

COM120 - Exercício Prático 11 – EP11 – 04/11/19

Em um certo SO, o disco rígido está organizado conforma Fig. 1:

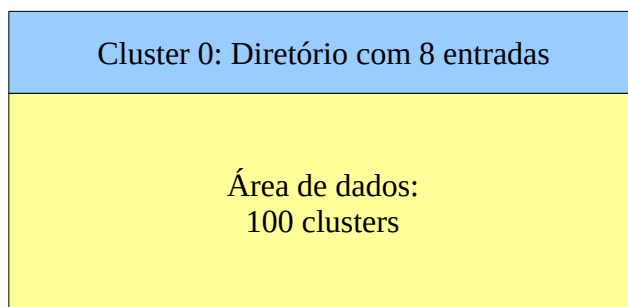


Fig. 1: Organização do disco

Cada entrada no diretório é feito utilizando as seguintes estruturas:

```
typedef struct  
{
```

```
    unsigned short dia;  
    unsigned short mes;  
    unsigned short ano;  
    unsigned short hora;  
    unsigned short min;  
    unsigned short seg;
```

```
}data;
```

```
typedef struct  
{
```

```
    char nome[8]; //nome do arquivo iniciado com o hexadecimal 0EBh são arquivos deletados  
    char extensao[3];  
    // Proteção  
    unsigned short sistema; //arquivo do sistema  
    unsigned short hidden; //arquivo oculto  
    unsigned short archived; //arquivo arquivado  
    // Data e hora da criação  
    data criacao;  
    // Data do último acesso  
    data acesso;  
    // Tamanho  
    unsigned long int tamanho;  
    // Cluster inicial  
    unsigned long int cluster;
```

```
}arquivo;
```

Os clusters de dados possuem a seguinte estrutura:

```
typedef struct  
{
```

```
    char dados[512];  
    unsigned long int prox;
```

```
} cluster;
```

Para acessar um arquivo no disco são necessários duas informações contidas na estrutura arquivo, o **tamanho** que é o tamanho do arquivo em bytes e **cluster** que é o número do cluster inicial, ou seja aponta para o primeiro cluster que deve ser lido. Cada cluster lido possui na sua estrutura um ponteiro para o próximo cluster, até encontrar o valor zero. A Fig. 2 apresenta um esquema da localização de um arquivo.

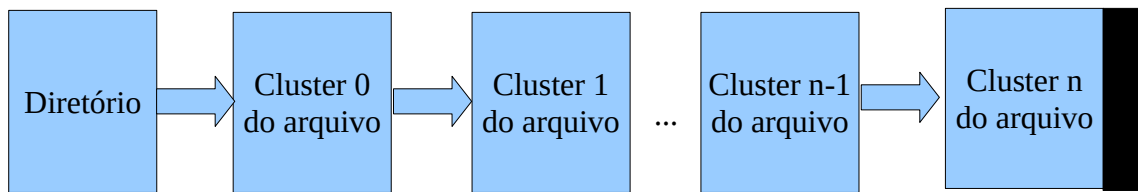


Fig. 2: Localização do arquivo no disco

Observando a Fig. 2 vemos uma parte preta no final do último cluster, é a parte não utilizada do cluster, esta é uma parte que não pode ser utilizada por nenhum outro arquivo. Podemos considerar fragmentação interna no cluster.

Escreva um programa em linguagem C que localize, leia e imprima um arquivo denominado Ext4.txt, dentro de Disco.dat, que é um arquivo binário simulando um disco rígido, este arquivo foi enviado por e-mail e possui 52.520 bytes. Imprima somente o conteúdo do arquivo, no último cluster encontramos informações que não são do arquivo. Após a impressão de Ext4.txt imprima a sequência dos clusters lidos, a quantidade de clusters, o tamanho do arquivo e a quantidade de bytes desperdiçado no último cluster.

Sugestão: quando trabalhamos com arquivos binários podemos utilizar recursos como a função `fseek()`.