco
  - FCFS
  - SSF
  - elevador

## FCFS (First Come, First Served)

- As requisições são atendidas na ordem de chegada
- Consideremos um exemplo
  - disco tem 40 cilindros
  - está sendo atendida uma requisição no cilindro 11
  - chegam requisições para os cilindros 1, 36, 16, 34, 9 e 12 (nessa ordem)
  - a distância total percorrida para atender o conjunto de requisições é de 111 cilindros

Sistemas Operacionais Maurício A. Pillon & Rafael R. Obelheiro

Introdução
Princípios de hardware de E/S
Princípios do software de E/S
Organização dos discos

Camadas do software de E/S
Discos
E/S no Linux

Desempenho do disco
Escalonamento de disco

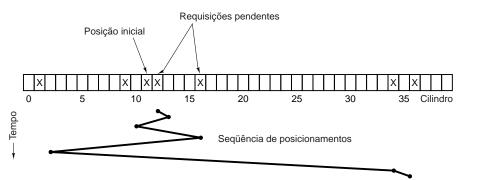
Introdução Princípios de hardware de E/S Princípios do software de E/S Camadas do software de E/S **Discos** E/S no Linux Maurício A. Pillon & Rafael R. Obelheiro 58/72

Organização dos discos
Desempenho do disco
Escalonamento de disco

イロト (個) (国) (国) (国)

#### SSF (Shortest Seek First)

- A requisição seguinte a ser atendida é aquela que estiver mais próxima do cilindro atual
- Algumas requisições podem levar muito tempo para serem atendidas
   concentração de requisições em uma região do disco
- Exemplo

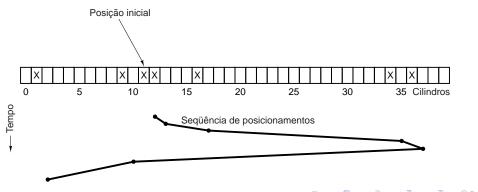


#### Algoritmo do elevador

• Atende a todas as requisições em uma direção

Sistemas Operacionais

- Quando chega ao final, muda a direção e atende às demais requisições
- Exemplo (inicialmente subindo)



イロト イ御ト イヨト イヨト 一ヨー

#### Sumário

- Introdução
- 2 Princípios de hardware de E/S
- 3 Princípios do software de E/S
- 4 Camadas do software de E/S
- Discos
- 6 E/S no Linux

#### Acesso a dispositivos de E/S

- No UNIX/Linux, cada dispositivo de E/S pode ser acessado por um arquivo especial
  - geralmente em /dev
    - /dev/sda: primeiro disco SCSI/SATA
    - /dev/psaux: porta de mouse PS/2
  - uma aplicação pode usar as mesmas chamadas de sistema válidas para arquivos
    - open(), read(), write(), close(), ...
- Existem duas grandes categorias de arquivos especiais
  - de bloco: disco, fita
  - de caracter: teclado, mouse, impressora, rede
- Cada arquivo especial possui dois números de dispositivo
  - **principal** (*major*): identifica o driver
  - secundário (minor): identifica o dispositivo (dentre vários)



4□ → 4同 → 4 = → → ■ 900

61/72 Maurício A. Pillon & Rafael R. Obelheiro

Conceitos fundamentais Escalonamento de disco

Sistemas Operacionais Princípios de hardware de E/S Princípios do software de E/S Camadas do software de E/S E/S no Linux

Maurício A. Pillon & Rafael R. Obelheiro

Maurício A. Pillon & Rafael R. Obelheiro

◆□▶ ◆□▶ ◆■▶ ◆■ ● のQ○

62/72

64/72

Conceitos fundamentais Escalonamento de disco

#### Escalonamento de disco no Linux

Sistemas Operacionais

Discos E/S no Linux

Princípios de hardware de E/S

Princípios do software de E/S

Camadas do software de E/S

- Funções do escalonador de disco
  - fundir requisições: quando uma requisição é recebida, o escalonador verifica se existe uma requisição adjacente; se houver, as duas requisições são combinadas em uma
    - request coalescing
    - minimiza seeks
  - ordenar requisições: ordena as requisições com o propósito de minimizar o tempo de seek e evitar inanição
- Princípios gerais
  - cada dispositivo de bloco tem uma (ou várias) filas de requisições
  - cada dispositivo tem uma fila de despacho (dispatch), que contém as requisições enviadas ao dispositivo
  - um escalonador de disco manipula fila(s) de requisições pendentes para alimentar a fila de despacho
    - em algoritmos simples, a fila de requisições pendentes e a fila de despacho são uma só

#### Escalonadores de disco no Linux

- Elevador do Linus
- Escalonador com deadlines
- Escalonador antecipatório
- Escalonador com enfileiramento completamente justo
- Escalonador noop
- O escalonador em uso no dispositivo DEV pode ser examinado ou modificado via /sys/block/DEV/queue/scheduler

#### Exemplo

- \$ cat /sys/block/sda/queue/scheduler noop deadline [cfq]
- # echo deadline >/sys/block/sda/queue/scheduler

Conceitos fundamentais Escalonamento de disco Introdução
Princípios de hardware de E/S
Princípios do software de E/S
Camadas do software de E/S
Discos
E/S no Linux

Conceitos fundamentais Escalonamento de disco

#### O elevador do Linus

- Usado no kernel 2.4
- Realiza ordenação e fusão de requisições, da seguinte forma:
  - 1. Se uma requisição para um setor adjacente já se encontra na fila, a requisição existente e a nova são fundidas
    - fusão pode ser recursiva, até um limite de blocos por requisição
  - 2. Se uma requisição na fila for suficientemente antiga, a nova requisição entra no final da fila
    - tenta evitar inanição de requisições, mas não funciona muito bem
  - 3. Caso contrário, insere a requisição na fila seguindo o algoritmo do elevador

#### Inanição de requisições

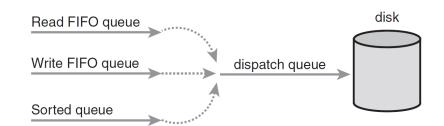
- O elevador do Linus está sujeito a inanição
- Concentração de requisições em uma região do disco
- Escritas causando inanição de leituras
  - operações de escrita são geralmente assíncronas
    - dados são copiados para um buffer no kernel antes de serem gravados no disco
    - a aplicação não fica esperando o resultado
    - leituras podem usar o buffer
  - operações de leitura são geralmente síncronas
    - aplicação fica bloqueada esperando os dados do disco
  - requisições de leitura tendem a ser encadeadas
    - uma requisição precisa esperar a anterior
  - latência de leitura afeta mais perceptivelmente o desempenho do que latência de escrita

		: ► <b>= *</b> ) <b>Q</b> ( <b>*</b>			<b>≡</b> Ψ) Q (Ψ
Sistemas Operacionais	Maurício A. Pillon & Rafael R. Obelheiro	65/72	Sistemas Operacionais	Maurício A. Pillon & Rafael R. Obelheiro	66/72
Introdução Princípios de hardware de E/S Princípios do software de E/S Camadas do software de E/S Discos E/S no Liny	Conceitos fundamentais Escalonamento de disco		Introdução Princípios de hardware de E/S Princípios do software de E/S Camadas do software de E/S Discos F/S no Linux	Conceitos fundamentais Escalonamento de disco	

#### O escalonador com deadlines

- Introduzido no kernel 2.6
- Objetivo é reduzir os problemas de inanição do elevador do Linus
- Cada requisição possui um tempo de expiração
  - default: 500 ms para leituras, 5 s para escritas
- Mantém três filas de requisições
  - fila ordenada (como no elevador)
  - fila FIFO de leituras
  - fila FIFO de escritas
- Uma requisição é inserida na fila ordenada e no final da fila FIFO correspondente
- Geralmente, requisições são passadas da fila ordenada para a fila de despacho
  - se a requisição no início de uma das filas FIFO tiver expirado, ela é colocada na fila de despacho

#### O escalonador com deadlines



4□ → 4同 → 4 = → → ■ 900

Princípios de hardware de E/S
Princípios do software de E/S
Camadas do software de E/S
Camadas do software de E/S
E/S no Linux

Introdução
Conceitos fundamentais
Escalonamento de disco

Introdução
Princípios de hardware de E/S
Princípios do software de E/S
Camadas do software de E/S
Discos

Conceitos fundamentais Escalonamento de disco

#### O escalonador antecipatório

- O escalonador com deadlines pode incorrer em seeks longos devido à expiração de requisições
  - se o sistema está num ciclo de escritas em uma região do disco e uma requisição de leitura expira, é necessário levar o braço até a trilha da leitura e depois retornar para as escritas
  - se as requisições de leitura forem encadeadas, isso pode ocorrer diversas vezes
  - vazão global do sistema é prejudicada
- O escalonador antecipatório adiciona uma heurística para minimizar o problema
  - quando uma requisição de leitura é atendida, o escalonador espera um tempo (default: 6 ms) antes de retornar à ordem normal de atendimento
  - se outra requisição de leitura para uma região adjacente chegar durante o tempo de espera, ela é atendida imediatamente
- Requer estatísticas por processo para antecipação funcionar bem

### Escalonador com enfileiramento completamente justo

- Completely fair queueing (CFQ)
- Cada processo tem uma fila de requisições pendentes
  - uma nova requisição é ordenada e fundida com as requisições pendentes do mesmo processo
- O escalonador atende as filas em round-robin
  - um número de requisições de um processo (default: 4) são atendidas, e depois o escalonador escolhe a fila de outro processo
- Cada processo obtém uma fatia justa do disco
- Escalonador default no kernel 2.6

				200
istemas Operacionais	Maurício A. Pillon & Rafael R. Obelheiro 69/72	Sistemas Operacionais	Maurício A. Pillon & Rafael R. Obelheiro	70/72
Introdução de hardware de E/S s do software de E/S s do software de E/S Discos E/S no Linux	Conceitos fundamentais Escalonamento de disco	Introdução Princípios de hardware de E/S Princípios do software de E/S Camadas do software de E/S Discos <b>E/S no Linux</b>	Conceitos fundamentais Escalonamento de disco	

Bibliografia básica

#### Escalonador noop

• Noop  $\rightarrow$  No operation

Princípios

- Realiza apenas fusão de requisições adjacentes
- Usado em dispositivos de acesso realmente direto  $(t_{seek} \rightarrow 0)$ 
  - pendrives e outras memórias de estado sólido



Abraham Silberchatz, Greg Gagne e Peter Baer Galvin. Fundamentos de Sistemas Operacionais, 6ª Edição. LTC - Livros Técnicos e Científicos Editora S.A., 2004.

Sistemas Operacionais