Estudo de Sistemas de Arquivos

Cronograma:

- EXT e EXT2.
- Melhorias do EXT3.
- Visão Geral do EXT4.
- Testes Práticos com o EXT2.
- Novos Paradigmas: LISFS - Logical Information System as a File System

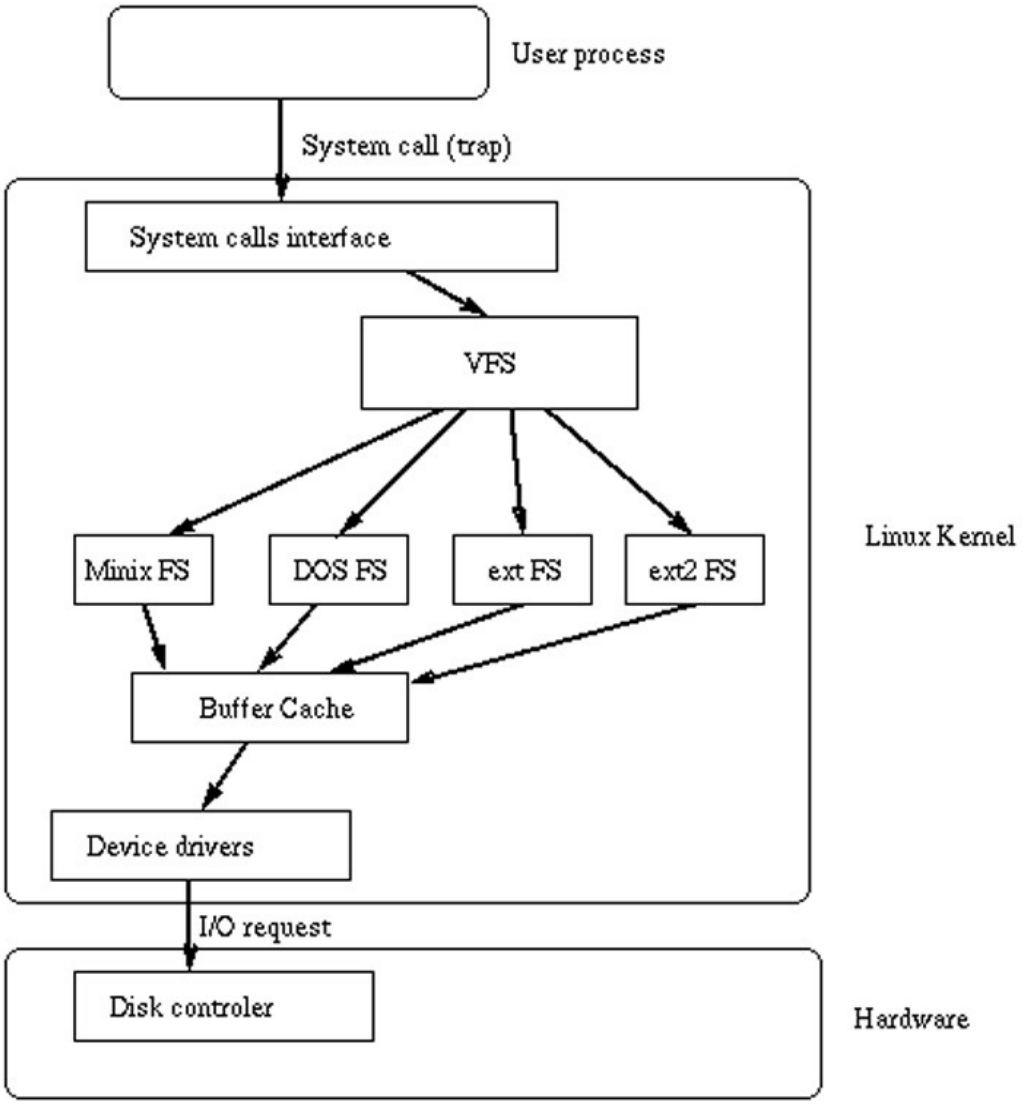
Conceitos

- INode: Estrutura de representação de um arquivo contendo sua descrição e o apontador para o bloco no disco.
- Diretório: Estruturas organizadas em árvore. Cada diretório possui uma lista de entradas contendo números de INodes e nome de arquivos. Um diretório é tratado como um arquivo especial.
- Links: Uma entrada no diretório que aponta para um INode. Pode ser Hard (com incremento de contador) ou Soft.

Conceitos

- Virtual File System (VFS): Um VFS define um conjunto de funções que todo sistema de arquivos tem para implementar.
- Estas funções compreendem operações associadas à três tipos de objetos: filesystem, inodes e open files.
- Esta camada é usada durante as chamadas de sistema ao atuarem em arquivos.

Conceitos


User process
System call (trap)
System calls interface
VFS
Minix FS
DOS FS
ext FS
ext2 FS
Linux Kernel
Buffer Cache
Device drivers
I/O request
Disk controler
Hardware

Linux Filesystems

- Como o Linux foi desenvolvido sobre o Minix, era natural criar um sistema de arquivos compatível entre os dois.
- Da integração do MFS com o VFS originou-se o EXT - Extended File System - em Abril de 1992.
- Essa implementação removia dois dos principais problemas do Minix:
  - Aumentava o tamanho máximo do sistema de arquivo de 64 MB para 2 GB; e
  - Aumentava o tamanho do nome do arquivo de 30 para 255 caracteres.
- Problemas: A lista ligada de INodes e Free Blocks ficava desordenada e fragmentava o sistema.

Second Extended File System - EXT2

- O EXT2 veio para sanar os problemas de fragmentação e desempenho do EXT.
- Duas da suas melhorias foram:
  - O aumento para 4 TB o tamanho da partição e o uso de 3 timestamps nos arquivos (criação, modificação e accesso).

Características - EXT2

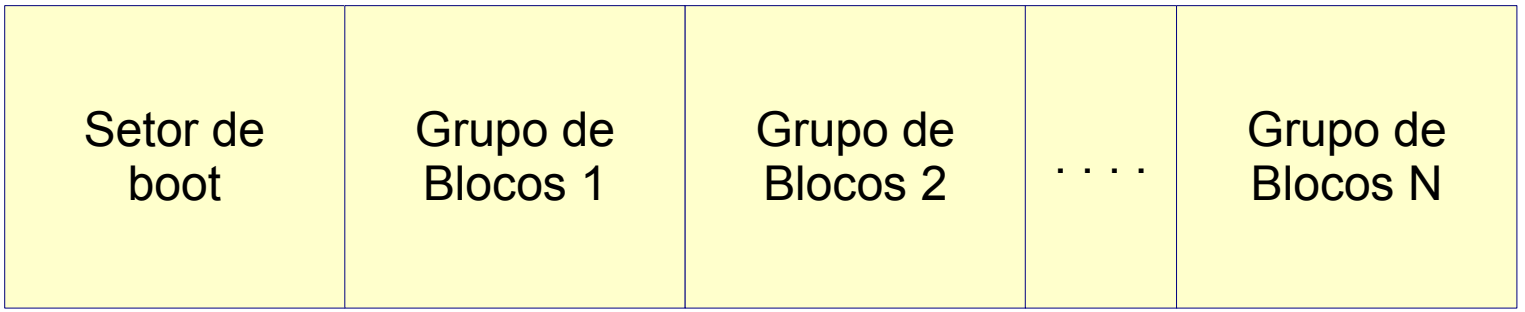
- Suporte aos tipos de arquivos UNIX padrão: arquivos, diretórios, arquivos especiais de dispositivos e links simbólicos.
- 4 TB para o sistema de arquivos.
- Pode-se estender o tamanho do nome de arquivos para no máximo 1012 caracteres.
- Reserva de 5% dos blocos para o ROOT.
- Pode-se escolher o tamanho do bloco lógico na criação do sistema de arquivos.
- Bloco que indica o status do sistema de arquivo: "NOT CLEAN", "CLEAN" ou "ERRONEOUS".
- Contador de mount/unmount para força verificação de integridade (presente no Ubuntu mesmo sendo EXT3).

Otimizações de Desempenho

- Quando um bloco é lido vários blocos contíguos também são lidos.
- O Kernel sempre tenta alocar o bloco de dados do arquivo no mesmo grupo que seu Inode.
- Na escrita de dados, até 8 blocos adjacentes são pré-alocados quando um novo bloco é alocado, permitindo que blocos contíguos sejam alocados e facilite uma futura leitura.

Estrutura Física - EXT2

- A estrutura foi fortemente influenciada pelo layout do BSD filesystem, onde o sistema de arquivos está disposto em grupos de blocos.



Estrutura Física - EXT2

- Cada grupo contém as seguintes informações:

Superbloco	Descritor de Grupo	Mapa de bits do bloco	Mapa de bits do inode	inodes	Blocos de Dados

Estrutura Física - EXT2

- Superbloco: informa quantos blocos e inodes existem e qual o tamanho de cada bloco.
- Descritor de Grupo: informa a localização do mapa de bits, do número de blocos e inodes livres no grupo e o número de diretórios no grupo.
- Mapas de bits: Controlam os blocos e inodes livres respectivamente.

Estrutura Física - EXT2

- A estrutura de cada entrada do diretório é formada por:

Número do INode	Tamanho da entrada	Tamanho do nome do arquivo	Nome do Arquivo

- A estrutura de cada INode é formada por:

Tipo do Arquivo	Permissões de acesso	Criador	Timestamps	Tamanho	Apontadores para o bloco de dados

Do EXT2 ao EXT3

- O EXT3 é fortemente baseado no EXT2, o que significa que um sistema EXT2 pode ser desmontado e remontado como EXT3 e vice-versa, tendo inclusive compatibilidade de metadados.
- Mas qual é a diferença entre os dois sistemas?
  - EXT3 ganhou uma poderosa ferramenta de fsck.
  - EXT3 tornou-se um journaling filesystem.

Journaling filesystem

- Um journaling filesystem significa que determinados eventos são “noticiados”.
- Com o conhecimento dos eventos pode-se recuperar o sistema de falhas.
- Dessa forma tende-se a evitar o uso do fsck.

Metadata-only journaling

- Outros sistemas de arquivos implementam journaling, tal como , ReiserFS, XFS e JFS, mas em todos eles somente o metadado é gravado.
- Se estiver gravando um arquivo quando o sistema reiniciar inesperadamente, você terá os metadados facilmente recuperados, mas os dados de sua atualização serão perdidos.

EXT3

- No EXT3 tanto os dados como os metadados são “noticiados”.
- A integridade dos dados pode ser feita de três modos (em ordem de velocidade):
  - “data=writeback” : Rápido, evita fsck, mas recupera dados antigos após um crash.
  - “data=ordered” : (default mode) Grava as modificações dos metadados e grava os blocos modificados.
  - “data=journal” : Todas as modificações no sistema de arquivos são gravadas possibilitando uma recuperação total, mas tornando o sistema muito lento.

EXT4 – Visão Geral

- EXT4 é um refinamento do EXT2 usando duas partições simultaneamente (em discos diferentes).
- Uma partição armazena os diretórios e Inodes e a outra os arquivos.
- A idéia do sistema é realizar leitura/gravação simultaneamente de diretórios e arquivos.

LISFS

Logical Information System as a File System

Logical Information System as a File System

- Artigo LISFS de Yoann, Benjamin e Olivier (2006).
- Criação de um Framework para uma busca lógica.

Operações Básicas

- Existem 3 operações básicas:
  - / : and.
  - | : or.
  - ! : not.
- Exemplo:

  /home/ec2003/ra027106

  Está em home E em ec2003 E em ra027106.

Modo de Operação

- Trabalha analogamente ao diálogo entre o cliente e o vendedor:

C: Eu quero comprar flores. Quais que você tem?
V: Você tem alguma idéia da cor, do tipo ou do tamanho do buquê?
C: Eu quero um buquê bem grande! Quais as cores que você tem?
V: Vermelho, branco ou amarelo.
.....

Exemplo: Banco de Dados de arquivos MP3

```
cd /music/year:[1982..1990]
cd !genre:Samba
cd time:<7min
cd .mp3
playmp3 *
......
cd /music/genre:Disco/
ls
   artist:BeeGees/
   [...]
   year:1976/  year:1977/
   [...]
```

Exemplo - Arquivos fontes

```
cd contains:sincronizacao
ls
   author:Tiago/  author:Henrique/
   [...]
   year:2007/  year:2006/
   [...]
   numWords:1578/  numwords:67569/
   [...]
   filetype:odt/  filetype:h/
cd .c
cd function:addNode
ls
   addPeer.c
emacs addPeer.c
```

Testes: Preparação

Criando uma imagem zerada:

```
# dd if=/dev/zero of=./hd.dmp bs=1k count=10000
```

Testes: Preparação

Criando uma imagem zerada:

```
# dd if=/dev/zero of=./hd.dmp bs=1k count=10000
```

Visualizando a imagem:

```
# hexedit hd.dmp
```

Testes: Preparação

Criando uma imagem zerada:

```
# dd if=/dev/zero of=./hd.dmp bs=1k count=10000
```

Visualizando a imagem:

```
# hexedit hd.dmp
```

Associando a um device:

```
# losetup /dev/loop0 hd.dmp
```

Testes: Preparação

Criando uma imagem zerada:

```
# dd if=/dev/zero of=./hd.dmp bs=1k count=10000
```

Visualizando a imagem:

```
# hexedit hd.dmp
```

Associando a um device:

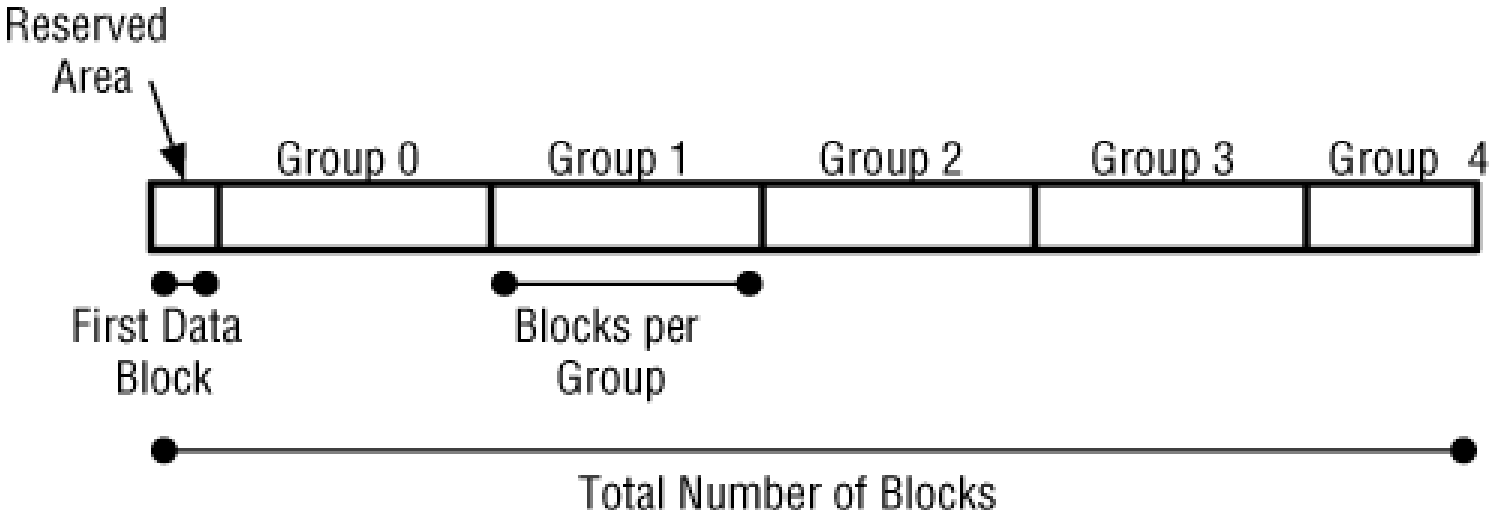
```
# losetup /dev/loop0 hd.dmp
```

Formatando com Ext2:

```
# mke2fs /dev/loop0
```

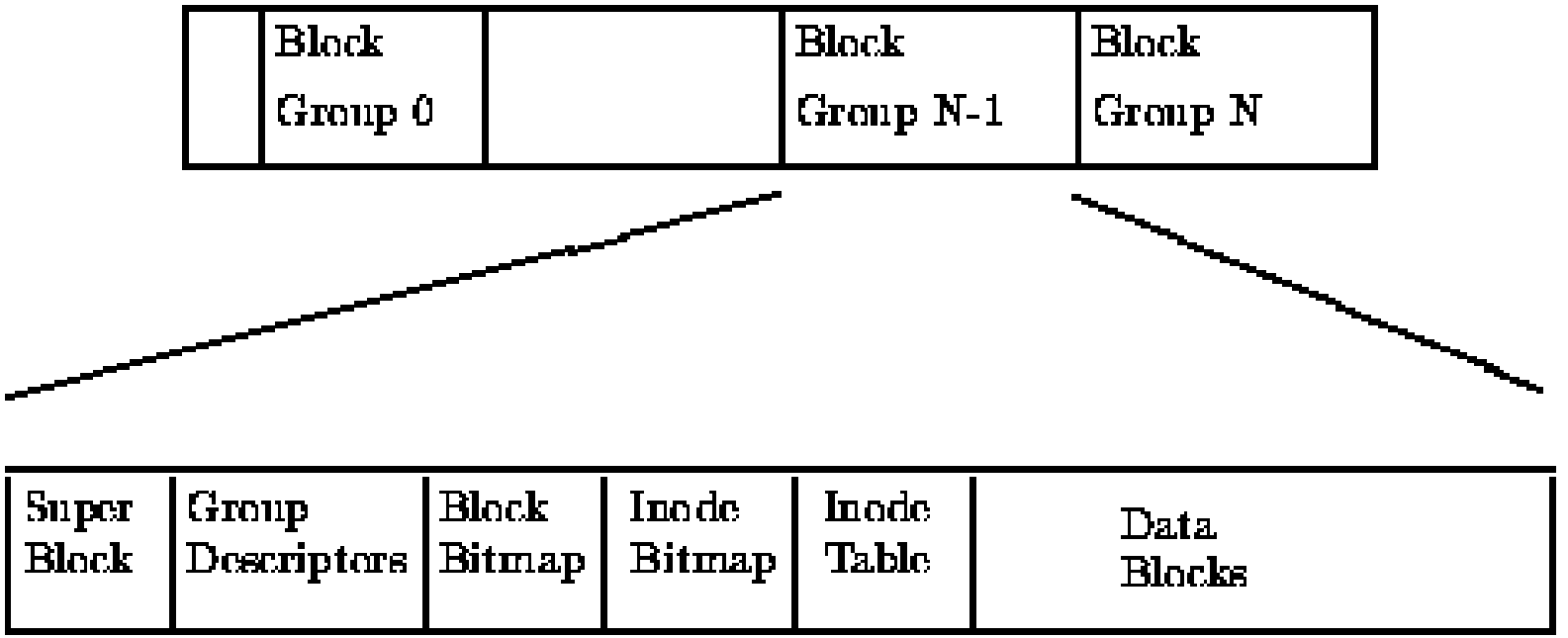
Testes: Analisando a imagem

Layout do Ext



Testes: Analisando a imagem

Superblocos e os Grupos de Blocos



Testes: Analisando a imagem

Layout de Superblock do Ext

Byte Range	Description	Essential
0–3	Number of inodes in file system	Yes
4–7	Number of blocks in file system	Yes
8–11	Number of blocks reserved to prevent file system from filling up	No
12–15	Number of unallocated blocks	No
16–19	Number of unallocated inodes	No
20–23	Block where block group 0 starts	Yes
	• • •	
104–119	File system ID	No
120–135	Volume name	No

Testes: Alterando o Superbloco

Setando o nome (label) de um filesystem:

```
# tune2fs -L 'Ext2 Dump' /dev/loop0
```

Testes: Alterando o Superbloco

Setando o nome (label) de um filesystem:

```
# tune2fs -L 'Ext2 Dump' /dev/loop0
```

Visualizando o resultado:

```
# debugfs /dev/loop0
debugfs:  show_super_stats
Filesystem volume name:   Ext2 Dump
```

Testes: Alterando o Superbloco

Setando o nome (label) de um filesystem:

```
# tune2fs -L 'Ext2 Dump' /dev/loop0
```

Visualizando o resultado:

```
# debugfs /dev/loop0
debugfs:  show_super_stats
Filesystem volume name:   Ext2 Dump
```

Alterando diretamente:

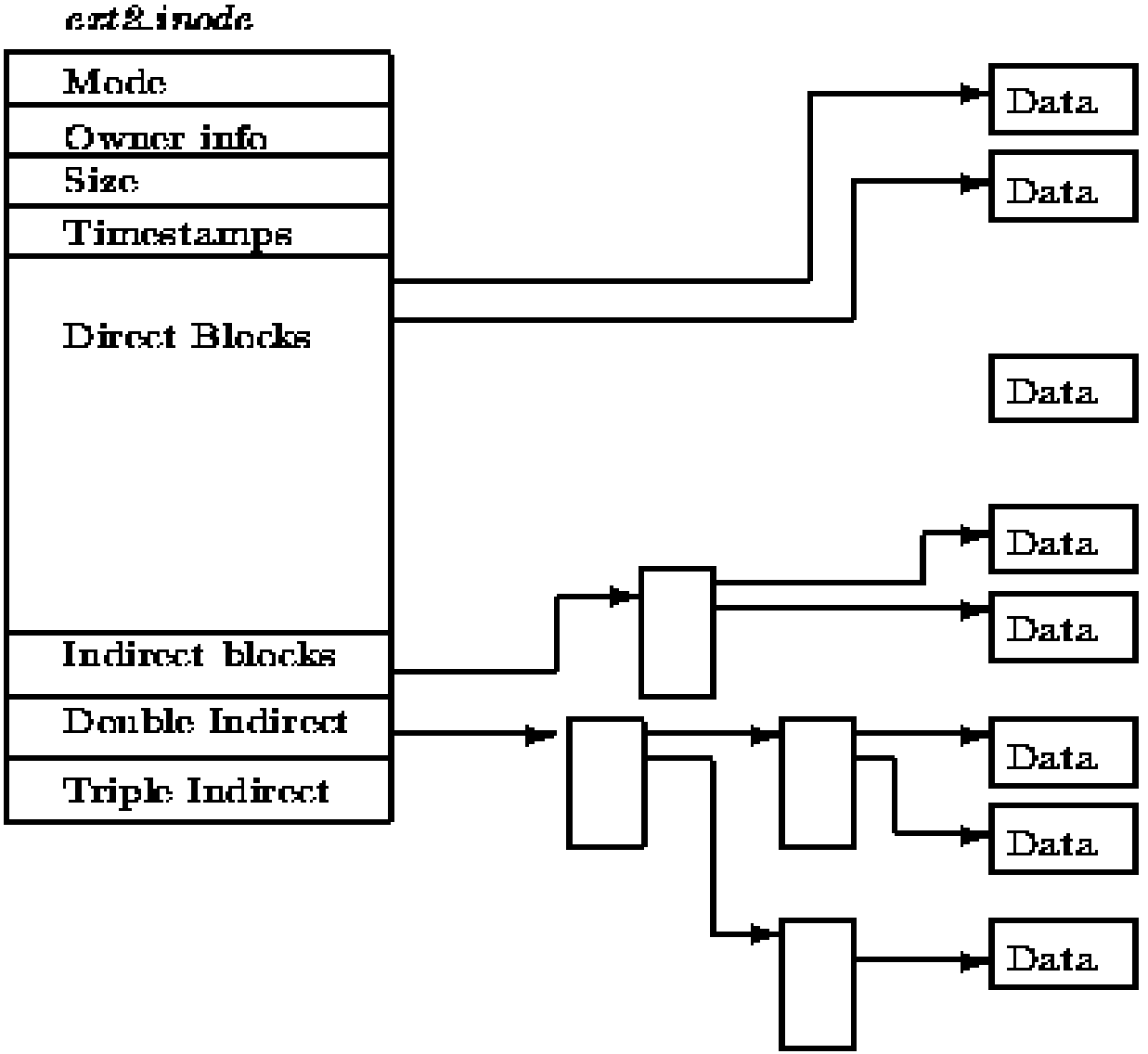
```
# hexedit hd.dmp
```

Analisando Inodes

Criando dois arquivos na imagem:

```
# mount /dev/loop0 /mnt/hd_dmp
# cat > /mnt/hd_dmp/a.txt
a
# cat > /mnt/hd_dmp/b.txt
b
# umount /mnt/hd_dmp
```

Analisando Inodes



Modificando Inodes

Trocando os apontadores de dois arquivos:

```
# debugfs -w /dev/loop0
debugfs:
```

Modificando Inodes

Trocando os apontadores de dois arquivos:

```
# debugfs -w /dev/loop0
debugfs:  stat a.txt
...
BLOCKS:
(0):1537
...
```

Modificando Inodes

Trocando os apontadores de dois arquivos:

```
# debugfs -w /dev/loop0
debugfs:  stat a.txt
...
BLOCKS:
(0):1537
...

debugfs:  stat b.txt
...
BLOCKS:
(0):2049
...
```

Modificando Inodes

Trocando os apontadores de dois arquivos:

debugfs: mi a.txt

...

Direct Block #0 [1537] 2049

Direct Block #1 [0]

...

Modificando Inodes

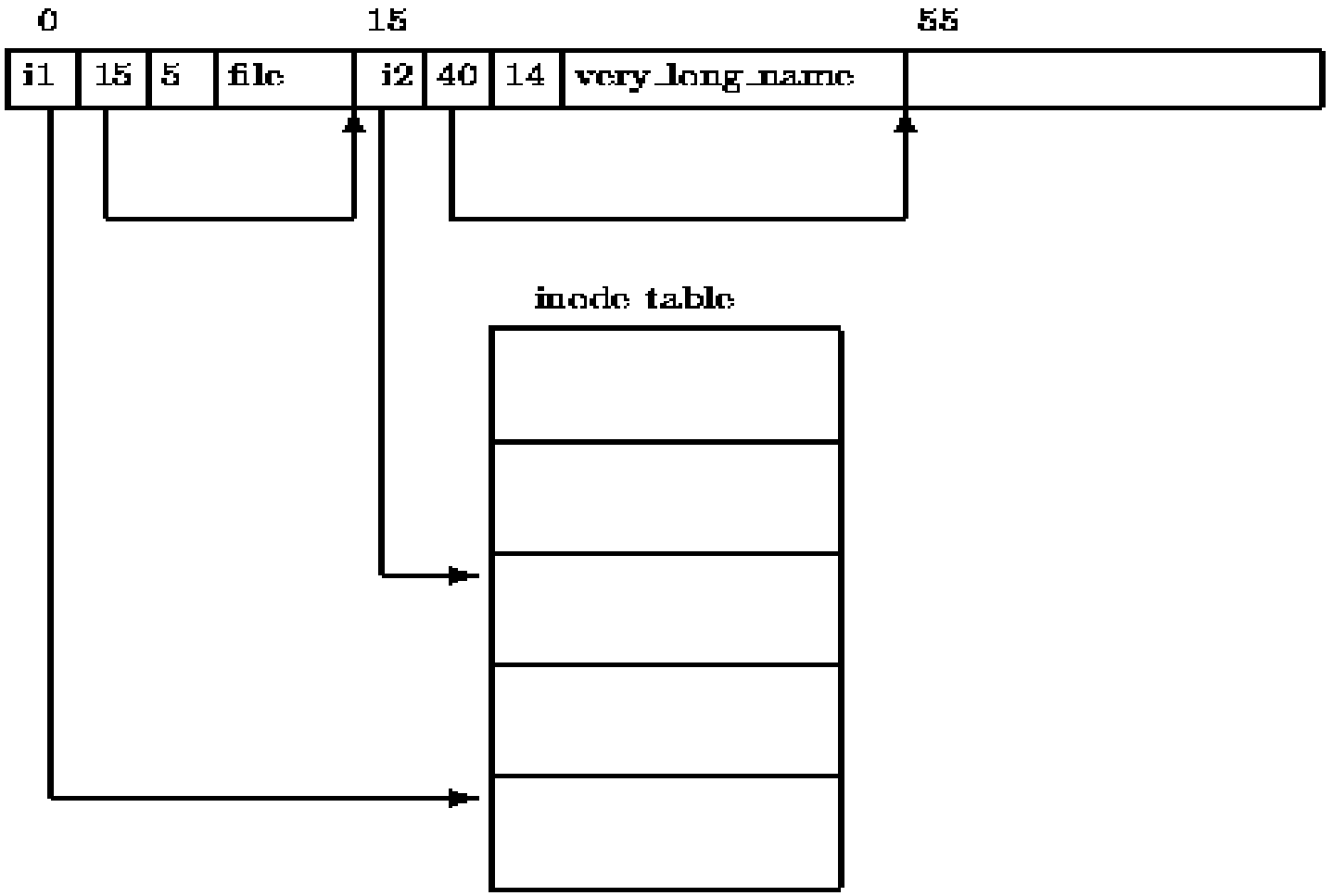
Trocando os apontadores de dois arquivos:

```
debugfs:  mi a.txt
...
        Direct Block #0    [1537] 2049
        Direct Block #1    [0]
...

debugfs:  mi b.txt
...
        Direct Block #0    [2049] 1537
        Direct Block #1    [0]
...
```

Analisando Diretórios

Layout de diretorio no EXT2



Modificando um diretório

Criando um link para um arquivo:

debugfs: mkdir dir1

debugfs: cd dir1

debugfs: ln <12> a_link.txt

debugfs: cat a_link.txt

Modificando um diretório

Criando um link para um arquivo:

debugfs: mkdir dir1
debugfs: cd dir1
debugfs: ln <12> a_link.txt
debugfs: cat a_link.txt

Modificando a estrutura de diretório:

hexedit hd.dmp

Procurar por a_link.txt e substituir por A_link.txt

mount /dev/loop0 /mnt/hd_dmp
ls /mnt/hd_dmp/dir1

Recuperando arquivos deletados

Recuperando um arquivo deletado:

```
# mount /dev/loop0 /mnt/hd_dmp
# cat > c.txt
c
# rm c.txt
# umount /mnt/hd_dmp
```

Recuperando arquivos deletados

Recuperando um arquivo deletado:

```
# mount /dev/loop0 /mnt/hd_dmp
# cat > c.txt
c
# rm c.txt
# umount /mnt/hd_dmp

# debugfs -w /dev/loop0
debugfs: lsdel
debugfs: undelete <14> c.txt
debugfs: cat c.txt
c
```

Bibliografia

- Card, R., Ts'o, T., Tweedie, S.; “Design and Implementation of the Second Extended Filesystem”, in Proceedings of the First Dutch International Symposium on Linux, ISBN 90-367-0385-9.
- Padioleau, Y., Sigonneau, B., Ridoux, O.;“LISFS: a logical information system as a file system”, International Conference on Software Engineering archive Proceeding of the 28th international conference on Software engineering table of contents, Shanghai, China, 2006.
- Budiu, M.; “A Dual-disk File System: ext4”, disponível em: http://www.cs.cmu.edu/~mihaib/fs/fs.html. Último acesso: 13 de Novembro de 2007.
- Robbins, D.; “Common threads: Advanced filesystem implementor's guide, Part 7”, disponível em: http://www-128.ibm.com/developerworks/linux/library/l-fs7.html. Último acesso: 13 de Novembro de 2007.

Bibliografia

- Rémy Card,Theodore Ts'o,Stephen Tweedie ; “Design and Implementation of the Second Extended Filesystem”, disponível em: http://web.mit.edu/tytso/www/linux/ext2intro.html, último acesso: 13/11/2007
- David A Rusling; “The Linux Kernel”, disponível em: http://www.science.unitn.it/~fiorella/guidelinux/tlk/tlk-html.html, último acesso: 13/11/2007
- Ralf Hölzer; “Cryptoloop HOWTO, Chap 5. Setting up the loop device”, disponível em: http://tldp.org/HOWTO/Cryptoloop-HOWTO/loopdevice-setup.html, último acesso: 13/11/2007
- Brian Carrier, “File System Forensic Analysis”, Addison Wesley Professional, 17/03/2005