LÍVIA STÉFFANNY DE SOUSA – 20.1.40.29

REVISÃO:

O que são arquivos?

Arquivos são unidades de armazenamento que devem ser utilizados quando não existe espaço em memoria principal para um grande volume de dados e quando é necessário armazenar permanente.

Quais tipos de arquivos em disco existem?

Existem dois tipos de arquivos em disco, arquivos de textos e arquivos binários. O primeiro é formado por caracteres de texto e o segundo formado por bytes (codificado em binário). Para utilizar qualquer um dos dois arquivos é preciso associar a uma variável logica e manipula-la.

Onde ocorre a associação dos arquivos?

A associação ocorre na operação de abertura do arquivo. Quando um arquivo é aberto ele é associado a uma variável logica.

Quais são as operações básicas em um arquivo?

As Operações básicas em um arquivo consistem em leitura(consulta), gravação(inclusão) e alteração ou exclusão de um dado.

Quais são os passos pra manipulação dos dados de um arquivo no próprio arquivo?

- Abrir ou criar o arquivo, associando o nome físico do arquivo ao nome logico;
-Manipular os dados do arquivo utilizando as operações básicas: consulta, inclusão, exclusão e alteração;
-Fechar o arquivo;

Pra que serve o ponteiro de um arquivo?

Serve pra fazer referência ao arquivo a ser tratado pelo programa. O ponteiro não aponta diretamente para o arquivo. Nele contém as seguintes informações sobre o arquivo: nome, situação (aberto ou fechado) e posição atual sobre o arquivo.

A primeira coisa a se fazer é criar uma variável que por meio dela manipulamos o arquivo. Depois é necessário associar o nome logico ao físico.

*FILE * Arquivo;*

O que é FILE?

O file é um registro que guarda o nome, a situação e a posição atual do arquivo.

Funções e o que fazem:

Fopen () Abertura de Arquivo

Arquivo = fopen (“nomefisico.txt”, “r”);

r	*Abre um arquivo texto existente para leitura*
w	*Cria um arquivo texto para escrita*
a	*Abre um arquivo texto para inserção no final*
r+	*Abre um arquivo texto existente para leitura e escrita*
w+	*Cria um arquivo texto para leitura e escrita*
a+	*Abre um arquivo texto para leitura e inserção no final*
rb	*Abre um arquivo binário existente para leitura*
wb	*Cria um arquivo binário para escrita*
ab	*Abre um arquivo binário para inserção no final*
r+b	*Abre um arquivo binário existente para leitura e escrita*
w+b	*Cria um arquivo binário para leitura e escrita*
a+b	*Abre um arquivo binário para leitura e inserção no final*

fclose () Fecha um arquivo

fclose (arquivo);

feof () Verifica o final de um arquivo

feof (arquivo);

putc () e fputc () Escreve um caractere em um arquivo texto

putc (caractere, arquivo);

getc () e fgetc () Lê um caractere de um arquivo texto

Se o ponteiro do arquivo estiver no final do mesmo ou ocorrer um erro na leitura, a função retorna EOF.

caractere = getc (arquivo);

fprintf () Permite impressão formatada em um arquivo texto

fprintf (arquivo, "formatação", variáveis);

fscanf () Permite leitura formatada de um arquivo texto

fscanf (arquivo, "formatação", variáveis);

fseek () Posiciona em um item (registro) de um arquivo binário, um dos comandos principais de manipulação de arquivos binários

fseek (arquivo, deslocamento, origem);

fwrite () Escreve tipos maiores que 1 byte em um arquivo binário, um dos comandos principais de manipulação de arquivos binários

fwrite (variável, tamanho, quantidade, arquivo);

fread () Lê tipos maiores que 1 byte de um arquivo binário, um dos comandos principais de manipulação de arquivos binários

fread (variável, tamanho, quantidade, arquivo);

fgets () Lê uma cadeia de um arquivo texto.

A função lê a cadeia até que um caractere de nova linha seja alcançado ou (tamanho - 1) caracteres tenham sido lidos.

fgets (cadeia, tamanho, arquivo);

fputs () Escreve uma cadeia em um arquivo texto

fputs (cadeia, arquivo);

PESQUISA EXTERNA:

Quando é usada a memória secundária?

Quando um conjunto grande de dados não cabe na memória principal.

Para que serve a Medida de complexidade?

É o que usamos para verificar se o método de pesquisa externa utilizado é considerado bom em relação a outros métodos.

O que é Sistema de paginação?

Uma estratégia que pode promover a implementação eficiente de qualquer método que envolva um grande volume de dados que se encontram na memória secundária. O arquivo é transformado em páginas e essas páginas são levadas para memória quando necessário. Tem como objetivo tentar diminuir a quantidade de acessos na memória secundaria para trazer dados para memória principal.

O que são páginas ativas e inativas?

As páginas ativas são aquelas que se encontram em memória principal e as inativas que ainda se encontram no arquivo.

Quais são as funções do mecanismo do sistema de paginação?

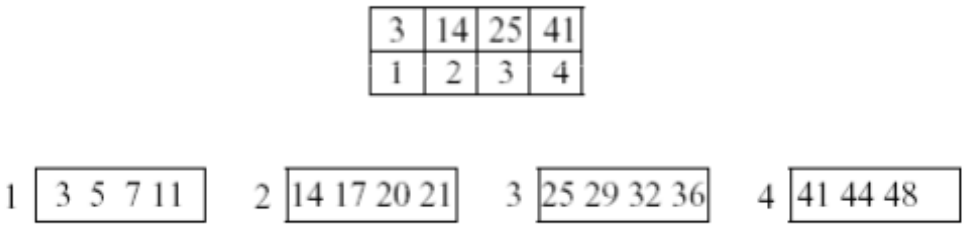
Ele possui duas funções principais, a transferência de páginas onde ocorre a transferência da memória secundaria para a principal e vice-versa e o Mapeamento de endereços que determina qual pagina da memória secundária um programa está endereçado.

Quais são as políticas de remoção de páginas da memória principal?

Menos recente utilizada: É criado uma fila, que guarda os acessos feitos as páginas que estão em memoria principal, desta forma, a página do início é a menos acessada.
Menos frequentemente utilizada: Utilizando um contador, é possível retirar uma página cujo contador é o menor.
Ordem de Chegada: A página que chega primeiro é a primeira a sair. Remove a página que se encontra na memória principal há mais tempo.

Métodos de pesquisa externa

***Acesso sequencial indexado**: Realiza uma pesquisa, de forma sequencial, trabalhando em cima de um índice, facilitando a busca da chave desejada (arquivo precisa estar ordenado).*

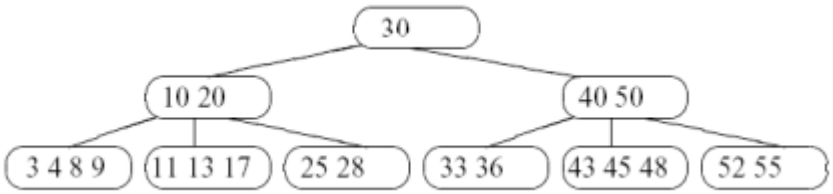


Inicialmente é preciso criar um índice de páginas que armazenam a chave do primeiro item de cada página e o endereço dessa página. Posteriormente, é usado o índice de páginas para encontrar a chave desejada utilizando a comparação da chave desejada com a chave do primeiro item de cada página e depois carregar a página encontrada para a memória principal.

***Arvore B:** É uma estrutura de dados em árvore, que armazena dados. Uma das principais características é que ela é formada por páginas, pode ter mais de dois filhos, ou seja, pode conter vários itens.*

Árvore B de ordem m > página raiz: contém entre 1 e 2m itens

> *demais páginas: contém, no mínimo, m itens e m+1 descendentes e, no máximo, 2m itens e 2m+1 descendentes*
> *páginas folhas: aparecem todas no mesmo nível;*
> *itens: aparecem dentro de uma página em ordem crescente, de acordo com suas chaves, da esquerda para a direita.*



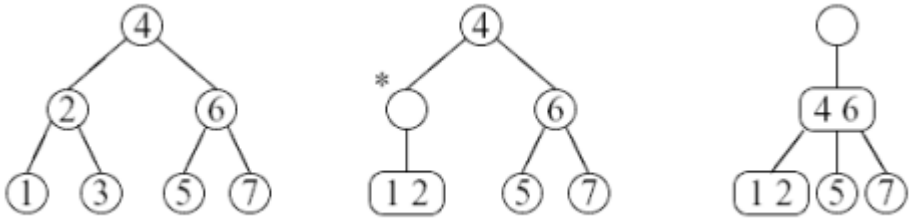
Cada uma das páginas é formada por vários itens e por vários filhos já que é sempre um filho a mais do que a quantidade de itens presentes dentro de uma página. Todas as páginas folhas aparecem no mesmo nível, são arvores sempre balanceadas e os itens sempre dentro de uma página de forma ordenada da esquerda para a direita, ou seja, todos os itens menores que a raiz a esquerda e os maiores a direita.

A busca pelas chaves é feita por meio de comparações. É necessário comparar a chave do item desejado com as chaves que estão na página raiz ou o intervalo no qual ela se encaixa até encontrar a chave desejada.

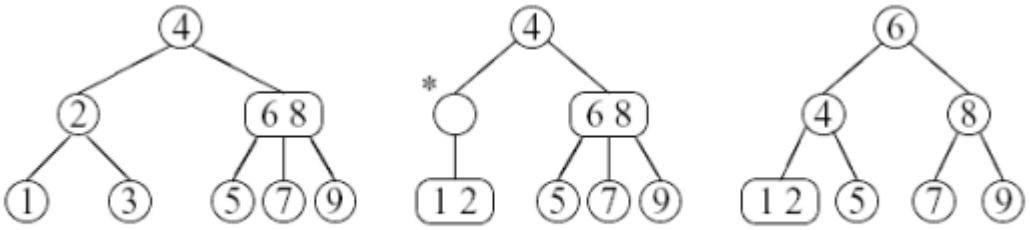
Para a inserção de um item é necessário primeiramente localizar a página que ele deve ser inserido. Depois é preciso conferir se o item a ser inserido encontra uma página com menos de 2m itens. Pode acontecer de encontrar uma página cheia, então é necessário criar uma página para dividir os itens entre essa nova página, a página que estava cheia anteriormente e a página pai de ambas.

Para a remoção de um item é importante lembrar que só é possível remover um item presente na página folha. Se o item a ser removido não for uma página folha é necessário substitui-lo pelo seu sucessor ou antecessor. Posteriormente deve ser feita uma pesquisa a partir da raiz da arvore e realizar o caminhamento fazendo as comparações necessárias. Encontrando o item, é necessário verificar se pertence a uma página folha e rearranjar o vetor. Depois faz-se necessário verificar se não foi violado nenhuma regra da arvore b. Se tiver sido violada, é necessário tomar emprestado um item da página do lado:

Se a página vizinha possuir m itens as páginas se juntam, tomando emprestado da página pai o item do meio e permitindo liberar uma das páginas.



Se a página vizinha possuir mais que m itens é necessário pegar emprestado um item da página do lado por meio da página pai.



Importante:

```
typedef long TipoChave;
typedef struct TipoRegistro {
        TipoChave Chave;
                /* outros componentes */
} TipoRegistro;
typedef struct TipoPagina* TipoApontador;
typedef struct TipoPagina {
        short n;
        TipoRegistro r[MM];
        TipoApontador p[MM + 1];
} TipoPagina;
```

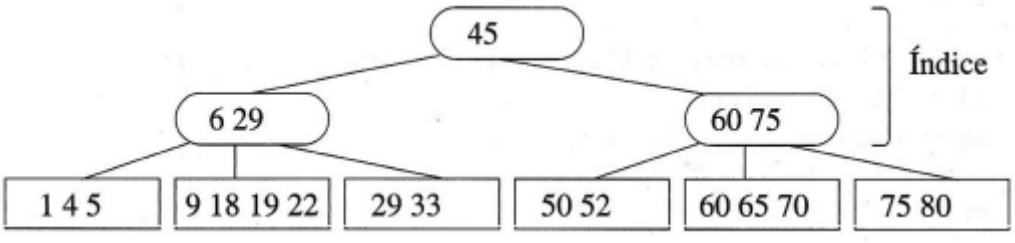
Como inicializar:

```
void Inicializa (TipoApontador Arvore) {
        Arvore = NULL;
}
```

Caminhamento:

```
void Imprime (TipoApontador arvore){

        int i = 0;

        if (arvore == NULL)

                return;

        while (i <= arvore->n) {

                Imprime(arvore->p[i]);

                if (i != arvore->n)

                        cout << arvore->r[i].Chave << " ";

                 i++;
        }
}
```

Arvore B*: É uma forma de implementar arvore B, com algumas características que as diferem, como: todos os itens estão armazenados no nível das paginas folhas e os níveis acima do último servem apenas para direcionar a pesquisa de um determinado registro que se encontra dentro de uma página folha que pode ou não estar conectada da esquerda para a direita.



Para realizar uma pesquisa é preciso levar em conta que a pesquisa sempre termina em uma página folha, ao encontrar a chave desejada em uma página do índice, a pesquisa não termina, a pesquisa continua até que se encontre uma página folha.

A inserção é semelhante à de uma arvore B. Elas se diferem somente que na arvore b*, o algoritmo faz uma cópia da chave que pertence ao item do meio para a página pai no nível anterior quando se divide uma folha em duas.

A remoção também é semelhante a uma arvore B. Como todos os itens estão armazenados no nível das páginas folhas o item a ser removido sempre estará em uma página folha. Porem se a folha ficar com menos itens que m as páginas do índice precisarão ser rearranjadas.