**Arquivo Sequencial**

**Operações: Acesso**

**Acesso Serial**

É um tipo de acesso que consiste na obtenção do registro que segue ao último acessado, na seqüência ditada pela chave de ordenação. Em um arquivo sequencial, este tipo de acesso é extremamente eficiente por estarem fisicamente armazenados de acordo com a sequência na qual são solicitados . Deste modo , na maioria dos acessos o registro desejado estará presente na memória, por pertencer ao mesmo bloco do seu antecessor.

Chave de ordenação é o atritubo NUMERO.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **NÚMERO** | **NOME** | **SALÁRIO** |
| 1 | 100 | PEDRO | 3000 |
| 2 | 150 | JOÃO | 1500 |
| 3 | 200 | MARIA | 2500 |
| 4 | 250 | CARLA | 3050 |
| 5 | 300 | MAX | 2000 |

Exemplo:

Pesquisa Sequencial

Base de Dados (100 150 200 250 300)

Valor procurado: 250

250 é igual a 100? Não

250 é igual a 150? Não

250 é igual a 200? Não

250 é igual a 250? Sim

Fim da pesquisa com sucesso!

Total de passos: 04

**Acesso Aleatório**

Caracteriza-se por ser a indicação do registro desejado feita pela especificação de um argumento de pesquisa. Neste caso a sequência na qual os registros são acesados nào mantém necessariamente a relação com a ordenação física do arquivo, podem ocorrer dois casos  no acesso aleatório :

Chave de acesso ≠ Chave de Ordenação

        Neste caso é utilizado a pesquisa seqüencial, onde é examinado cada registro, até ser localizado ou não, o registro que  possui para a chave de acesso um valor igual ao pesquisado.

Exemplo:

Pesquisa Sequencial

Base de Dados (3000 1500 2500 3050 2000)

Valor procurado: 3050

3050 é igual a 3000? Não

3050 é igual a 1500? Não

3050 é igual a 2500? Não

3050 é igual a 3050? Sim

Fim da pesquisa com sucesso!

Total de passos: 04

Chave de acesso = Chave de Ordenação

        Se o argumento estiver armazenado em um dispositivo de acesso serial a vantagem é a constatação mais rápida que o argumento de pesquisa não está no arquivo , quando for o caso. Porém, se o arquivo está armazenada em dispositivo  de acesso direto(HD’s, CD’s), é utilizado um método de pesquisa, mais eficiente, a Pesquisa Binária. Porém, em um arquivo seqüencial, só existe a pesquisa seqüencial, pois ele não dispõe acesso direto.

Exemplo:

Pesquisa Binária

Base de Dados (100 150 200 250 300)

Valor procurado: 250

Passo 1 => Início: 0 Fim: 4 Meio: 2

250 é igual a 200(meio)? Não => 250 é menor que 200(meio)? Não

Passo 2 => Início: 3 Fim: 4 Meio: 3

250 é igual a 250(meio)? Sim => 13 é menor que 15? Sim

Fim da pesquisa com sucesso!

Total de passos: 02

Com a pesquisa binaria, o número máximo de comparações necessários para localizar um registro ou determinar que ele não está presente no arquivo é igual a []+1, sendo *N* o número de registros do arquivo. Este desempenho é melhor do que o de pesquisa sequencial tendo como calculo (N+1) / 2.

**Operações: Inserção de um registro**

A maneira usual de processar inserções de registros em um arquivo sequencial *S* consiste em montar um arquivo *T* de transações contendo os registros a serem inseridos, ordenado pela mesma chave de ordenação de *S.* Os arquivos *S* e *T* são estão intercalados, gerando o arquivo *A* que é a versão atualizada de *S*.

Este procedimento é adotado, porque a inserção de um registro isolado seria muito custoso, pois todos aqueles resgistros com chaves superiores ao inserirdo seriam necessariamente deslocados de sua posição, o que é inviável em se tratando de arquivos armazendos em um meio externo.

O arquivo *T* pode ser usado como uma extensão de *S,* até assumir um tamanho que justifique a efetivação da operação de intercalação.

**Operações: Exclusão de um registro**

É geralmente implementado como a inserção, utilizando um arquivo de transações. As indicações de exclusão são coletadas no arquivo de transações para posterior efetivação das operações. Pode, ainda, ser implementada com o uso de um campo adicional que indique o estado de cada registro, sendo “inativo”. Neste caso, a operações de exclusão consiste em localizar o registro a ser excluido e alterar o seu campo de estado, assim os registros marcados como “inativo” são desconsiderados. Com este procedimento, é eliminada a necessidade de movimentação de outros registros para o preenchimento do espaço liberado pelo excluído.

Exemplo:

Ativo = 1

Inativo = 0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **STATUS** | **NÚMERO** | **NOME** | **SALÁRIO** |
| 1 | 1 | 100 | PEDRO | 3000 |
| 2 | 1 | 150 | JOÃO | 1500 |
| 3 | 0 | 200 | MARIA | 2500 |
| 4 | 1 | 250 | CARLA | 3050 |
| 5 | 0 | 300 | MAX | 2000 |

Exemplo:

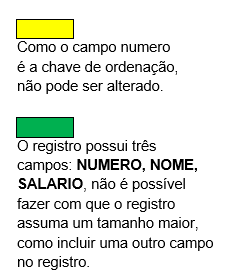
ARQUIVO *S* ARQUIVO *T*

|  |  |  |  |
| --- | --- | --- | --- |
|  | **NÚMERO** | **NOME** | **SALÁRIO** |
| 1 | 100 | PEDRO | 3000 |
| 2 | 150 | JOÃO | 1500 |
| 3 | 200 | MARIA | 2500 |
| 4 | 250 | CARLA | 3050 |
| 5 | 300 | MAX | 2000 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **NÚMERO** | **NOME** | **SALÁRIO** |
| 1 | 250 | CARLA | 3050 |
| 2 | 150 | JOÃO | 1500 |

**Operações: Alteração de um registro**

A operação de alteração consiste na modificação do valor de um ou mais atributos de um registro. O registro deve ser localizado, lido e seus campos alterados, sendo, após, gravados novamente. Não pode ocorrer uma alteração que faça com que o registro assuma um tamanho maior do que o original, pois não poderá ocorrer a gravação do novo registro na sua posição original e não se pode modificar o valor do campo que deternima a sequência dos registros dentro do arquivo (chave de ordenação).

Exemplo:

Chave de ordenação é o atritubo NUMERO.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **NÚMERO** | **NOME** | **SALÁRIO** |
| 1 | 100 | PEDRO | 3000 |
| 2 | 150 | JOÃO | 1500 |
| 3 | 200 | MARIA | 2500 |
| 4 | 250 | CARLA | 3050 |
| 5 | 300 | MAX | 2000 |

**Operações: Leitura exaustiva dos registros**

A leitura de todos os registros do arquivo com base na chave de ordenação, exige a manipulação do arquivo principal *S* e do arquivo *T* de transações. Neste processo, são desconsiderados os registros marcados com status “inativo”. O próximo registro a ser considerado é aquele que possuir a menor chave, entre o próximo registro de *S* e o próximo de *T*.

Exemplo:

ARQUIVO *S*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **STATUS** | **NÚMERO** | **NOME** | **SALÁRIO** |
| 1 | 1 | 100 | PEDRO | 3000 |
| 2 | 0 | 150 | JOÃO | 1500 |
| 3 | 1 | 200 | MARIA | 2500 |
| 4 | 1 | 250 | CARLA | 3050 |

ARQUIVO *T*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **STATUS** | **NÚMERO** | **NOME** | **SALÁRIO** |
| 1 | 1 | 300 | CARLA | 3050 |
| 2 | 0 | 308 | JOÃO | 1500 |
| 3 | 0 | 422 | MAX | 2000 |