

1.

Marque a alternativa correta:

(F) No Linux, em um computador compatível com a família x86, um arquivo de 512 bytes pode sofrer fragmentação interna.

A fragmentação interna ocorre em um bloco e o tamanho mínimo de um bloco é 1k.

(V) No Windows, em um computador compatível com a família x86, um arquivo de 4096 bytes pode sofrer fragmentação interna.

(F) O maior tamanho de arquivo que pode ser armazenado em uma partição é o próprio tamanho da partição, para qualquer sistema de arquivos.

Tal afirmação depende do sistema de arquivos utilizado.

Ex:

Tamanho máximo do armazenamento físico: 1tb

Tamanho máximo do armazenamento lógico: 16gb

(F) O maior tamanho de partição é o próprio tamanho do disco, para qualquer sistema de arquivos.

Ao formatar um disco, uma pequena parte da memória recebe metadados do sistema de arquivos (utilizados pelo SO), com isso, o tamanho da partição não terá o mesmo tamanho do disco.

2.

Marque a alternativa incorreta:

(V) A cópia de um arquivo de um diretório para outro, na mesma partição, não envolve acesso à área de dados do arquivo.

(V) A cópia de um arquivo de um diretório para outro, em partição diferente, envolve acesso à área de metadados do arquivo.

(F) A fragmentação externa, ou fragmentação do sistema de arquivos, é reduzida com o método de alocação contígua.

Na verdade tal tipo de fragmentação ocorre justamente na alocação contígua. A fragmentação externa é reduzida com o método de alocação encadeada.

(V) O tamanho dos blocos (cluster) em um sistema de arquivos é definido na formatação da partição.

3.

Explique (tecnicamente, de acordo com o estudado na Unidade VI) porque um arquivo cujo conteúdo totaliza 100 bytes ocupa mais do que 100 bytes fisicamente no disco.

Por padrão, um setor (menor unidade de armazenamento físico) possui 512 bytes e um cluster (menor unidade de armazenamento lógico) possui 8 setores. Portanto, nesse caso, um bloco (pertence ao espaço físico) possui $512 * 8 = 4096$ bytes.

Devido a esses fatos, se um arquivo possui 100 bytes no armazenamento lógico, ele vai ocupar 4096 bytes no espaço físico.

4.

Assim como no sistema de arquivos UFS, estudado na na Unidade VI, o Linux utiliza o conceito de i-node. Liste e explique os passos que um usuário no Linux pode seguir para identificar o número de i-node de um arquivo e na sequência imprimir seus metadados associados.

Passo 1: Abra o terminal do linux

Passo 2: Digite no terminal do linux o seguinte comando

```
stat ./nome_do_arquivo
```

Ex:

```
stat ./test.c
```

File: ./test.c

Size: 3991 Blocks: 8 IO Block: 4096 regular file

Device: 360005ch/56623196d Inode: 329 Links: 1

Access: (0644/-rw-r--r--) Uid: (1000/ runner) Gid: (1000/ runner)

Access: 2021-05-23 20:15:44.837904221 +0000

Modify: 2021-05-04 05:16:37.393288379 +0000

Change: 2021-05-23 20:15:38.033248946 +0000

Passo 3: Identifique o inode

Passo 4: Identifique os metadados associados ao inode

5.

O sistema operacional Linux suporta a operação truncate em arquivos? Em caso positivo, explique como ela funciona e forneça um pequeno programa (em Linguagem C) que exemplifique o uso desta operação.

Função ftruncate:

Parâmetros:

int fd => Arquivo aberto
off_T length => Tamanho desejado

Funcionamento:

A função ftruncate altera o tamanho de um arquivo.

Se o tamanho desejado for menor que o tamanho do arquivo:

A parte que se inicia no tamanho desejado e termina no tamanho do arquivo é preenchida com bytes nulos ('\0')

Se o tamanho desejado for maior que o tamanho do arquivo:

A parte que se inicia no tamanho do arquivo e termina no tamanho desejado é preenchida com bytes nulos ('\0')

```
#include<stdio.h>
#include<fcntl.h>
#include<unistd.h>
#include<stdlib.h>

int main(void){

    int tamanho = 0;
    int fd = 0;
    int truncate = 0;
    tamanho = 100;

    fd = open("file.txt",O_WRONLY);
    if(fd == -1){
        printf("Erro ao abrir o arquivo no modo escrita"
        \n");
    }

    truncate = ftruncate(fd,tamanho);
    if(truncate == -1){
        printf("Erro ao truncar o arquivo\n");
    }

    printf("Arquivo truncado com sucesso!!\n");
    close(fd);
    return 0;
```

