

SISTEMAS OPERACIONAIS 1

21270 A



Departamento de Computação
Prof. Kelen Cristiane Teixeira Vivaldini

Lab. Sistema de Arquivos

Aula 13

Lab. Sistema de Arquivos

Partições

São divisões existentes no disco rígido que marcam onde começa onde termina um sistema de arquivos. As partições nos permitem usar mais de um sistema operacional no mesmo computador (como o GNU/Linux, Windows e DOS), ou dividir o disco rígido em uma ou mais partes para ser usado por um único sistema operacional ou até mesmo por diferentes arquiteturas (32 e 64 bits).

Para gravar os dados, o disco rígido deve ser primeiro particionado (usando o cfdisk, parted, diskdruid, fdisk), escolher o tipo da partição (*Linux Native*, *Linux Swap*, etc) e depois aquela partição deve ser formatada com o `mkfs.ext3`.

Lab. Sistema de Arquivos

Partições

Lembre-se:

- Quando se apaga uma partição, você estará apagando TODOS os arquivos existentes nela!
- A partição do tipo Linux Native (Tipo 83) é a usada para armazenar arquivos no GNU/Linux, tanto ext2, ext3, ext4, reiserfs, xfs, etc.
- A partição do tipo Linux Swap (Tipo 82) é usada como memória virtual.
- Em sistemas novos, é comum encontrar o Windows instalado em uma partição que consome TODO o espaço do disco rígido. Uma solução para instalar o GNU/Linux é apagar a partição Windows e criar três com tamanhos menores (uma para o Windows, uma para o GNU/Linux e outra para a *Memória Virtual do Linux (SWAP)*). Ou criar apenas 2 se você não quiser mais saber mais do Windows .

Lab. Sistema de Arquivos

Sistema de Arquivos

- É criado durante a "formatação" da partição de disco . Após a formatação toda a estrutura para leitura/gravação/permisões de arquivos e diretórios pelo sistema operacional estará pronta para ser usada. Normalmente este passo é feito durante a instalação de sua distribuição GNU/Linux.
- Cada sistema de arquivos tem uma característica em particular mas seu propósito é o mesmo: Oferecer ao sistema operacional a estrutura necessária para ler/gravar os arquivos/diretórios.

Lab. Sistema de Arquivos

Sistema de arquivos ext2

- O sistema de arquivos ext2 (também conhecido como o segundo sistema de arquivos estendido) foi desenvolvido para abordar defeitos no sistema de arquivos Minix usado em versões anteriores do Linux. Ele foi usado intensivamente no Linux por muitos anos. Não há *journaling* no ext2, que foi substituído em grande parte pelo ext3. Usado em partições Linux Nativas para o armazenamento de arquivos (cód83). Seu tamanho deve ser o suficiente para acomodar todo os arquivos e programas que deseja instalar no GNU/Linux .

Lab. Sistema de Arquivos

Sistema de arquivos ext3

- O sistema de arquivos ext3 adiciona melhorias em relação ao ext2, como destaque o recurso de journaling e suporte a arquivos de até 16Gb. Ele também é identificado pelo tipo 83 e totalmente compatível com o ext2 em estrutura. O journal mantém um log de todas as operações no sistema de arquivos, caso aconteça uma queda de energia elétrica (ou interrompa o funcionamento do sistema), o fsck verifica o sistema de arquivos no ponto em que estava quando houve a interrupção, evitando a demora para checar todo um sistema de arquivos.

Sistema de arquivos ReiserFS

- ReiserFS é um sistema de arquivos baseado em árvore B que possui um bom desempenho geral, especialmente para um grande número de arquivos pequenos. Possui os mesmos recursos do ext3, mas seu design é diferente e seu tempo de recuperação em caso de queda de energia menor que o ext3.
- O ReiserFS também escala bem e possui journaling. Ele não está mais em desenvolvimento ativo e foi amplamente substituído pelo Reiser4.

Lab. Sistema de Arquivos

Sistema de arquivos Swap

- Swap - Usado em partições Linux Swap para oferecer memória virtual ao sistema. (menos que 16MB de memória RAM).

Sistema de arquivos do Kernel

- proc - Sistema de arquivos do kernel

Sistema de arquivos FAT

- FAT12 - Usado em disquetes no DOS. Não possui suporte a permissões, journaling.
- FAT16 - Usado no DOS e oferece suporte até discos de 2GB. Não possui suporte a permissões e journaling.
- FAT32 - Também usado no DOS e oferece suporte a discos de até 2 Terabytes. Não possui suporte a permissões e journaling. NTFS - Formato nativo de discos de sistemas operacionais Windows XP e superiores. Possui suporte a permissões de acesso e compactação nativa.

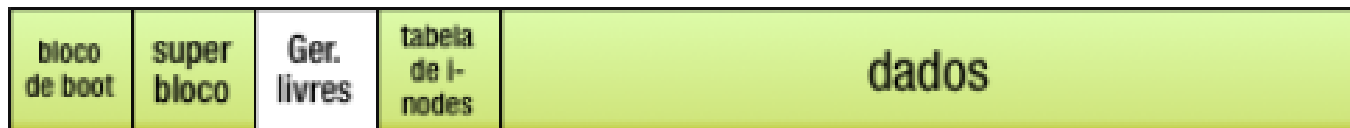
Lab. Sistema de Arquivos

Partição EXT2

- A partição EXT2 é o tipo usado para criar o sistema de arquivos Linux Native usado para armazenar o sistema de arquivos EXT2 (após a formatação) e permitir o armazenamento de dados. Note que também é possível criar um sistema de arquivos EXT2 em um arquivo (ao invés de uma partição) que poderá ser montado e acessado normalmente pelo sistema de arquivos

Lab. Sistema de Arquivos

Partição EXT2



O **super-bloco** contém parâmetros do FS tais como:

- Tamanho da tabela de *i-nodes*;
- Tamanho dos blocos lógicos de disco;
- Tamanho do sistema de arquivos em número de blocos lógicos.

A tabela de *i-nodes* contém um *i-node* para cada arquivo no sistema de arquivos. *i-nodes* são *identificados numericamente pela sua posição* na tabela de *i-nodes*. Para visualizar os valores de *i-nodes* para os arquivos utilize:

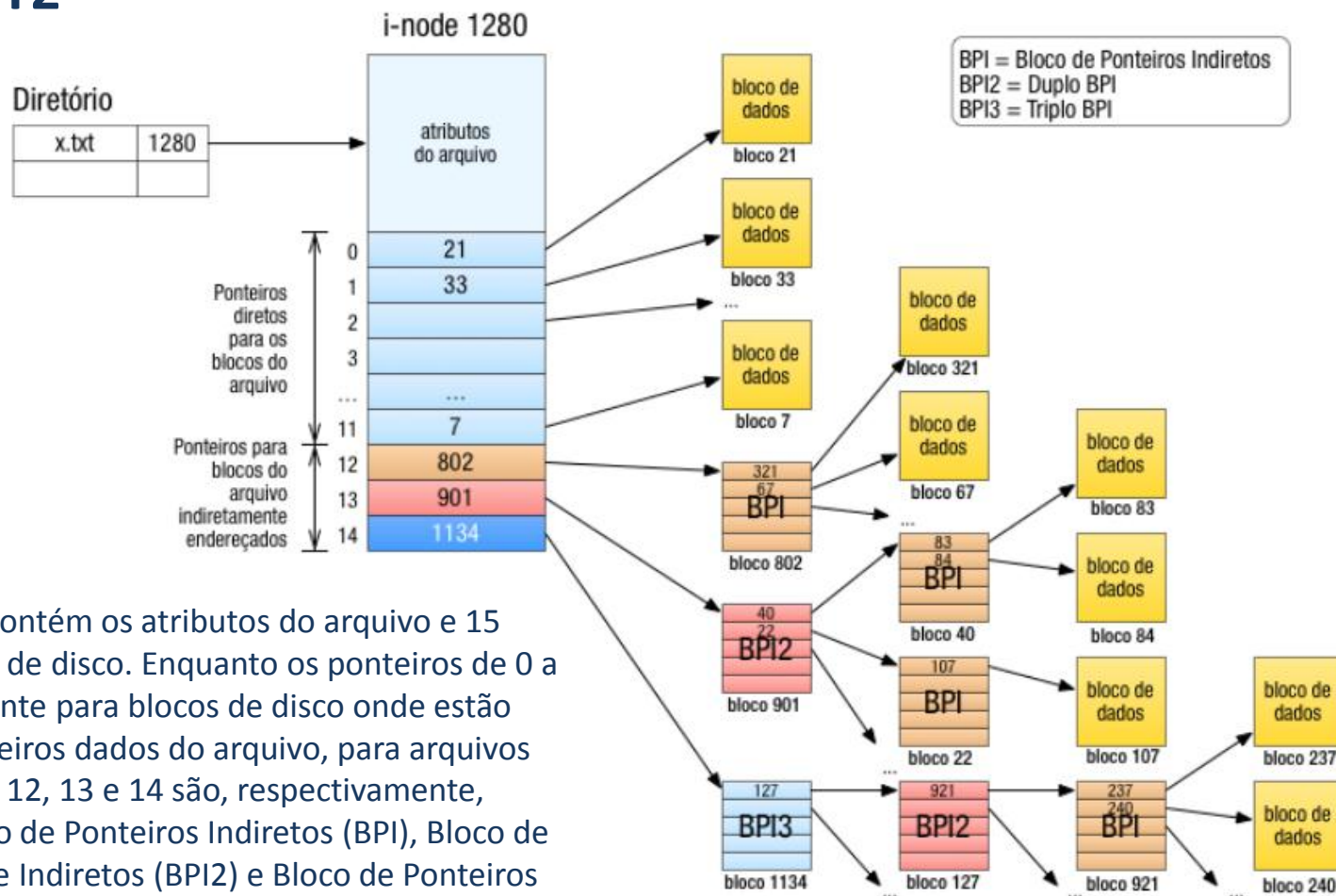
```
ls -li
```

Informações contidas nos *i-nodes*:

- Tipo do arquivo (regular, diretório, especial, etc.)
- Proprietário do arquivo (UID)
- Grupo do arquivo (GID)
- Permissões de acesso
- *Timestamps* (último acesso, última modificação)
- Tamanho do arquivo
- Número de blocos de 512 bytes alocados para o arquivo (pode ser diferente do tamanho do arquivo em arquivos esparsos)
- Ponteiros para os blocos de disco onde estão armazenados os dados do arquivo

Lab. Sistema de Arquivos

Partição EXT2



Cada i-node no ext2 contém os atributos do arquivo e 15 ponteiros para blocos de disco. Enquanto os ponteiros de 0 a 11 apontam diretamente para blocos de disco onde estão armazenados os primeiros dados do arquivo, para arquivos maiores, os ponteiros 12, 13 e 14 são, respectivamente, ponteiros para o Bloco de Ponteiros Indiretos (BPI), Bloco de Ponteiros Duplamente Indiretos (BPI2) e Bloco de Ponteiros Triplamente Indiretos (BPI3).

Lab. Sistema de Arquivos

Sistemas de Arquivos pode ser dividido em duas categorias de objetos:

- Dados de usuários: Dados reais armazenados em diretórios.
- Metadados – Informações estruturais do sistemas de arquivos, como diretórios, superblocos e inodes.

O espaço em disco é dividido em múltiplos blocos, os quais são usados para:

- Armazenar os dados de usuários (maioria dos blocos);
- Armazenar os metadados do sistemas de arquivos.

Lab. Sistema de Arquivos

Superblocos

O superbloco é um bloco especial que contém informações sobre o sistema de arquivos, como:

- Tipo do sistema de arquivos;
- Status;
- Tamanho;
- Estrutura de metadados;

O Linux mantém cópias do superbloco em cada sistema de arquivos.

Superblocos

```
$ dumpe2fs /dev/partição | grep -l superblock
```

É possível restaurar os backups de superblocos com o comando `fsck`. Os blocos se encontram nos seguintes locais:

- fs com blocos de 1K -> no bloco 8193
- fs com blocos de 2K -> no bloco 16384
- fs com blocos de 4K -> no bloco 32768

Lab. Sistema de Arquivos

inodes (index nodes)

Cada objeto no sistema de arquivos é representado por um inode. O inode identifica um arquivo ou diretório e seus atributos, como:

- Permissões
- Tamanho
- Proprietário e Grupo
- Tipo, hora de acesso e modificação

Cada inode é identificado por um número único no sistema de arquivos

Lab. Sistema de Arquivos

inodes (index nodes)

```
$ ls -li <nome do arquivo>
```

```
$ stat arquivo
```

É possível usar o número do inode para excluir um arquivo que o sistema não consiga excluir de forma convencional.

Recuperar .

Lab. Sistema de Arquivos

fsck

O comando fsck verifica as inconsistências em sistemas de arquivos

- A roda a verificação em todos os fs listados em /etc/fstab (no momento do boot)
- t tipo especifica um tipo de fs a ser verificado
- c Verifica se há bad blocks
- f Força uma verificação, mesmo que o fs apareça estar

Lab. Sistema de Arquivos

fsck

Para cada sistemas de arquivos existe uma ferramenta que faz a checagem apropriada para o sistema verificar

```
$ cd /sbin
```

```
$ ls *fsck*
```

Lab. Sistema de Arquivos

Criando arquivos de sistema

- O Linux usa o comando `mkfs` (*make filesystem*) para criar sistemas de arquivo e o comando `mkswap` para fazer espaço de troca.
- O comando `mkfs` é um front-end para diversos comandos específicos de sistema de arquivos.
- Verifique qual suporte de sistema de arquivos em seu sistema:

```
$ ls /sbin/mk*
```

Lab. Sistema de Arquivos

Criando um sistema de arquivos EXT2

É possível criar um sistema de arquivos EXT2 que poderá ser montado e acessado normalmente como se fosse uma partição normal. Isto é possível por causa do recurso loop oferecido pelo kernel do GNU/Linux

Lab. Sistema de Arquivos

Criando um sistema de arquivos EXT2

- Use o comando

`dd if= [origem] of= [destino] [opções]`

Exemplo:

`$ dd if=/dev/zero of=/diretorio/arquivo-ext2 bs=1024
count=10000`

Diretório onde será criado

Tamanho
do bloco

Cria um arquivo `arquivo-ext2` vazio de 10Mb de tamanho em `/diretorio`.

Lab. Sistema de Arquivos

Criando um sistema de arquivos EXT2

- Formate o arquivo com
`$ mkfs.ext2 /diretorio/arquivo-ext2.`

O Sistema foi criado está pronto para ser usado

Lab. Sistema de Arquivos

Criando um sistema de arquivos EXT2

Listando partições - fdisk

O comando fdisk com a opção -l é usado para listar a tabela de partições. Adicione um nome de dispositivo, como ex. /dev/zero, se quiser verificar as partições em uma unidade específica.

Obs. as ferramentas de particionamento exigem acesso do root.

```
$ fdisk -l /dev/zero
```

```
$ fdisk -l /dev/sda
```

As colunas *Start* e *End* mostram o cilindro que inicia e termina cada partição.

O campo *Id* indica o uso pretendido da partição.

Lab. Sistema de Arquivos

Criando um sistema de arquivos EXT2

Diretório lost+found

Este diretório é criado automaticamente após a formatação da partição com o `mkfs.ext2`

Sua função é pré-alocar os blocos de arquivos/diretórios durante a execução do programa `fsck.ext2` na recuperação de um sistema de arquivos

Lab. Sistema de Arquivos

Criando um sistema de arquivos EXT2

Monte o arquivo arquivo-ext2 com o comando:

```
$ mount /diretorio/arquivo-ext2 /mnt -o loop=/dev/loop1.
```

Note que foi usado o parâmetro -o loop para dizer ao comando mount para usar os recursos de loop do kernel para montar o sistema de arquivos.

Lab. Sistema de Arquivos

Criando um sistema de arquivos EXT2

Confira se o sistema de arquivos EXT2 em arquivo-ext2 foi realmente montado no sistema de arquivos digitando:

```
$ df -T
```

df apresenta informações relevantes sobre o sistema de arquivos

Lab. Sistema de Arquivos

Criando um sistema de arquivos EXT2

Para verificar as informações sobre o sistemas de arquivos:

```
$ dumpe2fs /dev/loop1
```

superblocos:

```
$ dumpe2fs /dev/loop1 | grep -l superblock
```

O

inode:

```
$ dumpe2fs /dev/loop1 | grep -l inode
```

Lab. Sistema de Arquivos

Criando um sistema de arquivos EXT2

```
$ umount -a arquivo-ext2
```

Lab. Sistema de Arquivos

- Operações como criação e remoção envolvem várias escritas em disco
- Um crash pode ocorrer durante o processo e
 - um bloco foi incluído no inode, mas não está marcado como em uso no bitmap
 - um bloco foi removido do inode, mas não está marcado como livre no bitmap
 - um bloco foi reutilizado, mas o inode original ainda aponta para ele.
 - um inode foi marcado como em uso, mas não há entrada em diretório apontando para ele.

Lab. Sistema de Arquivos

fsck e recuperação

Testando com debugfs

Exercício

- Crie um sistema de arquivos ext2 com `mkfs.ext2`
- Utilize o `debugfs` para corromper este sistema
- Verifique se o `fsck` encontra os erros

Lab. Sistema de Arquivos

Criando e fazendo verificação inicial

```
$ dd if=/dev/zero of=ext2.dmp bs=1k count=256
```

```
$ mkfs.ext2 ext2.dmp
```

```
$ fsck ext2.dmp
```

```
e2fsck 1.42.13 (17-May-2015)
```

```
ext2.dmp: clean, 11/32 files, 23/256 blocks
```

```
$ fsck -f ext2.dmp
```

```
e2fsck 1.42.13 (17-May-2015)
```

```
Pass 1: Checking inodes, blocks, and sizes
```

```
Pass 2: Checking directory structure
```

```
Pass 3: Checking directory connectivity
```

```
Pass 4: Checking reference counts
```

```
Pass 5: Checking group summary information
```

```
ext2.dmp: 11/32 files (0.0% non-contiguous), 23/256 blocks
```


Lab. Sistema de Arquivos

lost+found

```
$ debugfs: write teste.txt teste.txt
```

```
$ debugfs: modify_inode <12>
```

```
Link count [1] 2
```

← Modifique o link count para 2

```
debugfs: ls
```

```
2 (12) . 2 (12) .. 11 (20) lost+found 12 (980) a.txt
```

```
debugfs: rm teste.txt
```

```
debugfs: quit
```

```
$ fsck.ext2 -f ext2.dmp
```

Lab. Sistema de Arquivos

lost+found

- Por que o inode <12> seria colocado no diretório de “achados e perdidos”?

← Modifique o link count para 2

Obs. /lost+found – Local para a gravação de arquivos/diretórios recuperados pelo utilitário fsk.ext2. Cada partição possui seu próprio diretório lost+found

Lab. Sistema de Arquivos

Alterando bitmap dos inodes

```
$ debugfs -w ext2.dmp
```

```
debugfs: write teste.txt teste.txt
```

```
debugfs: ls
```

```
2 (12) . 2 (12) .. 11 (20) lost+found 12 (980) a.txt
```

```
debugfs: freei <12>
```

Como o fsck consegue corrigir o erro?