

Sistema de Arquivos

Implementação

vsfs – Very Simple File System

- Estruturas básicas
- Métodos de acesso
- Políticas

Modelo Mental

- Que estruturas em disco armazenam dados e metadados do Sistema de Arquivo?
- O que acontece quando um processo abre um arquivo?
- Quais estruturas em disco são acessadas durante um `read` ou `write`?

Modelo Mental

- Que estruturas em disco armazenam dados e metadados do Sistema de Arquivo?

- O que acontece quando um processo abre um arquivo?

- Quais estruturas em disco são acessadas durante um `read` ou `write`?

Modelo Mental

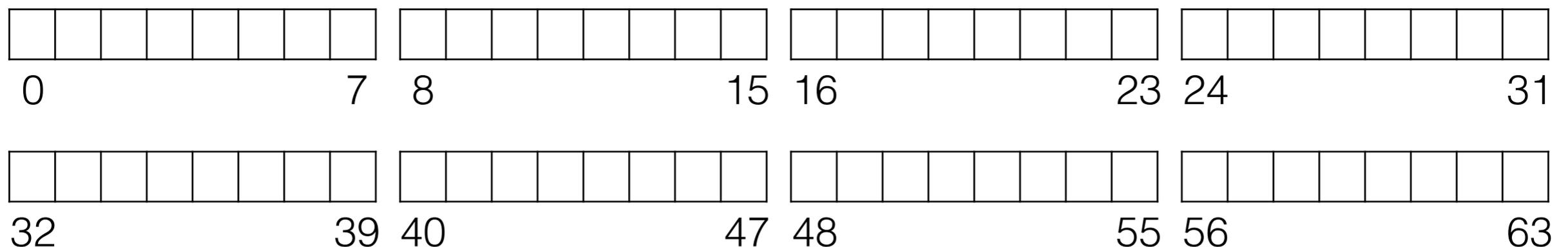
- Que estruturas em disco armazenam dados e metadados do Sistema de Arquivo?
- O que acontece quando um processo abre um arquivo?

- Quais estruturas em disco são acessadas durante um `read` ou `write`?

“If you understand the **data structures** and **access methods** of a file system, you have developed a good mental model of how it truly works”

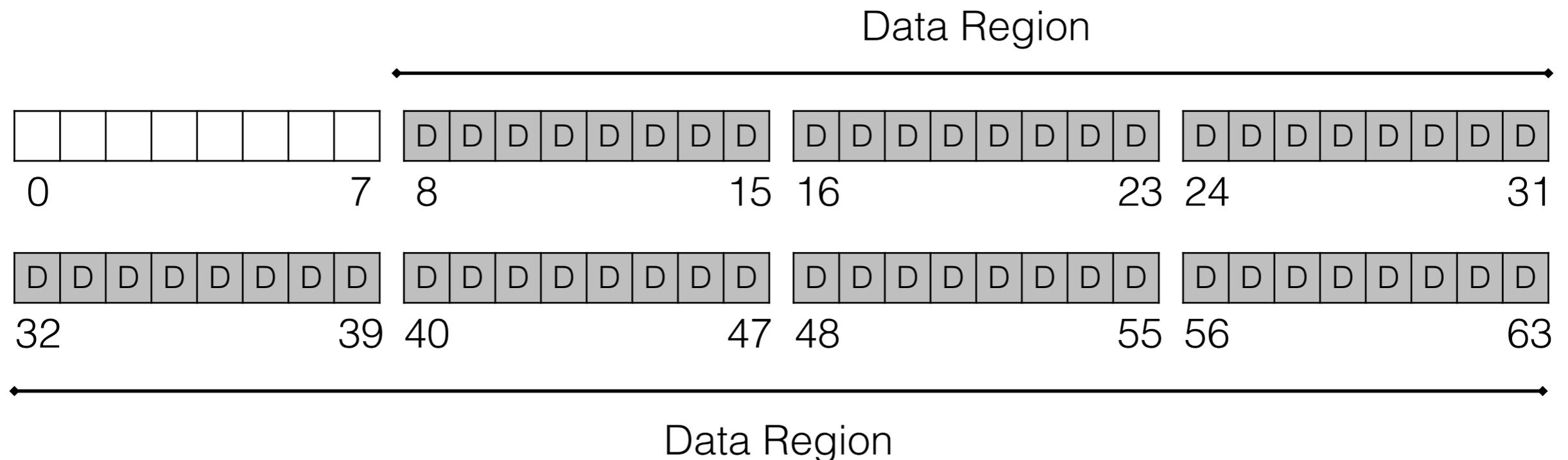
Organização Geral

- Organização em disco das estruturas de dados
 - Divide o disco em **blocos** – tamanho comumente usado? 4KB
 - Visão da **partição**: uma série de blocos de 4KB
 - Os blocos são endereçados de 0 a N-1, em uma partição de N blocos (de 4KB)



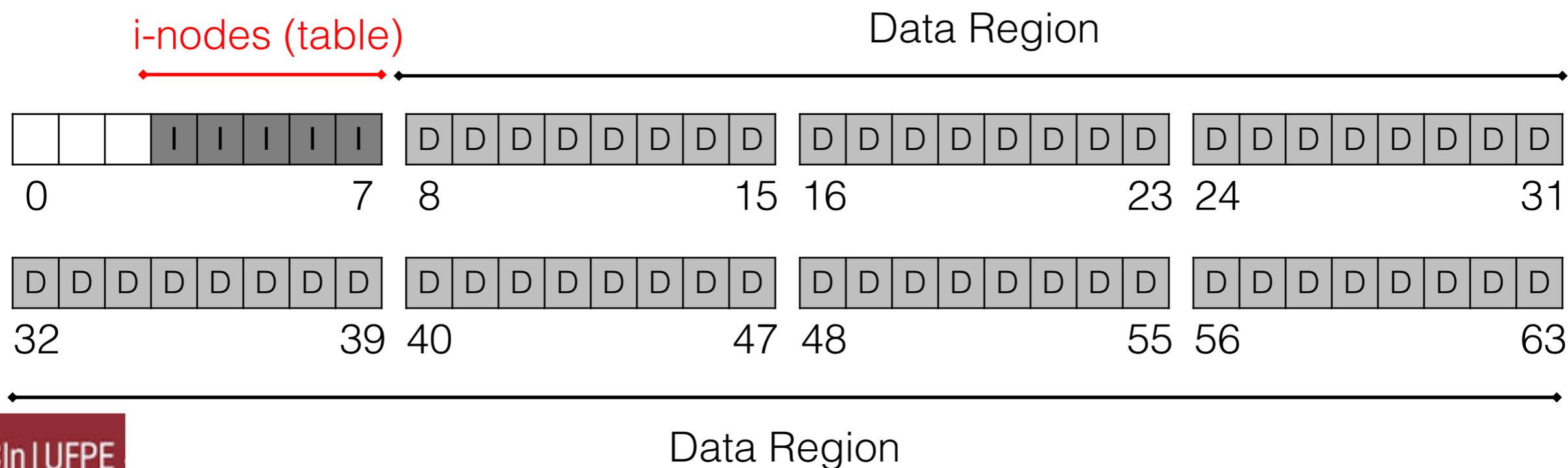
Região de Dados

- O que primeiro pensamos em armazenar nestes blocos?
 - **Dados de usuário**



i-nodes

- Estrutura para armazenar **metadados** a respeito dos arquivos
 - Quais blocos de dados formam um arquivo
 - Tamanho do arquivo
 - Proprietário e direitos de acesso
 - Tempos de acesso, atualização etc.
- **Tabela de i-nodes** = vetor de i-nodes



Metadados de um arquivo

```
exercicios — sh — 82x24
[sh-3.2# ls -l
total 232
-rw-r--r--@ 1 cagf  staff  10939 May 15 14:11 Infra de Software - CC - 2018.1 - No
tas da lista 1.xlsx
-rw-r--r--@ 1 cagf  staff  47894 Apr 13 14:20 ListaAtividadesInfraSW.pdf
-rwxr-xr-x  1 root  staff   8564 Apr  2 08:52 pratica
-rw-r--r--  1 root  staff   146 Apr  2 08:51 pratica.c
-rw-r--r--@ 1 cagf  staff  39121 Mar 13 15:41 propostas_ExercicioConcorrencia.pdf
[sh-3.2# stat -x pratica
File: "pratica"
Size: 8564          FileType: Regular File
Mode: (0755/-rwxr-xr-x)  Uid: (  0/   root)  Gid: ( 20/   staff)
Device: 1,4  Inode: 8595721149  Links: 1
Access: Tue May 15 11:24:57 2018
Modify: Mon Apr  2 08:52:00 2018
Change: Mon Apr  2 08:52:00 2018
sh-3.2#
```

Metadados de um arquivo

```
exercicios — sh — 82x24
[sh-3.2# ls -l
total 232
-rw-r--r--@ 1 cagf  staff  10939 May 15 14:11 Infra de Software - CC - 2018.1 - No
tas da lista 1.xlsx
-rw-r--r--@ 1 cagf  staff  47894 Apr 13 14:20 ListaAtividadesInfraSW.pdf
-rwxr-xr-x  1 root  staff   8564 Apr  2 08:52 pratica
-rw-r--r--  1 root  staff   146 Apr  2 08:51 pratica.c
-rw-r--r--@ 1 cagf  staff  39121 Mar 13 15:41 propostas_ExercicioConcorrencia.pdf
[sh-3.2# stat -x pratica
File: "pratica"
Size: 8564      FileType: Regular File
Mode: (0755/-rwxr-xr-x)  Uid: (  0/   root)  Gid: ( 20/   staff)
Device: 1,4  Inode: 8595721149  Links: 1
Access: Tue May 15 11:24:57 2018
Modify: Mon Apr  2 08:52:00 2018
Change: Mon Apr  2 08:52:00 2018
sh-3.2#
```

Metadados de um arquivo

```
exercicios — sh — 82x24
[sh-3.2# ls -l
total 232
-rw-r--r--@ 1 cagf  staff  10939 May 15 14:11 Infra de Software - CC - 2018.1 - No
tas da lista 1.xlsx
-rw-r--r--@ 1 cagf  staff  47894 Apr 13 14:20 ListaAtividadesInfraSW.pdf
-rwxr-xr-x  1 root  staff   8564 Apr  2 08:52 pratica
-rw-r--r--  1 root  staff   146 Apr  2 08:51 pratica.c
-rw-r--r--@ 1 cagf  staff  39121 Mar 13 15:41 propostas_ExercicioConcorrencia.pdf
[sh-3.2# stat -x pratica
  File: "pratica"
  Size: 8564          FileType: Regular File
  Mode: (0755/-rwxr-xr-x)  Uid: (  0/   root)  Gid: ( 20/   staff)
Device: 1,4    Inode: 8595721149    Links: 1
Access: Tue May 15 11:24:57 2018
Modify: Mon Apr  2 08:52:00 2018
Change: Mon Apr  2 08:52:00 2018
sh-3.2#
```

Metadados de um arquivo

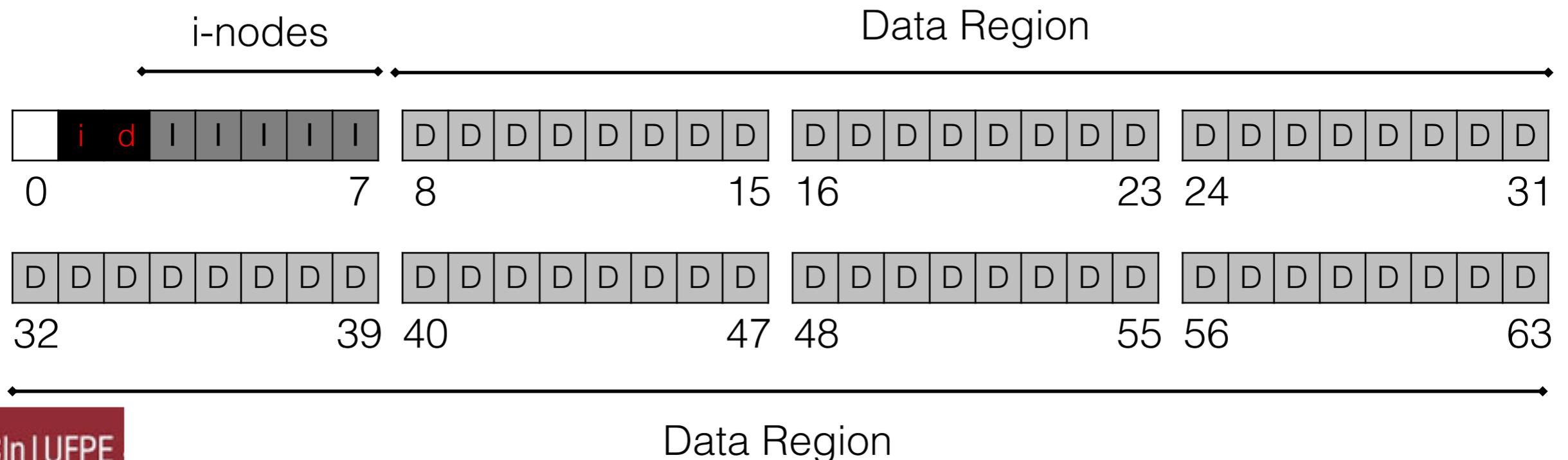
```
exercicios — sh — 82x24
[sh-3.2# ls -l
total 232
-rw-r--r--@ 1 cagf  staff  10939 May 15 14:11 Infra de Software - CC - 2018.1 - No
tas da lista 1.xlsx
-rw-r--r--@ 1 cagf  staff  47894 Apr 13 14:20 ListaAtividadesInfraSW.pdf
-rwxr-xr-x  1 root  staff   8564 Apr  2 08:52 pratica
-rw-r--r--  1 root  staff   146 Apr  2 08:51 pratica.c
-rw-r--r--@ 1 cagf  staff  39121 Mar 13 15:41 propostas_ExercicioConcorrenca.pdf
[sh-3.2# stat -x pratica
  File: "pratica"
  Size: 8564          FileType: Regular File
  Mode: (0755/-rwxr-xr-x)  Uid: (  0/   root)  Gid: ( 20/   staff)
Device: 1,4  Inode: 8595721149  Links: 1
Access: Tue May 15 11:24:57 2018
Modify: Mon Apr  2 08:52:00 2018
Change: Mon Apr  2 08:52:00 2018
sh-3.2#
```

Metadados de um arquivo

```
exercicios — sh — 82x24
[sh-3.2# ls -l
total 232
-rw-r--r--@ 1 cagf  staff  10939 May 15 14:11 Infra de Software - CC - 2018.1 - No
tas da lista 1.xlsx
-rw-r--r--@ 1 cagf  staff  47894 Apr 13 14:20 ListaAtividadesInfraSW.pdf
-rwxr-xr-x  1 root  staff   8564 Apr  2 08:52 pratica
-rw-r--r--  1 root  staff    146 Apr  2 08:51 pratica.c
-rw-r--r--@ 1 cagf  staff  39121 Mar 13 15:41 propostas_ExercicioConcorrencia.pdf
[sh-3.2# stat -x pratica
  File: "pratica"
  Size: 8564          FileType: Regular File
  Mode: (0755/-rwxr-xr-x)  Uid: (  0/   root)  Gid: ( 20/   staff)
Device: 1,4  Inode: 8595721149  Links: 1
Access: Tue May 15 11:24:57 2018
Modify: Mon Apr  2 08:52:00 2018
Change: Mon Apr  2 08:52:00 2018
sh-3.2#
```

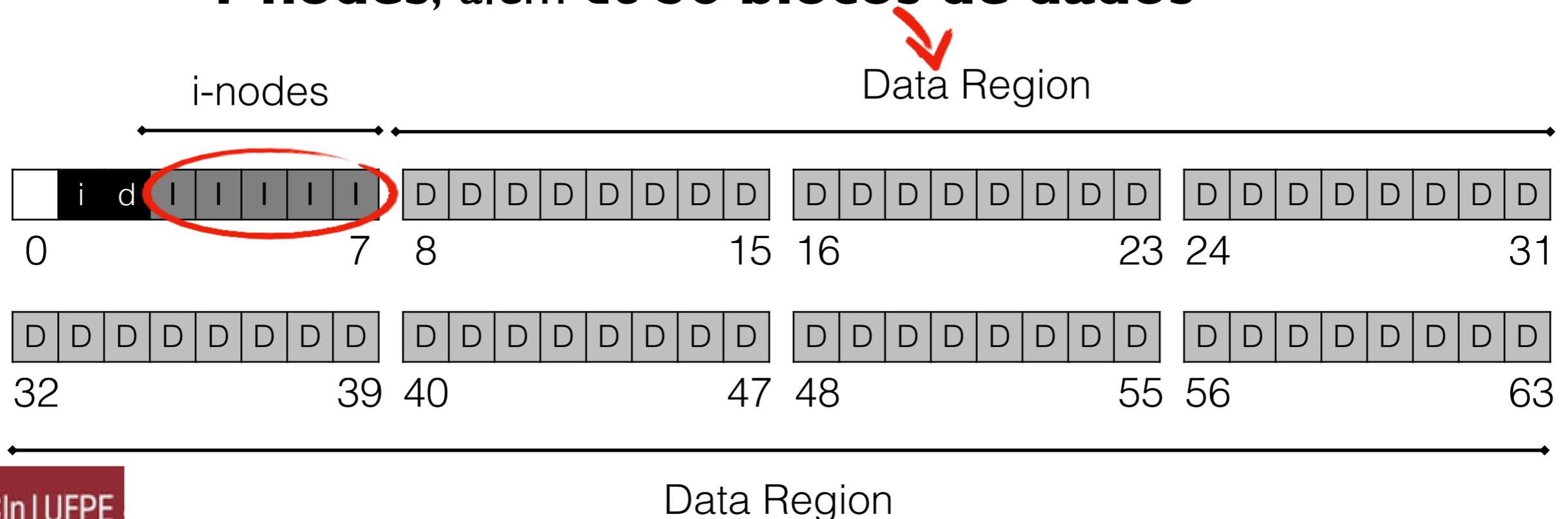
Estrutura de alocação

- Rastrear se i-nodes ou blocos de dados estão livres ou ocupados
 - Uma **freelist** aponta para o primeiro bloco livre, e assim por diante
 - Estrutura mais simples e popular: **bitmap**
 - Bitmap de i-nodes (**i**)
 - Bitmap de dados (**d**)



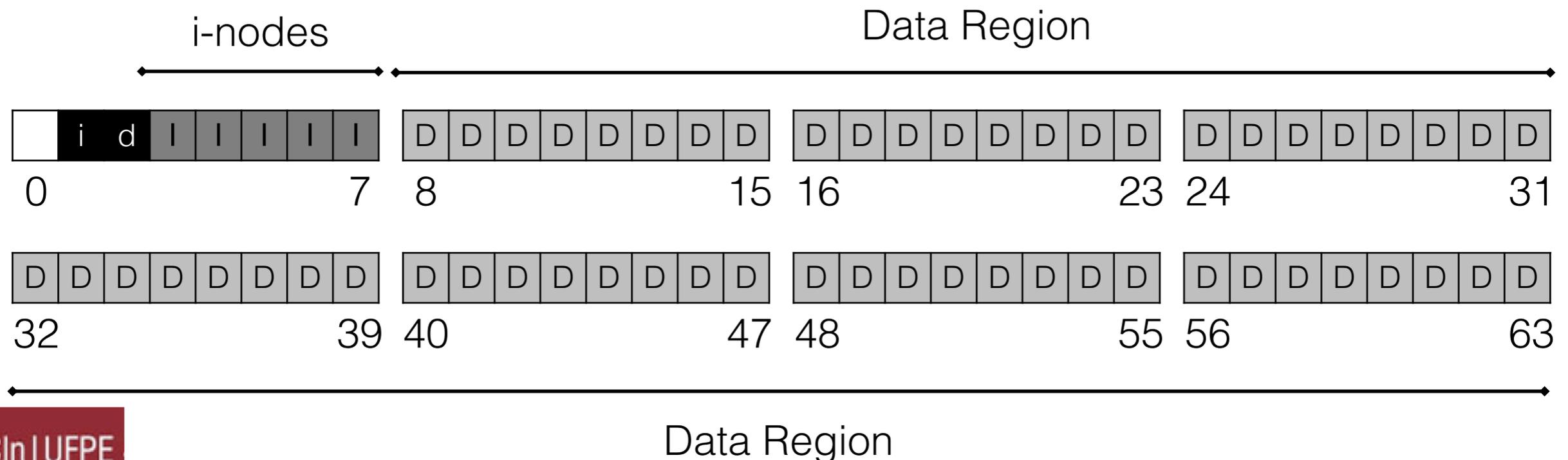
Observações

- i-nodes não são grandes
- Neste **vsfs** assumimos que um i-node possui 256 bytes
 - Assim, em um bloco de 4KB (4096 bytes) cabem 16 i-nodes: $4096 / 256 = 16$
 - Ou seja, **vsfs** possui **5 blocos de i-nodes** x 16 = **80 i-nodes**, além de **56 blocos de dados**



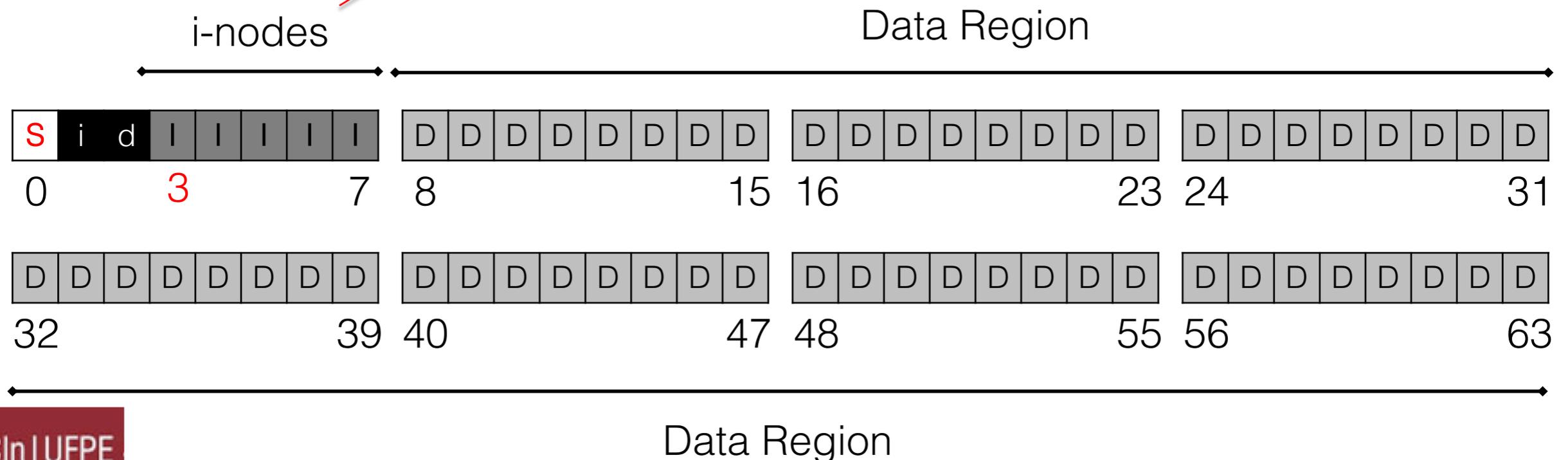
Observações (2)

- É um exagero usar um bloco inteiro (4KB) para representar os bitmaps **i** e **d**, pois cada **bit**map pode rastrear se 32K (4KB = 32K**b**) objetos estão alocados
 - mas neste **vsfs** só existem 80 i-nodes e 56 blocos de dados – muito menos do que 32768 i-nodes e 32768 blocos de dados, respectivamente
 - isto está sendo feito apenas por simplicidade – um bloco para cada bitmap (i,d)!



O superbloco

- **S** contém informações sobre o Sistema de Arquivo
 - Um **número mágico** para identificar o **tipo do sistema de arquivo** (neste caso, **vsfs**)
 - O **número de i-nodes** e o **número de blocos de dados** que há no sistema de arquivo (neste caso, 80 e 56, respectivamente)
 - Onde a **tabela de i-nodes** começa (neste caso, no **bloco 3**)
 - Etc...



Um vsfs: implementação parcial (até o momento...)

- Partição dividida em blocos
- Blocos de dados (de usuário)
- i-nodes: metadados dos arquivos
- Estruturas de alocação (ex. bitmaps)
- Superbloco: informações sobre o próprio Sistema de Arquivo

Infra-Estrutura de Software

Entrada / Saída

Diversidade de dispositivos

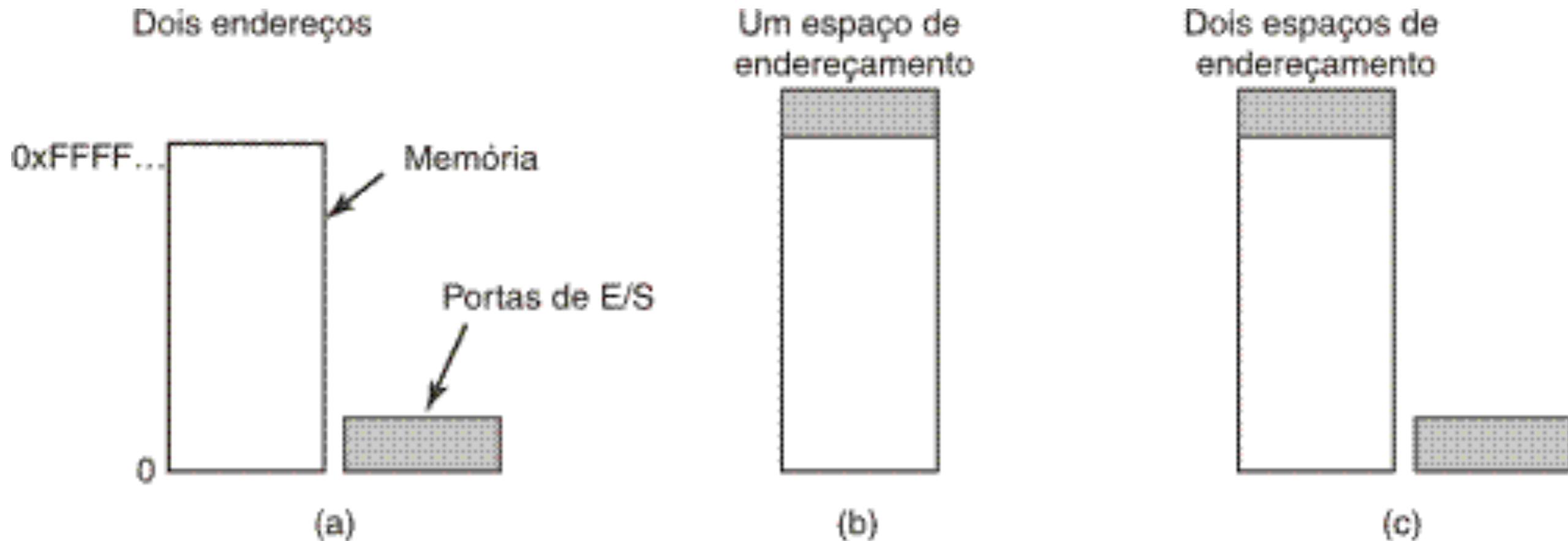
Hardware de E/S

Device	Data rate
Keyboard	10 bytes/sec
Mouse	100 bytes/sec
56K modem	7 KB/sec
Scanner at 300 dpi	1 MB/sec
Digital camcorder	3.5 MB/sec
4x Blu-ray disc	18 MB/sec
802.11n Wireless	37.5 MB/sec
USB 2.0	60 MB/sec
FireWire 800	100 MB/sec
Gigabit Ethernet	125 MB/sec
SATA 3 disk drive	600 MB/sec
USB 3.0	625 MB/sec
SCSI Ultra 5 bus	640 MB/sec
Single-lane PCIe 3.0 bus	985 MB/sec
Thunderbolt 2 bus	2.5 GB/sec
SONET OC-768 network	5 GB/sec

E/S: Como a CPU acessa a informação?

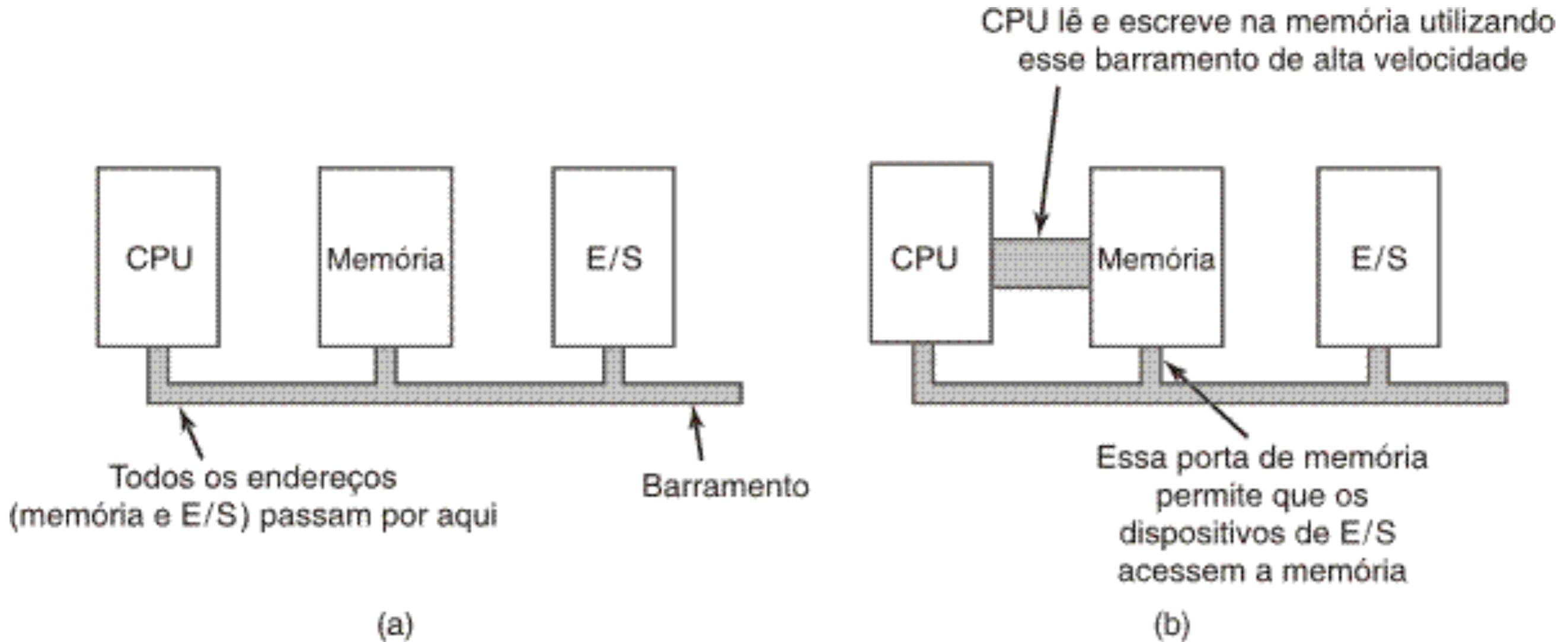
- ✓ **Espaço de endereçamento:** conjunto de endereços de memória que o processador consegue acessar diretamente
 - A forma de acessar os **registradores (das interfaces) dos periféricos** é definida no projeto do processador:
 - **Espaço único**
 - **Dois espaços**, um deles dedicado à E/S (**isolada**)
 - **E/S isolada**
 - Através de **instruções especiais** de E/S
 - Especifica a leitura/escrita de dados numa porta de E/S
 - **E/S mapeada em memória**
 - Através de **instruções de leitura/escrita** na memória
 - **Híbrido**
 - E/S mapeada em memória: memória de vídeo
 - E/S isolada: dispositivos em geral
- 

Espaços de Memória e E/S



- a) Espaços de memória e E/S separados - E/S isolada
- b) E/S mapeada na memória
- c) Híbrido

E/S mapeada na memória



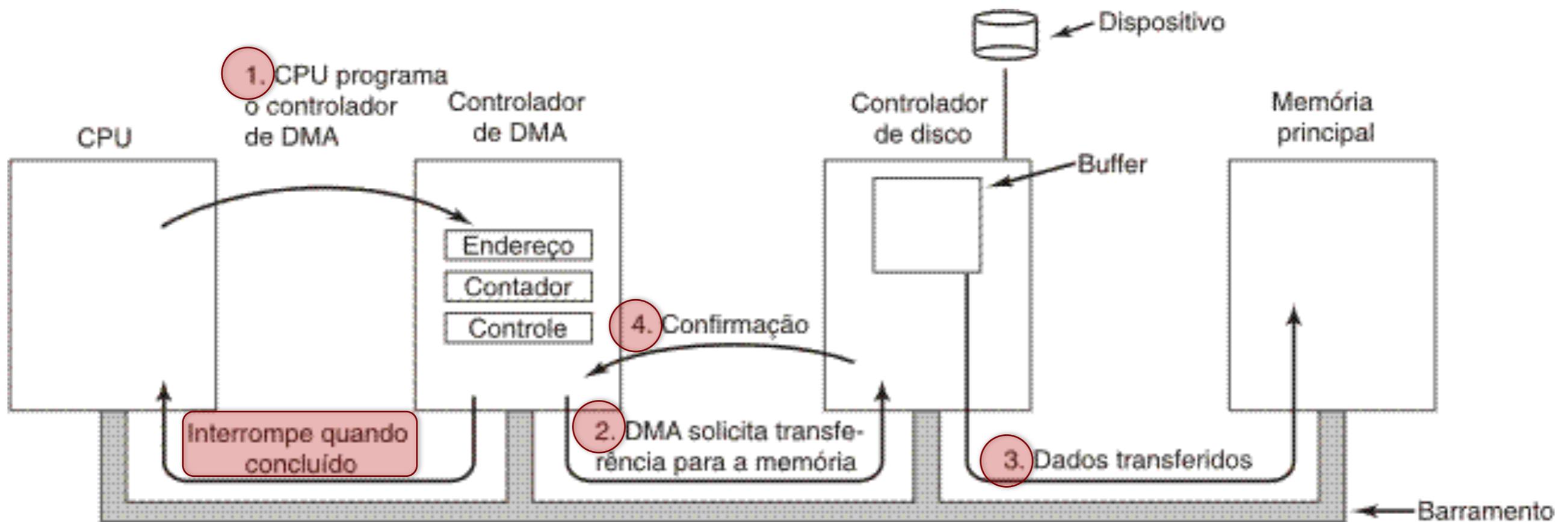
(a) Arquitetura com barramento único

(b) Arquitetura com barramento duplo (dual)

Como o processador “enxerga” a memória e os demais dispositivos (como o processador se comunica com o seu exterior)

- O processador realiza operações como:
 - Ler um dado da memória
 - Escrever um dado na memória
 - Receber (ler) um dado de dispositivos de E/S
 - Enviar (escrever) dados para dispositivos de E/S
- Nas operações de **acesso à memória**, o processador escreve e lê dados, **praticamente sem intermediários**
- Nos **acessos a dispositivos de E/S**, existem circuitos intermediários, que são as **interfaces**

Acesso Direto à Memória (DMA)



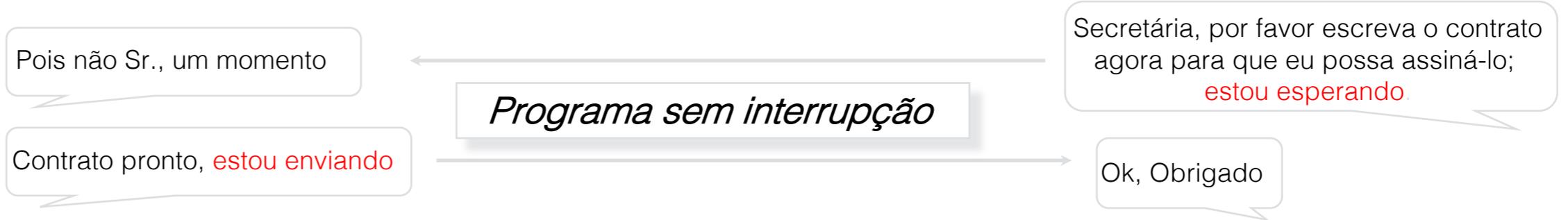
Operação de uma transferência com DMA

Como a CPU sabe que o dispositivo já executou o comando?

- **E/S Programada**
 - CPU lê constantemente o status do controlador e verifica se já acabou (*Polling* ou *Busy-waiting*)
 - Desvantagem: Espera até o fim da operação
- **E/S por Interrupção**
 - CPU é interrompida pelo módulo de E/S e ocorre transferência de dados
 - CPU continua a executar outras operações
 - Desvantagem: toda palavra lida do (ou escrita no) periférico passa pela CPU
- **E/S por DMA - Acesso Direto à Memória**
 - Quando necessário, o controlador de E/S solicita ao controlador de DMA a transferência de dados de/para a memória
 - Nesta fase de transferência não há envolvimento da CPU
 - Ao fim da transferência, a CPU é interrompida e informada da transação

Comunicação S.O.(CPU) – Controlador

Exemplo de comunicação com dispositivo



Entrada/Saída

- ✓ Princípios do hardware de E/S
 - Princípios do software de E/S
 - Camadas do software de E/S
 - Gerenciamento de energia

Objetivos da gerência de E/S

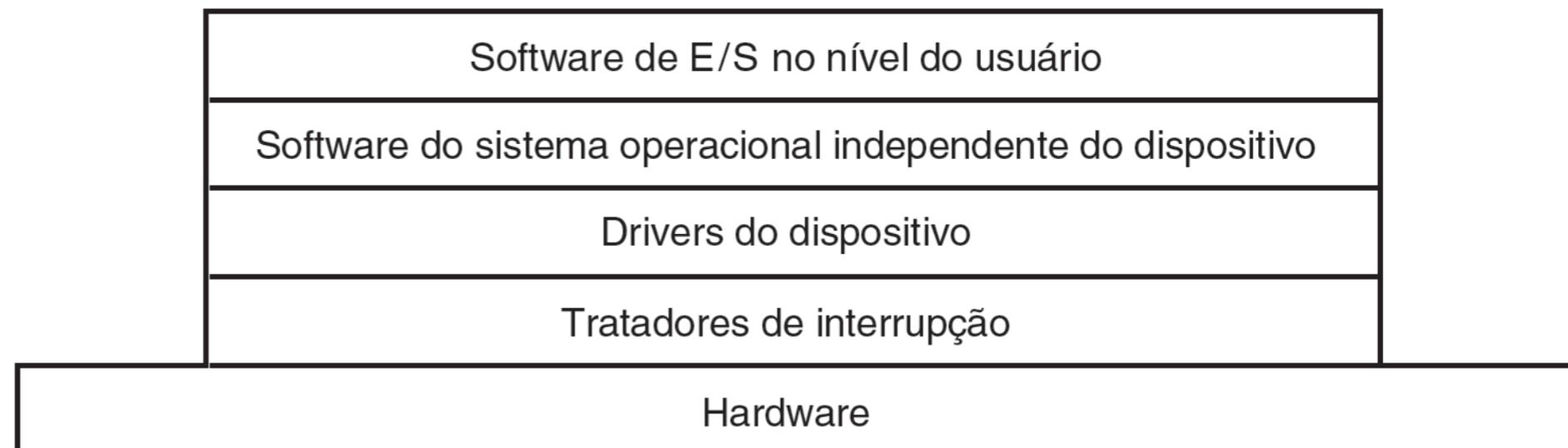
- Eficiência
- Uniformidade (desejável, diante da alta **diversidade**, associada à **heterogeneidade**):
 - Todos dispositivos enxergados da forma mais uniforme possível
- Esconder os detalhes (estes são tratados pelas camadas de mais baixo nível) – **abstrações**
- Fornecer outras **abstrações** genéricas: `read`, `write`, `open`, `close` etc.

Princípios básicos do software de E/S

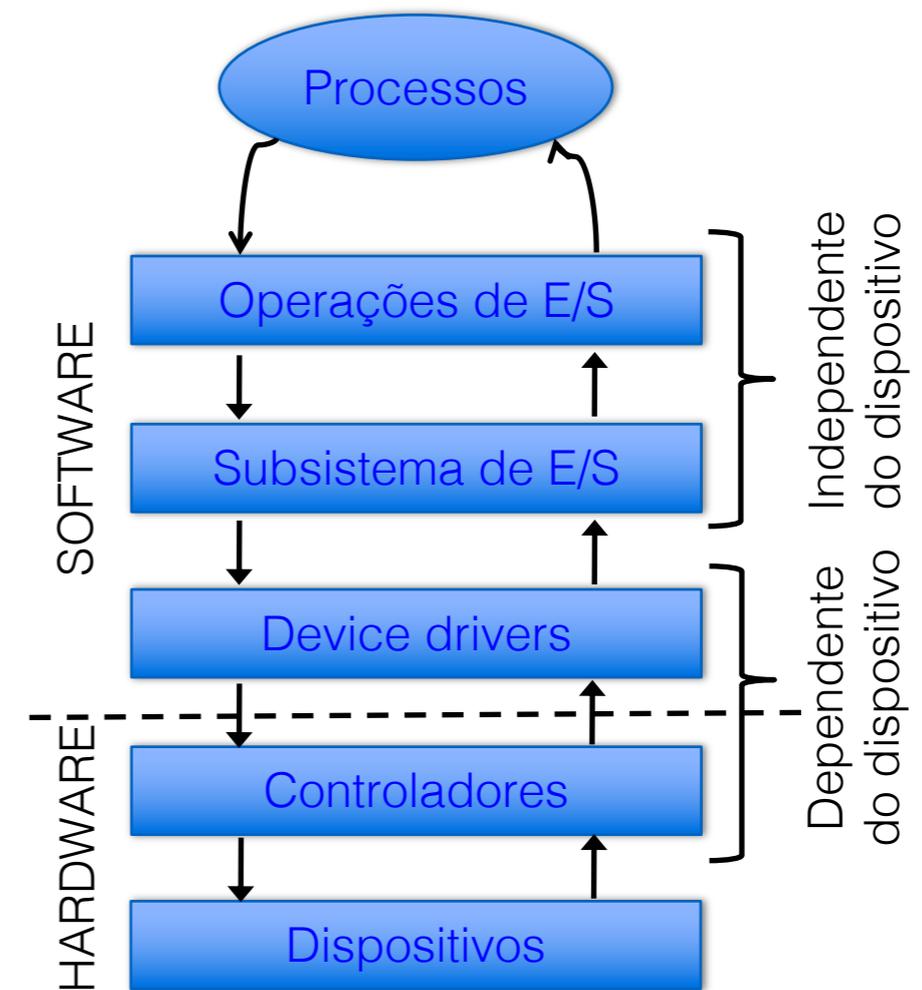
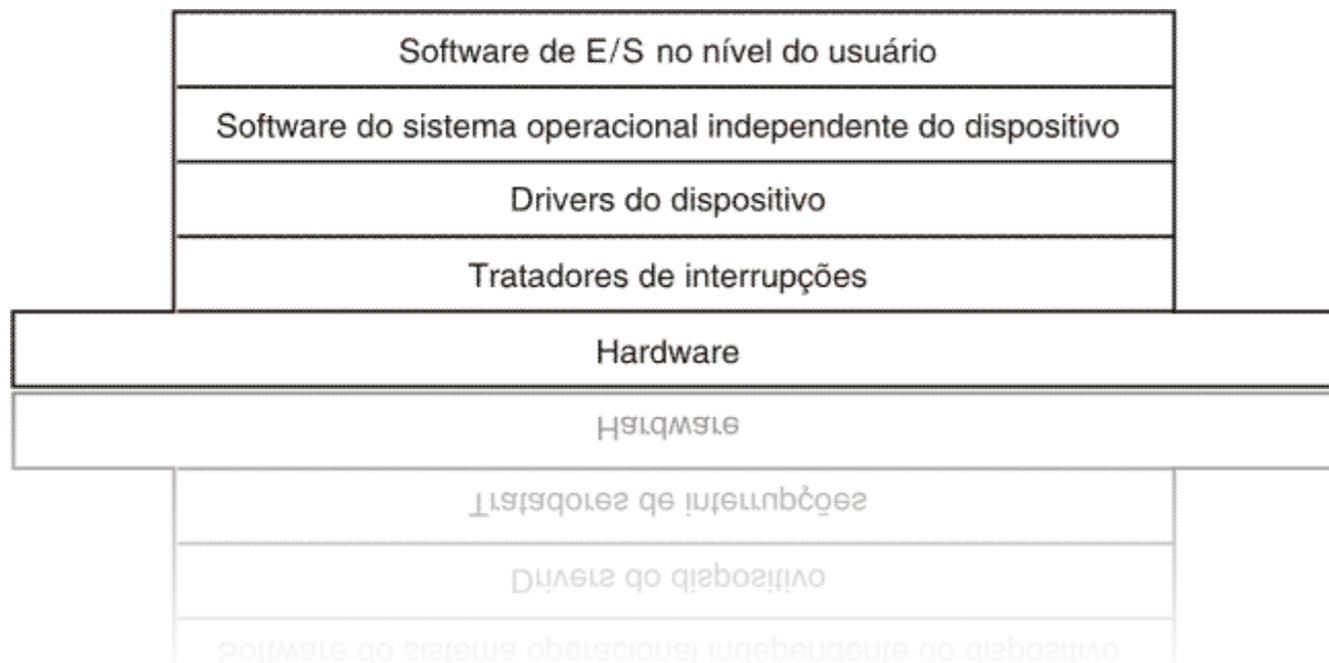
- Subsistema de E/S é complexo, dada a **diversidade** de periféricos
- Padronizar ao máximo para reduzir número de rotinas
 - **Novos dispositivos não alteram a visão do usuário em relação ao SO**
- Organizado em camadas

Visão Geral do software de E/S

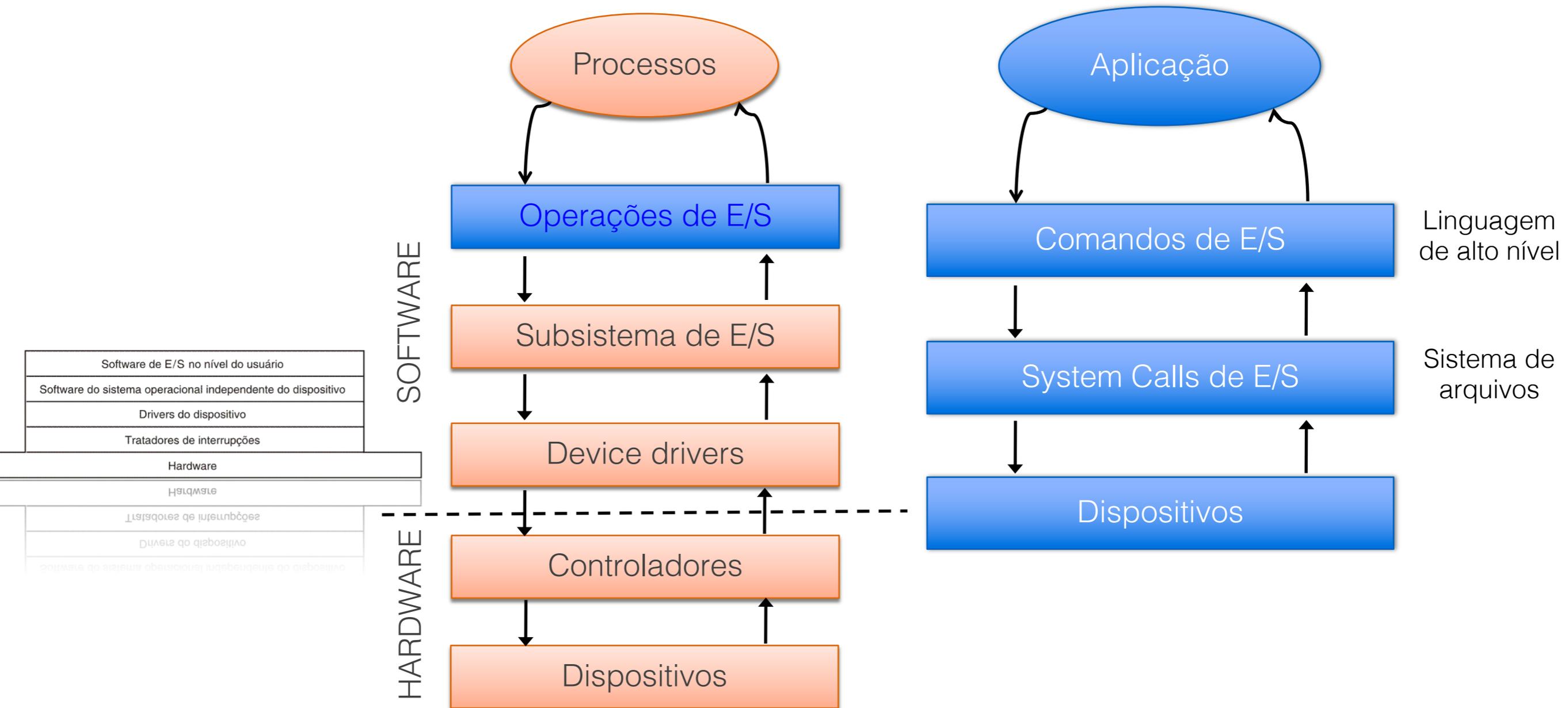
- Tratador de interrupção
 - É acionado ao final da operação de transferência
 - Aciona driver
- Driver de dispositivo
 - Recebe requisições
 - Configura (aciona) o controlador
- E/S independente de dispositivo
 - Nomes e proteção
 - bufferização
- E/S em nível de usuário
 - Chamadas de E/S (*System Calls*)



Camadas do Software de E/S



Camadas do Software de E/S

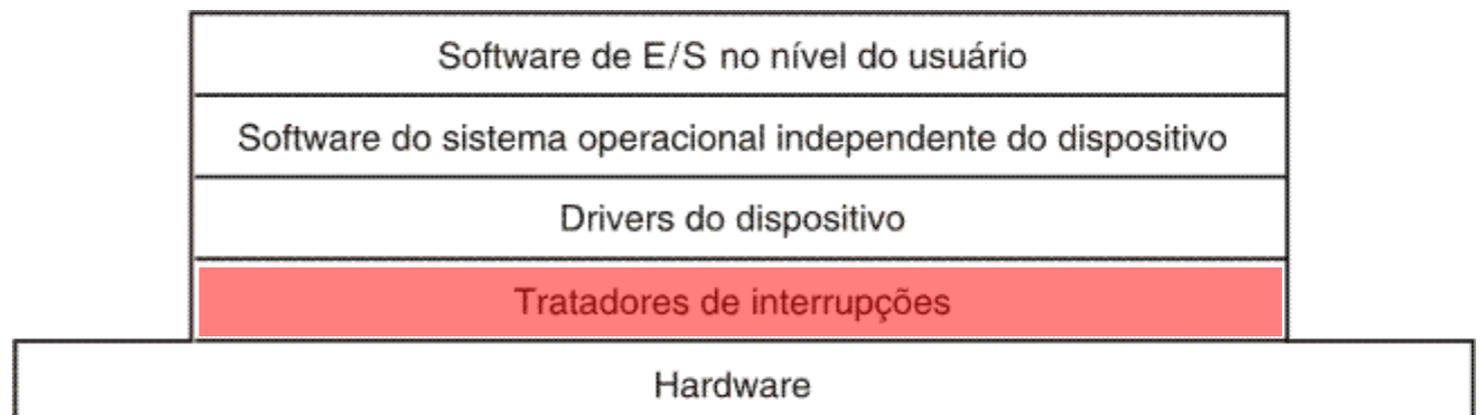


Tratador(es) de Interrupção

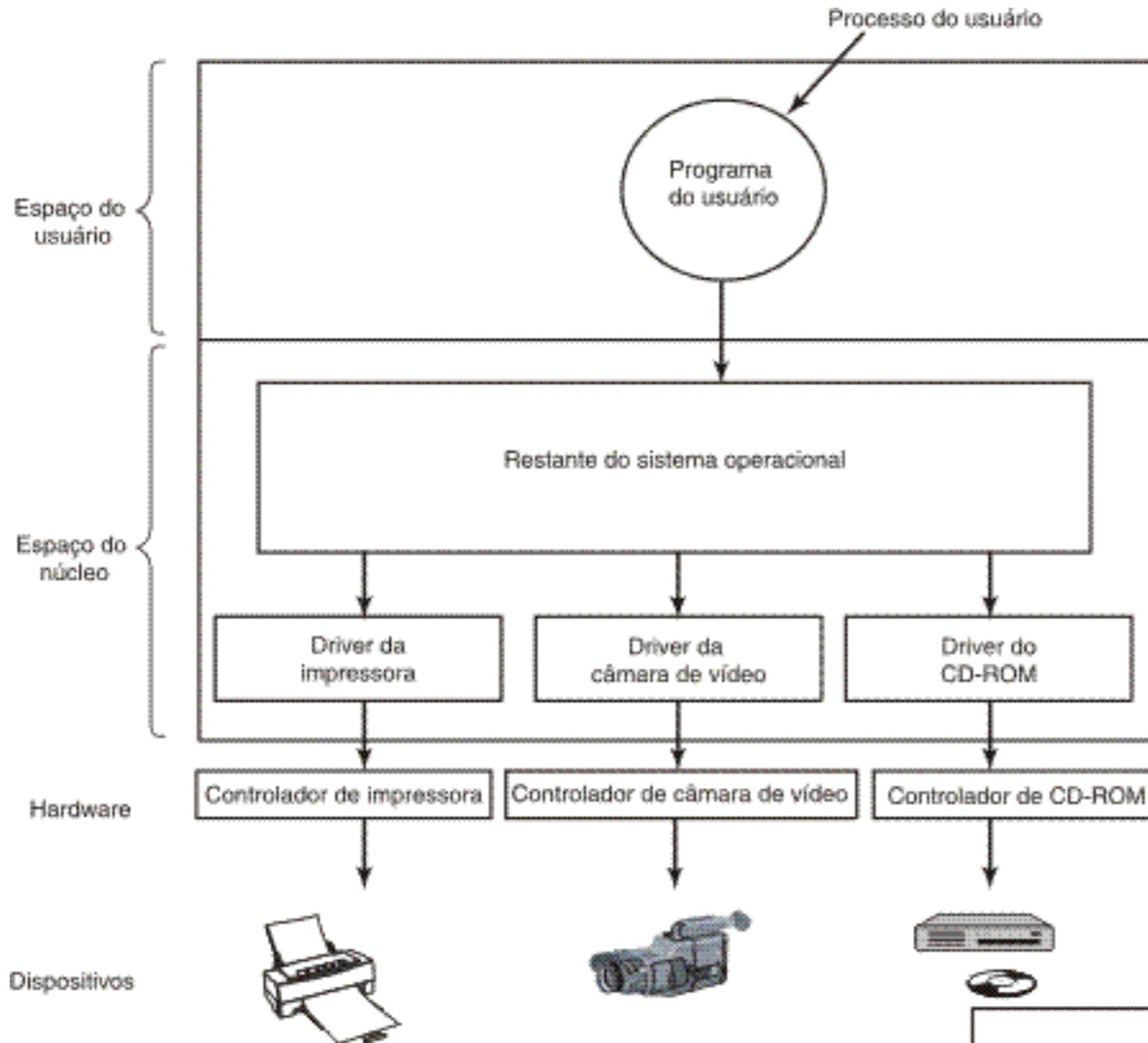
- As interrupções devem ser escondidas (transparentes) o máximo possível

Para tanto:

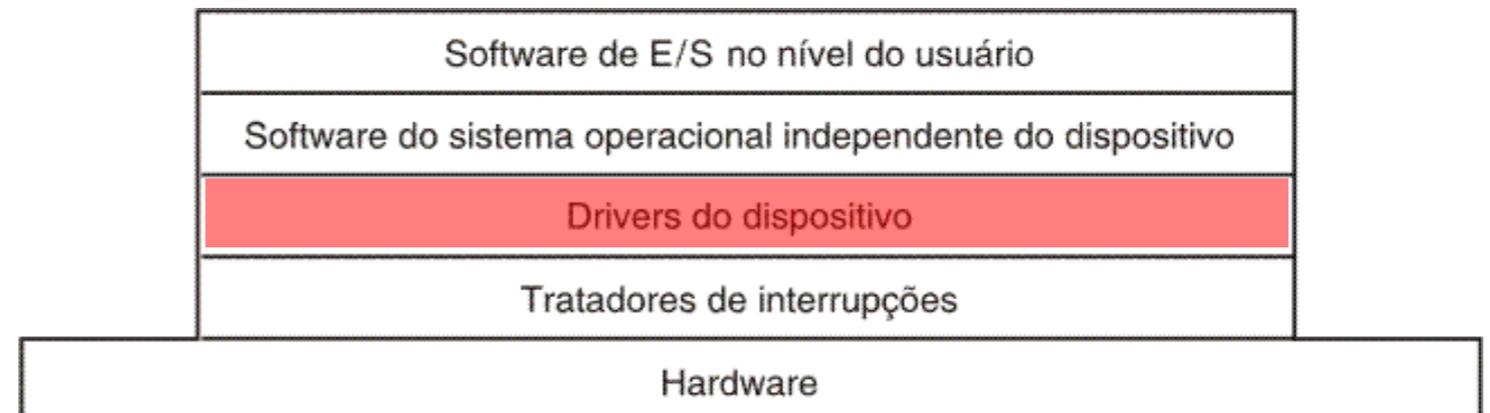
- bloqueia o *driver* que iniciou uma operação de E/S
- rotina de tratamento de interrupção cumpre sua tarefa
- notifica que a E/S foi completada
- e então desbloqueia o *driver* que a chamou



Drivers dos Dispositivos



- Posição lógica dos drivers dos dispositivos
- A comunicação entre os drivers e os controladores de dispositivos é feita por meio de barramento



Software de E/S Independente de Dispositivo

Funções do software de E/S independente de dispositivo

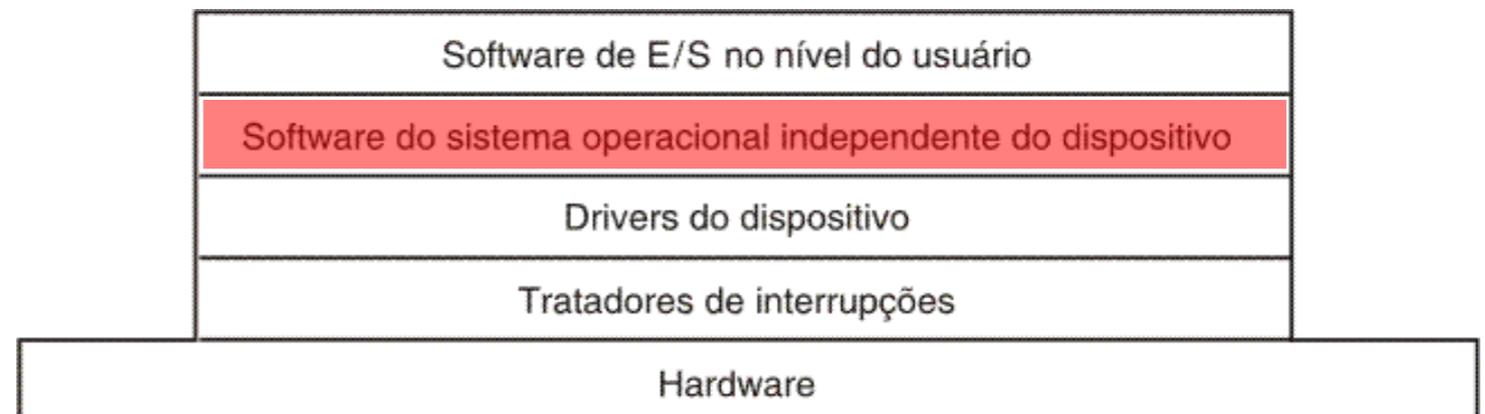
Interface uniforme para os drivers dos dispositivos

Armazenamento em buffer

Relatório de erros

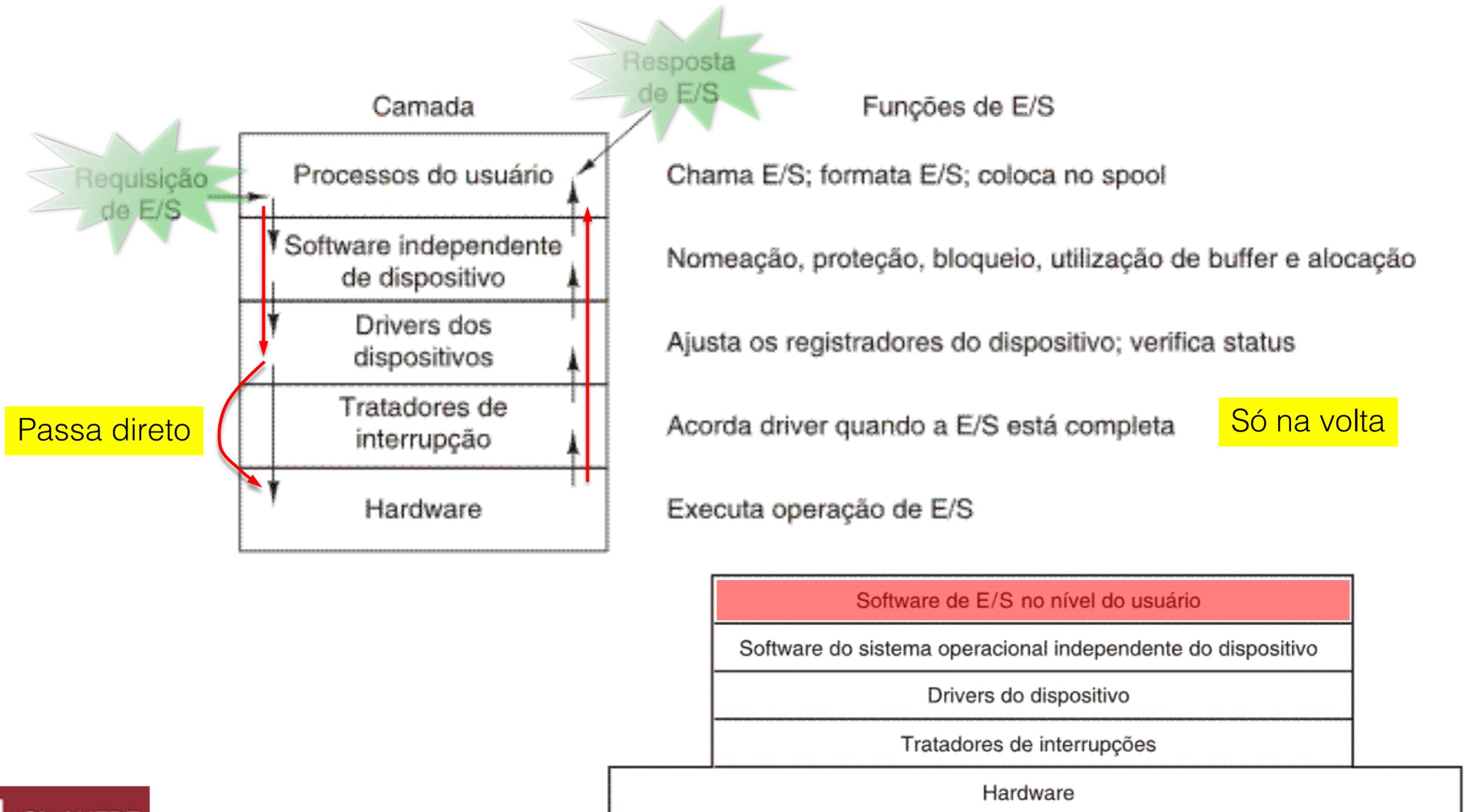
Alocação e liberação de dispositivos dedicados

Fornecimento de tamanho de bloco independente de dispositivo



Software de E/S do Espaço do Usuário

Camadas do sistema de E/S e as principais funções de cada camada



Entrada/Saída

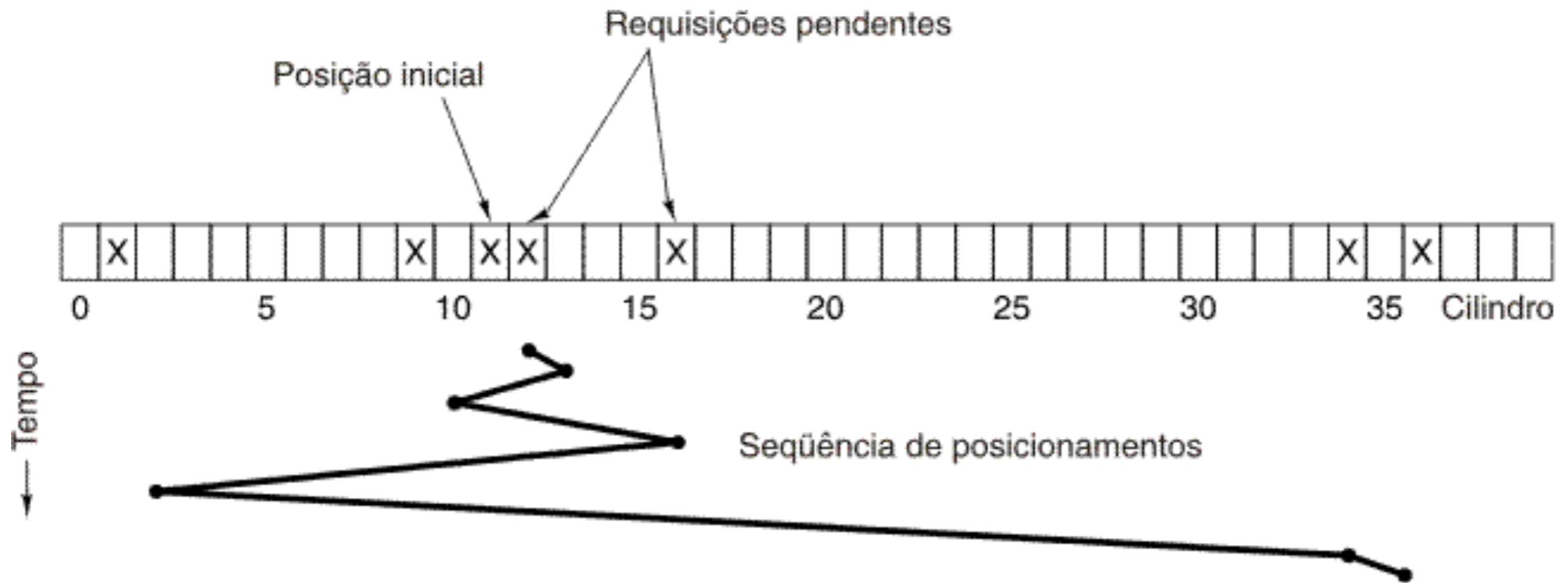
(Curiosidades)

- ❑ **E/S EM DISCO**
- ❑ **GERENCIAMENTO DE ENERGIA**

Algoritmos de Escalonamento de Braço de Disco (I)

- Tempo necessário para ler ou escrever um bloco de disco é determinado por 3 fatores
 1. tempo de posicionamento
 2. atraso de rotação
 3. tempo de transferência do dado
- **Tempo de posicionamento** domina
- Checagem de erro é feita por controladores

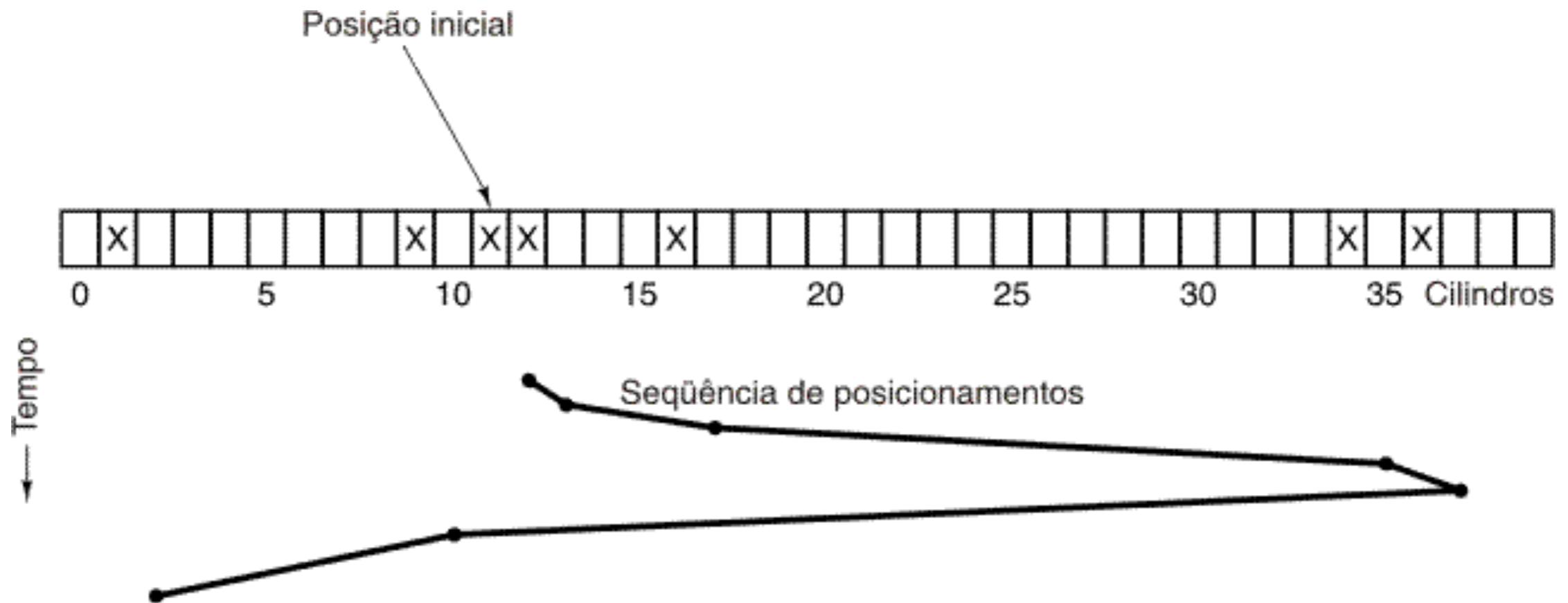
Algoritmos de Escalonamento de Braço de Disco (2)



Algoritmo de escalonamento de disco

Posicionamento Mais Curto Primeiro (SSF – *Shortest Seek First*)

Algoritmos de Escalonamento de Braço de Disco (3)



O algoritmo do **elevador** para o escalonamento das requisições do disco

Entrada/Saída

(Curiosidades)

- ✓ **E/S EM DISCO**
- **GERENCIAMENTO DE ENERGIA**

Gerenciamento de Energia (I)

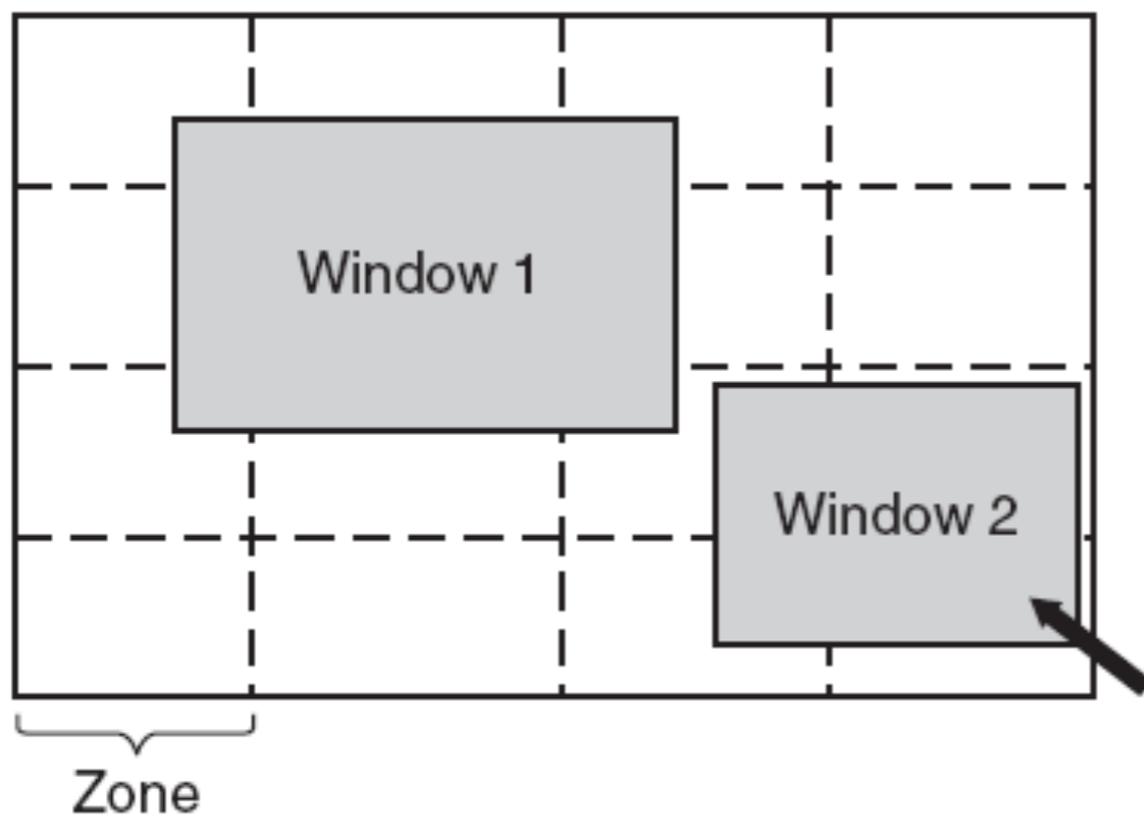
Dispositivo	Li et al. (1994)	Lorch e Smith (1998)
Monitor de vídeo	68%	39%
CPU	12%	18%
Disco rígido	20%	12%
Modem		6%
Som		2%
Memória	0,5%	1%
Outros		22%

Tecnologia mais avançada

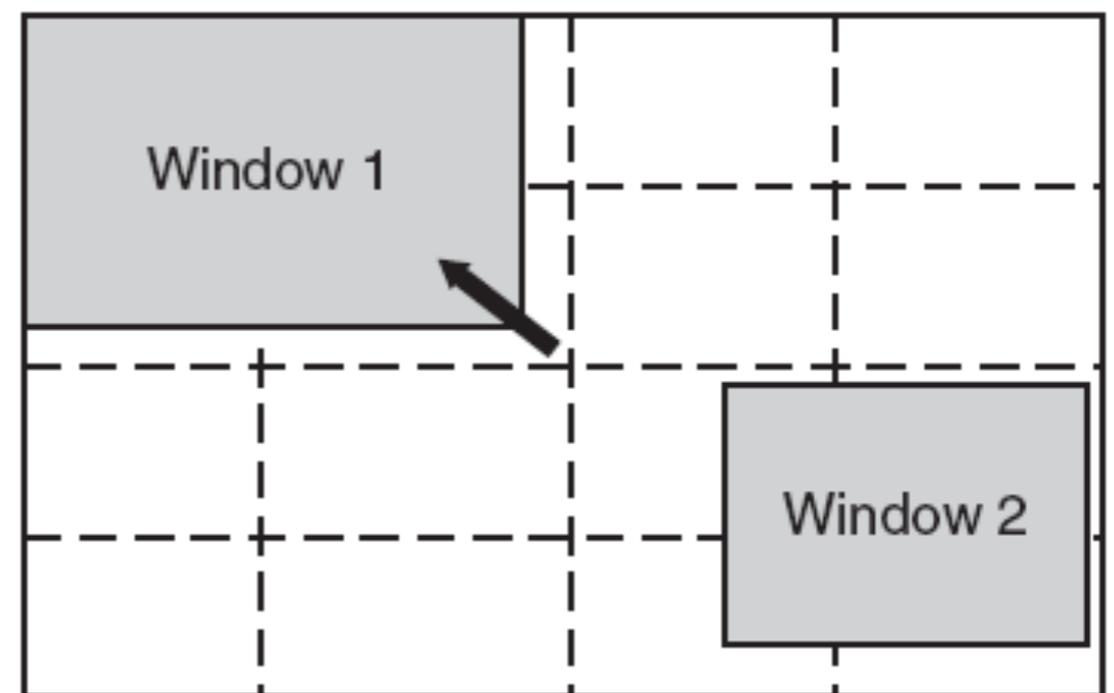
Aumento do desempenho

Consumo de energia de várias partes de um laptop

Gerenciamento de Energia (2): O uso de zonas para iluminação do monitor de vídeo



(a)



(b)

(a) Janela 2 selecionada e não se move.

(b) Janela 1 selecionada e se move para reduzir o número de zonas iluminadas.

Gerenciamento de Energia (3): Impactos na Entrada/Saída

- Dizer aos programas para usar menos energia
 - pode significar experiências mais pobres para o usuário
- Exemplos
 - muda de saída colorida para preto e branco
 - reconhecimento de fala com vocabulário reduzido
 - menos resolução ou detalhe em uma imagem

Conclusões: princípios básicos do software de E/S

- Subsistema de E/S é complexo dada a **diversidade** de periféricos
- Padronizar ao máximo para reduzir número de rotinas
- Novos dispositivos não alteram a **visão do usuário** em relação ao SO
- Organizado em **camadas**

Sistemas Distribuídos

A SEGUIR...