

# T3 - Sistemas de Arquivos

Vinicius Gasparini

SOP - BCC - UDESC - 2020.2

## 1 Exercício 1

- a) A alocação baseada em **extends** utiliza o par <endereço físico do primeiro bloco, quantidade de blocos contíguos a seguir> para descrever arquivos. Deste modo é possível reduzir significativamente a quantidade de metadados necessários para mapear arquivos grandes. Enquanto na alocação indexada cada bloco carece de uma chamada de alocação, com o uso de **extends** podemos simplificar a instrução de alocação ao agrupar de maneira contígua um conjunto de blocos e assim reservar um espaço maior de uma só vez.
- b) Diferentemente dos sistemas de arquivo ext2 e ext3 que utilizam *i-nodes* de 128 bytes, o sistema ext4 utiliza *i-nodes* de 256 bytes. Sendo assim, o sistema de alocação precisa suportar essa alteração bem como lidar com alocações de diversos blocos de uma só vez.
- c) O alocador para sistemas de arquivos ext4 necessita suporte a alocação multibloco. Como a premissa do ext4 é utilizar o máximo possível blocos contíguos, ao invés de realizar várias chamadas de alocação para cada bloco, evitamos *overhead* ao alocar um tamanho maior que represente  $x$  blocos.
- d) Na alocação indexada, um arquivo deste tamanho utilizaria apenas ponteiros diretos para os endereços dos blocos, tornando o acesso aleatório extremamente eficaz. Em contrapartida, ao utilizar **extends** seria mapeado o endereço do primeiro bloco e indicado que há 10 blocos seguintes.
  - No primeiro caso, onde devemos considerar que os blocos serão alocados de maneira contígua, o sistema baseado em **extends** causa menos *overhead* de acesso pois após localizar o primeiro bloco, o *i-node* já informa que os 9 próximos estão a seguir.
  - No caso onde os blocos não estarão dispostos em sequência, a alocação indexada se destaca pois os 10 endereços dispersos estão no *i-node*.

## 2 Exercício 2

- a) 73729 arquivos foram encontrados!
- b) O maior arquivo encontrado foi “usr/share/spotify/libcef.so” - 141.13 MB
- c) Existem 21 arquivos com tamanho igual a 0. Isso corresponde a 0.028% do total
- d) O tamanho médio é de 23.32 KB. 93.579% dos arquivos são de tamanho menor ou igual a esse média
- e) A mediana dos arquivos é 1.81 KB
- f) O menor tamanho de bloco tal que 50% dos arquivos ocupem apenas um bloco é de 2.0 KB
- g) Caso fosse adotado blocos de 2.0 KB, 39276 arquivos utilizariam apenas um bloco, correspondendo a 53.271% dos arquivos.
- h) O maior arquivo, “/usr/share/spotify/libcef.so” - 141.13 MB, utilizaria 72257 blocos
- i) Considerando que os ponteiros de disco sejam de 32 bits, 96.554% dos arquivos não precisam de indireção
- j) 3.395% dos arquivos precisam de indireção simples
- k) 0.052% dos arquivos precisam de indireção dupla
- l) 0% dos arquivos precisam de indireção tripla
- m) 9.637% do espaço alocado é desperdiçado por fragmentação interna. Isso corresponde a 179.09 MB de espaço em disco

Para os itens **i,j,k,l e m** foi necessário encontrar o tamanho de bloco utilizado pelo meu sistema. Para isso, foi utilizado o seguinte script bash.

```
$head /tmp/lista-arqs -n 1 | xargs stat
```

Tendo como retorno as seguintes informações:

```
File: /usr/bin/xhost
Size: 14328      Blocks: 32      IO Block: 4096   regular file
Device: 811h/2065d Inode: 5375195    Links: 1
Access: (0755/-rwxr-xr-x)  Uid: (   0/   root)   Gid: (   0/   root)
Access: 2021-03-12 08:21:11.086144360 -0300
Modify: 2018-03-18 19:19:30.000000000 -0300
Change: 2021-01-30 00:48:21.645940858 -0300
Birth: -
```

### 3 Exercício 3

Conforme tabela, o tamanho do bloco é de **2 KB** e o tamanho da área de dados é de **3 GB**.

Temos portanto,

$$\text{Tamanho do bloco} = 2 * 1024^1 = 2048 \text{ Bytes}$$

$$\text{Tamanho área} = 3 * 1024^3 = 3221225472 \text{ Bytes}$$

$$\text{Quantidade de blocos} = \frac{3221225472}{2048} = 1572864 \text{ blocos}$$

$$\text{Espaço necessário para mapa de bits} = \frac{1572864}{8} = 196608 \text{ Bytes}$$

$$\text{Espaço necessário para mapa de bits} = \frac{196608}{1024^1} = 192 \text{ KB}$$

### Referências

- Kernel Newbies - ext4
- UFSCAR LabSO - ext4
- Kernel Wiki - ext4 Howto
- Sans - Understanding EXT4