

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

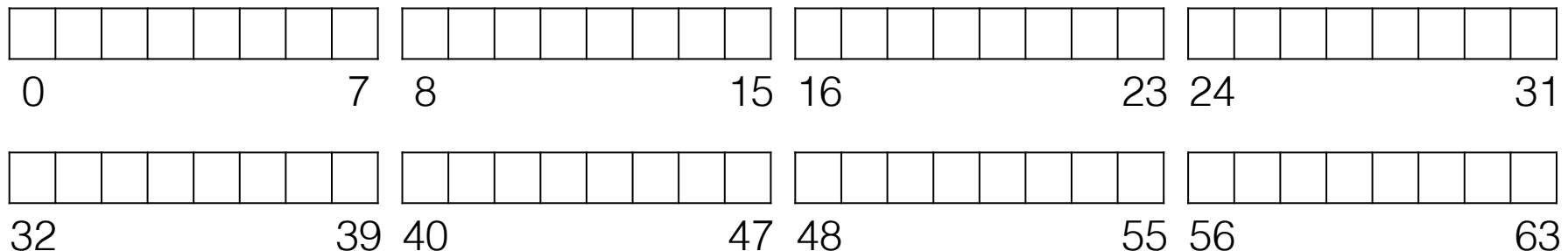
- Que estruturas em disco armazenam dados e metadados do Sistema de Arquivo?
- O que acontece quando um processo abre um arquivo?

- 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”

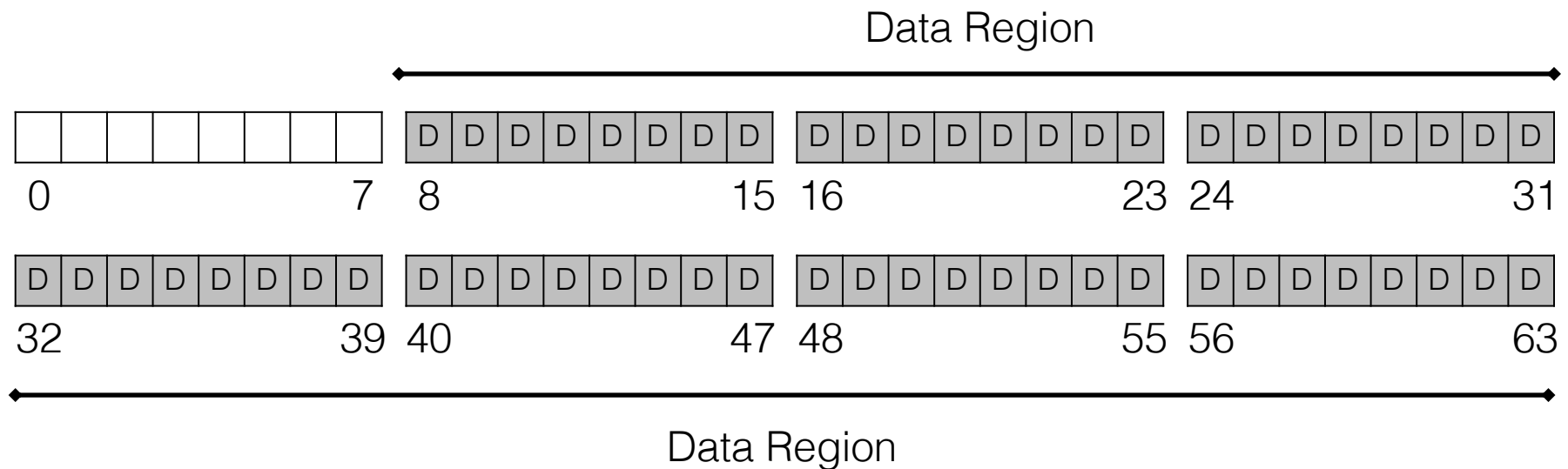
Organização Geral

- 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_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#
```

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

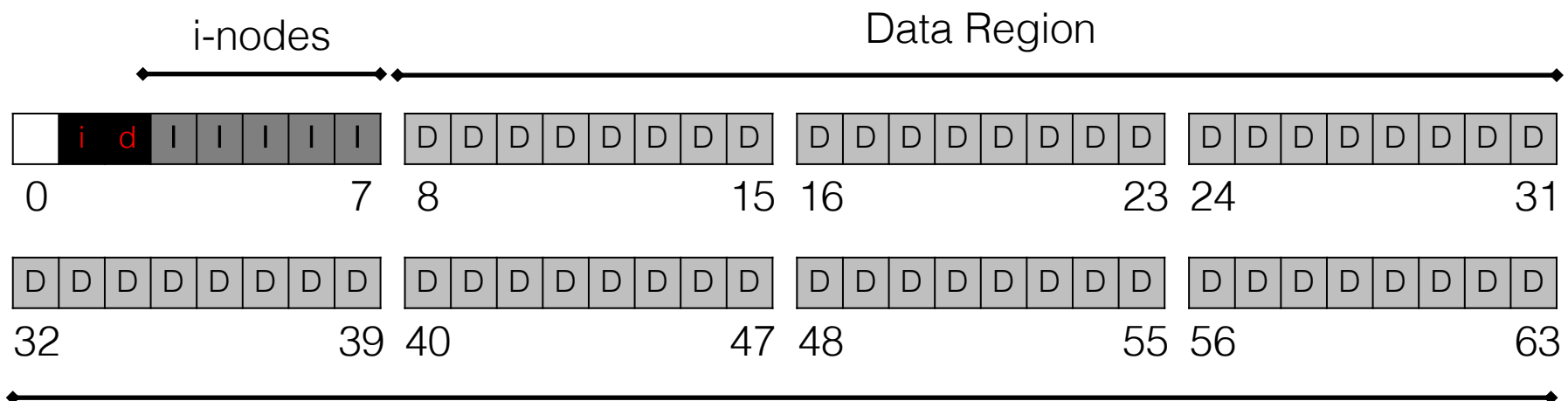
```
exercicios — sh — 82x24
[sh-3.2# ls -l
total 232
-rw-r--r--@ 1 cagf  staff  10939 May 15 14:11 Infra de Software - CC - 2018.1 - No
tas da lista 1.xlsx
-rw-r--r--@ 1 cagf  staff  47894 Apr 13 14:20 ListaAtividadesInfraSW.pdf
-rwxr-xr-x  1 root  staff   8564 Apr  2 08:52 pratica
-rw-r--r--  1 root  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#
```

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#
```

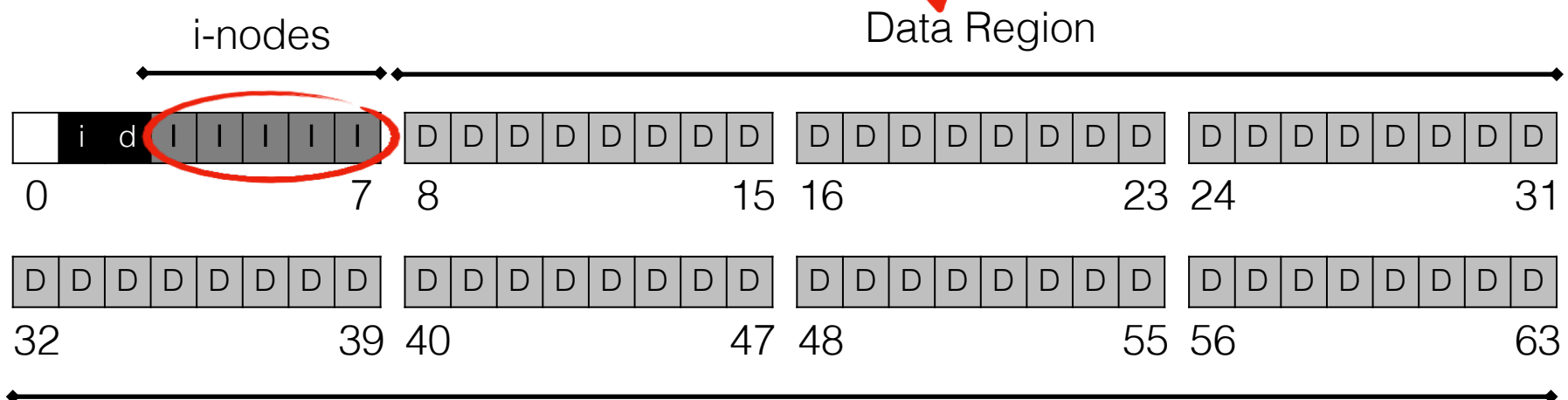
Estrutura de alocação

- 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**)



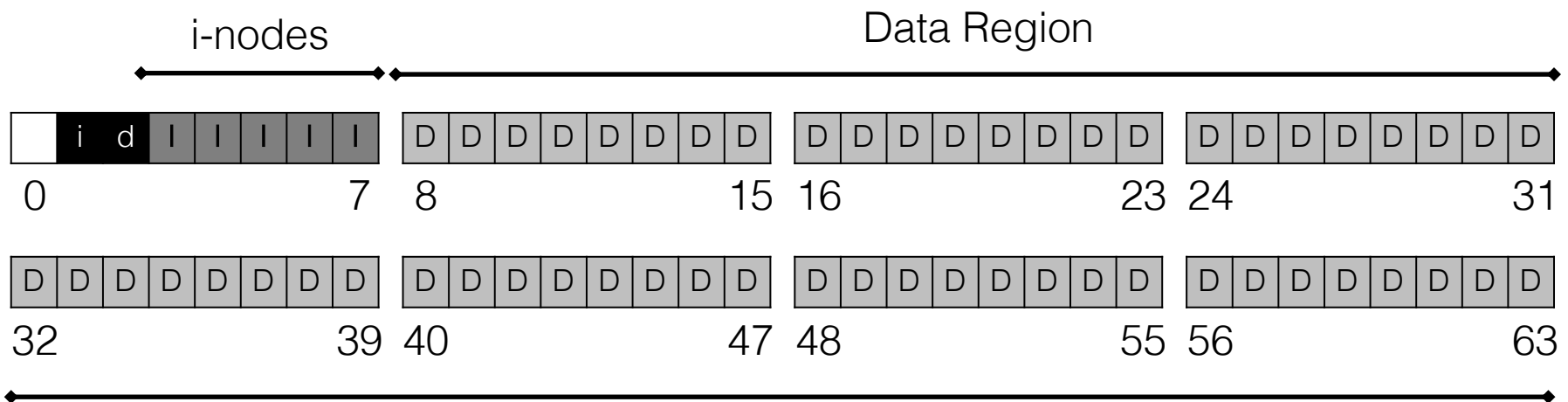
Observações

- 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**



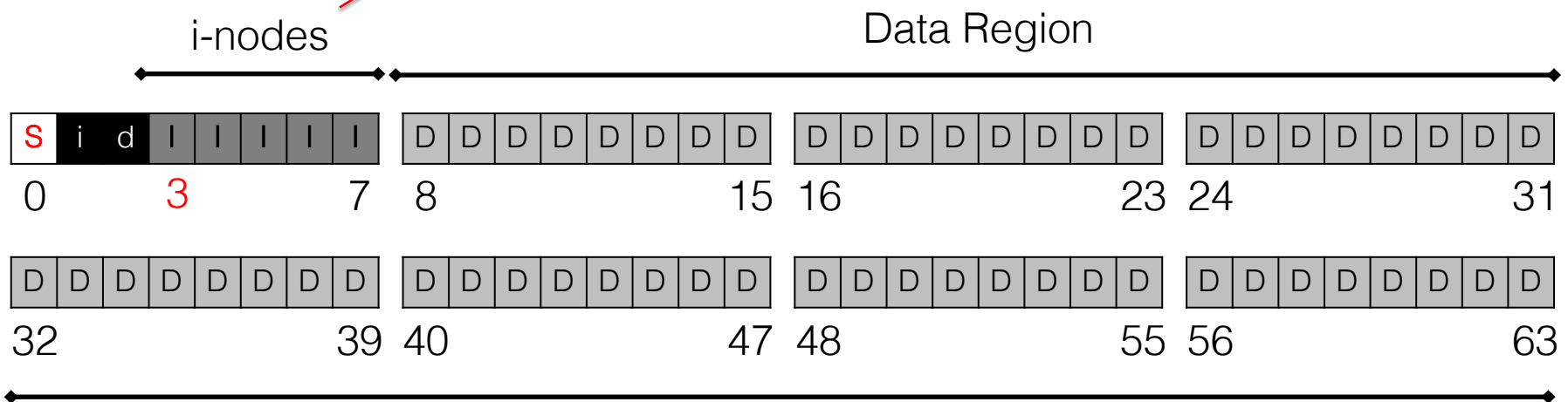
Observações (2)

- 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