

I - nodes

Cada arquivo está associado a um único i-node no disco

Dado que:

- TBD – Tamanho bloco de dados
- TR – Tamanho referência (endereço do bloco)

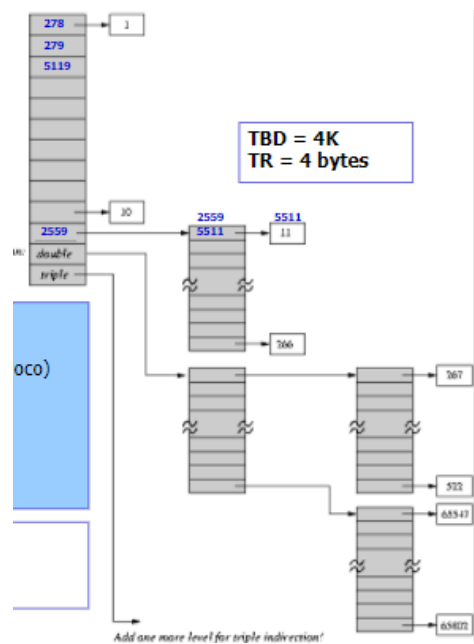
Qual será...?

- B_{max} – Nº Blocos máximo de um arquivo
- F_{max} – dimensão máxima de um arquivo

$$B_{max} = 10 + \frac{TBD}{TR} + \left(\frac{TBD}{TR}\right)^2 + \left(\frac{TBD}{TR}\right)^3$$

$$F_{max} = B_{max} \times TBD$$

Qual é o tamanho máximo de um arquivo com a seguinte configuração?

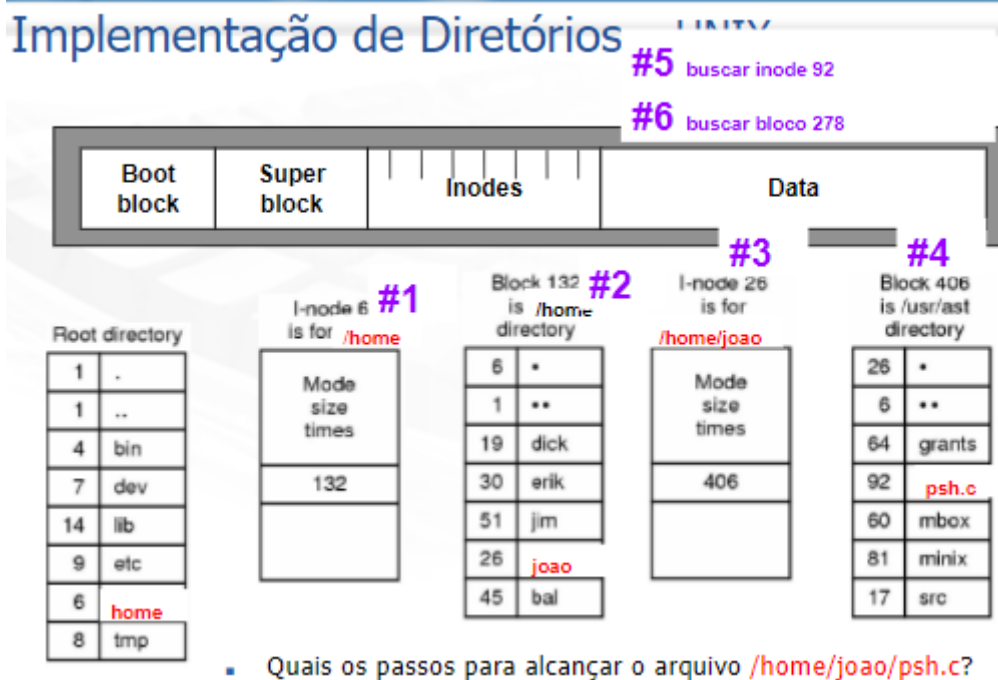


- $TBD/TR = 4 \text{ kbytes}/4 \text{ bytes} = 2^{12} \text{ bytes}/2^2 \text{ bytes} = 2^{10}$ (ponteiros que cabem em cada bloco)
- Na estrutura do I onde tem 10 ponteiros diretos + simple double + triple (indiretos)
 - simple = 1k
 - double = $1k^2 = 2^{20}$
 - triple = $1k^3 = 2^{30}$
- $B_{Max} = (10 + 2^{10} + 2^{20} + 2^{30})$ blocos
- $F_{max} \sim 2^{30} \times 4 \text{ kbytes} = 2^{30} \times 2^{12} \text{ bytes} = 2^{42} \text{ bytes} = 4 \text{ terabytes}$

Acessos a Disco

Joãozinho está implementando sua psh, e como um bom programador, ele usa Unix. Ele vai abrir o arquivo psh.c que se encontra no diretório /home/joao (a partição está formatada com ext3). Quantas operações em disco são necessárias para acessar o primeiro bloco do arquivo psh.c de Joãozinho, considerando que todos os diretórios dentro desse path ocupam apenas 1 bloco de disco? Suponha que o diretório raiz já esteja na memória, mas nenhum outro componente ao longo do caminho se encontre na memória.

Quantos acessos a disco?



ext3 = Tabela com duas informações por entrada, o nome e número do inode.

1 acesso -> já estando no diretório `/home` = possui inode 6 e deve buscar bloco 132

2 acesso -> no bloco 132, deve buscar o nome joão

3 acesso -> já estando no diretório `/joao` = possui inode 26 e deve buscar bloco 406

4 acesso -> no bloco 406, deve buscar o nome psh.c

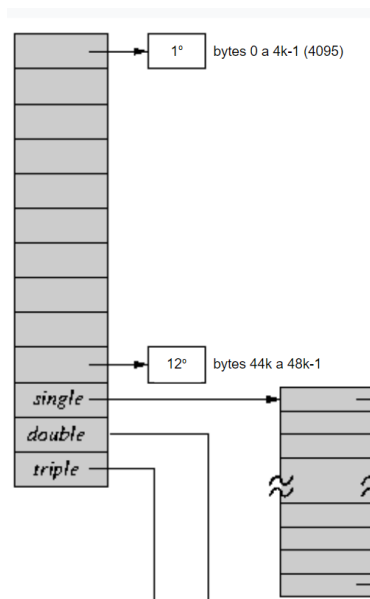
5 acesso -> já estando no diretório `/psh` =, busca pelo inode 92

6 acesso -> no inode 92, busca o bloco X.

Joãozinho se empolgou na sua implementação e após abrir o primeiro bloco, ele vai precisar editar o finalzinho do arquivo (ele esqueceu de desalocar umas áreas de memória heap). Com isso, Joãozinho vai precisar editar o arquivo a partir do byte 4.245.000 em diante (sim... o código de Joãozinho está gigante!). Quantos acessos a disco o ext3 vai precisar fazer para abrir o bloco que contém o byte 4.245.000, considerando que agora o inode do arquivo psh.c já está em memória. Considere ainda que no ext3 de Joãozinho (i) cada inode contém 12 ponteiros diretos para blocos, além dos ponteiros indiretos simples, duplo e triplo; (ii) o ponteiro para bloco ocupa 32 bits; (iii) o tamanho do bloco é 4 kbytes.

Informações fundamentais:

- (i) cada inode contém 12 ponteiros diretos para blocos, além dos ponteiros indiretos simples, duplo e triplo;
- (ii) o ponteiro para bloco ocupa 32 bits; (iii) o tamanho do bloco é 4 kbytes.



Como cada bloco tem 4 kbytes e o tamanho do endereço de bloco é 32 bits (4 bytes).

Dentro de um bloco cabem quantos ponteiros?

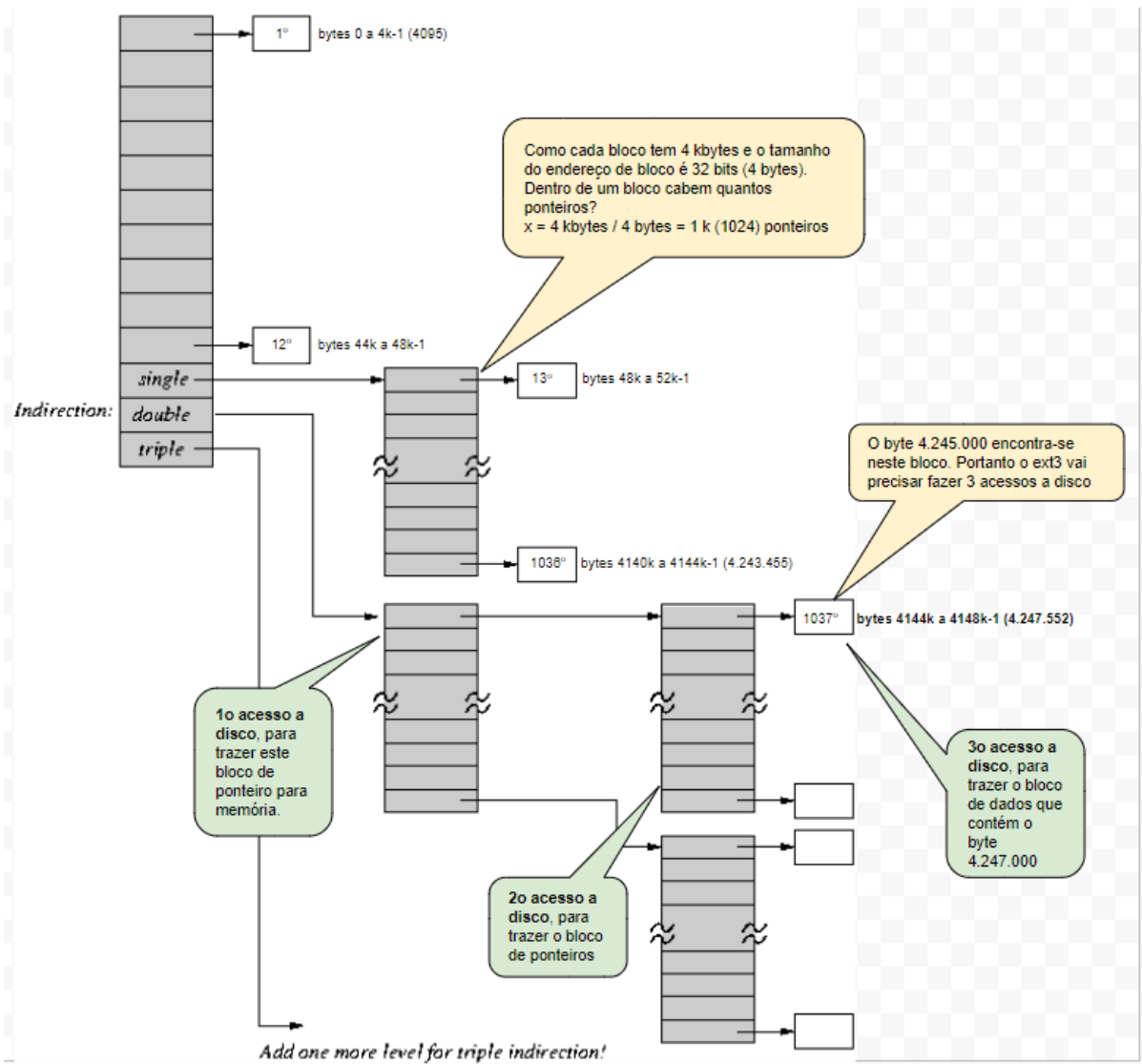
$x = 4 \text{ kbytes} / 4 \text{ bytes} = 1 \text{ k} (1024) \text{ ponteiros}$

- Até o momento temos 12 ponteiros (1 bloco), 1024 ponteiros (2 bloco) = 1036 blocos com o ponteiro indireto simples
- $1036 \times 4 \times 1024 = 4\,243\,456$ último byte que se consegue endereçar na estrutura simples indiretos + 12 diretos.

Como o endereço é de 4.245.000, não está no bloco indireto simples, e sim no próximo, o indireto duplo.

Número de acessos ao disco até chegar:

- 1 acesso -> acessar o disco para trazer o bloco de ponteiros para a memória;
- 2 acesso -> acesso a disco, para trazer o bloco de ponteiros
- 3 acesso -> acesso a disco, para trazer o bloco de dados que contém o byte 4.247.000



Quantos acessos ao disco teria caso o byte fosse 4 144k = 3 acessos