

Organização de arquivos:

Arquivo : Coleção de registros

Registro Lógico: Sequência de itens(campos ou atributos do registro)

Registro Físico: Armazenamento dos registros lógicos em blocos

Chave: Sequência de um ou mais atributos de um registro

Chave primária: Atributo que identifica um registro como único

Chave secundária: Atributo para identificação(em índices) e pode ter seu valor repetido

Chave de acesso: Utilizada para identificar registros em uma leitura

Argumento da pesquisa: valor da chave de acesso

Chave de um registro: Valor da chave primária

Chave de ordenação: Chave primária utilizada para estabelecer a sequência na qual devem ser dispostos os registros de um arquivo.

Arquivo Sequencial:

Registros distribuídos de forma arbitrária.

Ordem normalmente é a da geração dos registros.

Vantagem:

- Simples implementação.

Desvantagem:

- Busca de registro por meio de acesso sequencial.
- Ineficiente quando há muitos registros.

Arquivo Sequencial Ordenado:

Registro dispostos sequencialmente definidos por uma chave de ordenação.

Registros possuem o mesmo formato, o que pode causar desperdício de posições de armazenamento.

Vantagem:

- Operação de acesso sequencial eficiente quando a chave de acesso coincide com a ordenação.
- Operação de acesso sequencial eficiente quando a leitura dos registros do arquivo seguem na sequência da chave de ordenação.

Desvantagem:

- Operação de leitura é ineficiente quando a chave de acesso não coincide com a de ordenação.
- Operação de gravação no arquivo(inscrição, alteração e remoção) pode requerer uma ordenação.

Inserção de um registro com balance-line.

Balance-line consiste em gravações realizadas em um arquivo temporário que periodicamente são efetivadas no arquivo original, nessa efetivação é realizada a intercalação dos arquivos.

Exclusão também utiliza balance line ou atributo adicional(exclusão apenas lógica, ou seja o arquivo é apenas ignorado).

Alteração utiliza balance-line, pode aumentar o tamanho de um registro e modificar o campo usado como chave de ordenação.

Leitura exaustiva dos registros manipula em paralelo os arquivos mestres e transação.

Reorganização do arquivo: Operação de intercalação entre os arquivos mestre e transação.

Arquivo Sequencial-Indexado:

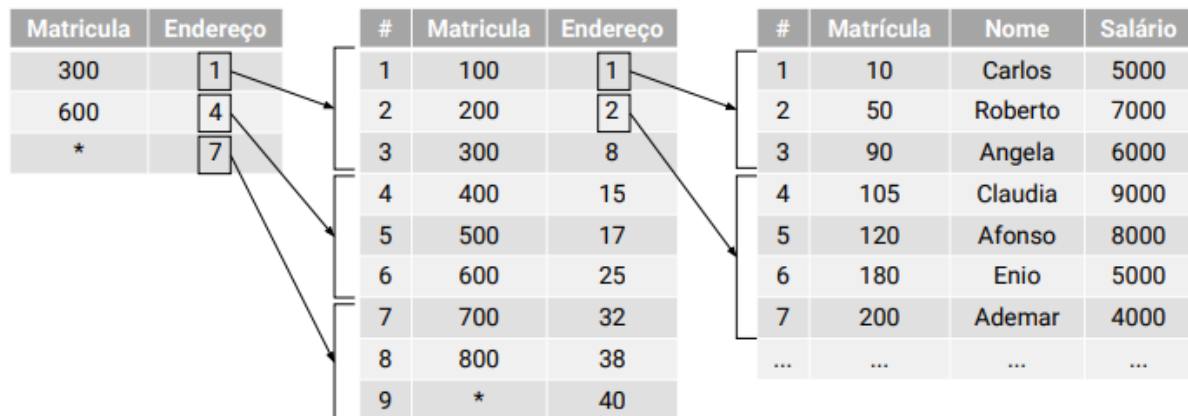
Arquivo sequencial com índice e área de extensão.

Área de índices : Arquivo sequencial gerado pelo sistema, cada registro deste arquivo estabelece uma divisão na área primária, contém o endereço do início do segmento e a chave mais alta do mesmo. Similar a busca por capítulo de um livro.

Área primária: Reservada para manter os registros que são classificados em ordem ascendente pela chave primária.

Área de excedentes: Reservada para novos registros que não são mantidos na área principal.

Índice: Cada entrada possui um valor de chave e o endereço do arquivo, permite uma rápida localização do arquivo, pode ser organizado em múltiplos níveis.



Não é viável realizar a inserção de registro igual em um arquivo sequencial pois os registros podem mudar de endereço e isso altera as entradas dos índices, por isso os registros inseridos após a criação são mantidos em uma área de extensão, que pode apresentar 2 modos:

Modo 1: cada registro possui um encadeamento indicando seu antecessor e sucessor.

Modo 2: usar um atributo para encadeamento de cada bloco de registro contendo a lista de extensão do bloco.

Operações de leitura:

-Acesso sequencial sem utilizar o índice.

-Acesso aleatório com uso do índice para encontrar o registro ou seu bloco(nesse caso se busca dentro do bloco).

-Leitura exaustiva: igual acesso sequencial.

Operações de gravação:

-Inserção com áreas de extensão.

-Exclusão com atributo adicional(exclusão lógica)

-Alteração se não afetar a chave de ordenação sobrescrever o registro, do contrário se usa inclusão e exclusão.

Reorganização:

A reorganização deve ser realizada periodicamente, pois as exclusões e inclusões podem degradar o desempenho das operações e as áreas de extensão devem ser integradas.

Arquivo Indexado:

Permite acesso aleatório satisfatório.

Acesso sequencial eficiente pela chave primária.

Facilita inserção e exclusão de registros pelo uso da área de extensão.

Os arquivos sequenciais requerem os registros fisicamente ordenados, o que dificulta as operações e torna a manutenção inviável quando a frequência dos acessos sequenciais for baixa e a dos acessos aleatórios alta.

Existência de um ou mais índices por registro, sem compromisso com a ordem física dos registros e considera a possibilidade de acesso por qualquer atributo do registro.

Suporte a múltiplos índices: pode existir um para cada chave de acesso dos registros.

Classificações de índices:

-Exaustivo : Uma entrada para cada registro do arquivo.

-Seletivo: Uma entrada para cada subconjunto de registros.

Entrada	Matrícula	Endereço		Endereço	Matrícula	Nome	Data nasc	Depto	Salário
1	1000	301		301	1000	Ademar	11/02/90	A	5000
2	1010	302		302	1010	Roberto	17/01/85	B	7500
3	1020	303	→	303	1020	Gerson	05/12/88	A	6000
4	1030	304		304	1030	Ieda	18/05/63	C	9000
5	1040	305		305	1040	Bernardo	14/12/92	C	4500
6	1050	306		306	1050	Angela	15/02/95	C	6500

Índice exaustivo (primário)		Área de dados	
Depto	Entrada	Salário	Entrada
A	1, 3	5000	1, 5
B	2	6000	3, 4
C	4, 5, 6	7500	2, 6

Índice seletivo (departamento) Índice seletivo (salário)

Operações de leitura:

-Acesso sequencial: Utiliza-se o índice apropriado pois as entradas dos índices são ordenadas. A memória mantém um bloco do índice, reduzindo acessos a memória secundária.

-Acesso aleatório: Busca completa pelos índices que possuem as chaves de acesso.

-Leitura exaustiva: Sucessivos acessos sequenciais sobre o índice exaustivo.

Operações de gravação:

-Inclusão: Registros podem ser armazenados em qualquer endereço disponível e seus pares são inseridos nos índices correspondentes, utiliza a Árvore B para tratamento dos índices.

-Exclusão: Libera a área de dados ocupadas pelo registro e são removidas as entradas nos índices correspondentes a esses registro.

-Alteração: Faz uma busca dos registros e seus atributos são alterados na mesma posição.

Vantagem:

-Inserção mais eficiente.

-Possibilidade de acessos aleatórios a partir dos índices.

Desvantagem:

-Acesso sequencial ineficiente.

-Necessidade de manutenção de índices.

-Inserções ou alterações envolvendo atributos associados aos índices.

Arquivo Direto:

Ao invés de utilizar índices, se utiliza uma função hash para calcular o endereço do registro a partir do valor da chave de registro.

Acesso rápido aos registros especificados por argumento de pesquisa sem percorrer uma estrutura auxiliar.

Acesso aleatório eficiente.

Não são previstos acessos seriais.

Funções para cálculo do endereço:

-Funções determinísticas: Para cada chave de acesso sempre gera um único endereço.

-Funções probabilísticas: O valor da chave de acesso gera um valor tão único quanto possível, podendo haver colisões (duas chaves diferentes gerando o mesmo endereço). O objetivo dessas funções é preservar a ordem dos registros e aumentar o grau de unicidade dos registros sobre o arquivo.

Tratamento de colisões:

-Endereçamento aberto: O endereço colidido é salvo no primeiro endereço livre.

-Encadeamento: Busca um endereço e adiciona uma ligação ao registro anterior, podendo formar uma lista encadeada ou utilizando áreas de extensão.

Operações de leitura:

-Acesso sequencial: Só é possível quando utilizada uma função que preserve a ordem dos registros, nesse caso basta ler a área de dados.

-Acesso aleatório: Aplica-se função hashing.

-Leitura exaustiva: Mesmo princípio do acesso sequencial.

Operações de gravação:

-Inserção: Aplica função hashing.

-Exclusão: Exclusão lógica.

Alteração: Quando não há chave de acesso, o registro deve ser localizado e alterado, caso contrário, o registro é excluído e inserido.

Arquivo Invertido:

Diferente dos demais arquivos, este utiliza técnicas para organizar por chaves secundárias.

Inversão: conjunto de listas invertidas ligadas a uma chave de acesso.

Endereço	Matrícula	Nome	Data nasc	Depto	Salário
301	1000	Ademar	11/02/1990	A	5000
302	1010	Roberto	17/01/1985	B	7500
303	1020	Gerson	05/12/1988	A	6000
304	1030	Ieda	18/05/1963	C	9000
305	1040	Bernardo	14/12/1992	C	4500
306	1050	Angela	15/02/1995	C	6500

Área de dados

Depto	Endereço
A	301, 303
B	302
C	304, 305, 306

Índice invertido

Vantagem:

Permite acesso direto a um conjunto de registros.

Desvantagem:

Listas só são válidas para aquela disposição física.

Hierarquia dos dispositivos de memória:

Aspectos chave da memória:

- Velocidade de acesso.

- Capacidade de armazenamento.

O ideal seria memória de tamanho ilimitado e com tempo de acesso muito rápido, a solução é criar uma ilusão para o processador de forma que a memória seja vista desse jeito.

Tipos de localidade:

- Temporal: Se um endereço é referenciado, ele tende a ser referenciado novamente.

- Espacial: Se um endereço é referenciado, endereços próximos tendem a ser referenciados em breve.

Hierarquia de memória:

- Registradores: dentro do processador.

- Cache: próximos ao processador.

- Memória principal: memória RAM.

- Memória secundária: HDs, DVDs, etc.

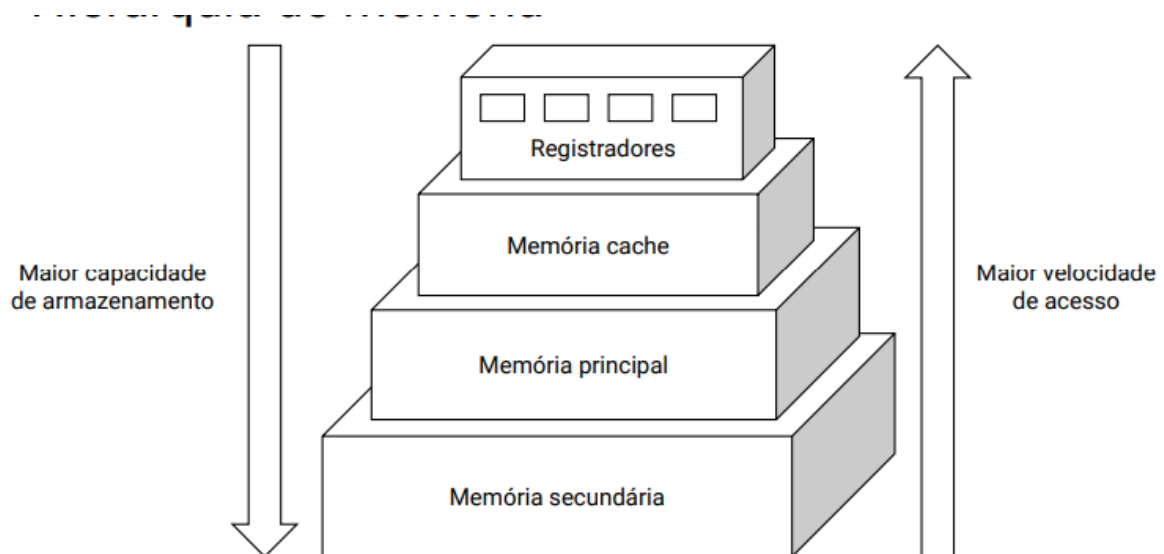
Tipos diferentes de memória existem devido a diferentes tempos de acesso, capacidade de armazenamento e custo.

Duas operações: Armazenar dados na memória e recuperar dados armazenados.

Célula é a unidade de armazenamento na memória principal.

Operação de escrita é destrutiva: Armazena o novo conteúdo sobre o anterior.

Operação de leitura não é destrutiva: Copia o valor de origem sem modificar.



Volatilidade: Capacidade de reter ou não informação quando a energia elétrica é desligada.

Memória volátil: Não retém a informação(Registradores, Cache e principal).

Memória não volátil: Retém a informação(Memória secundária).

Registradores:

- Armazenam instruções e dados que estão sendo manipulados pelas operações executadas pelo processador.(Armazenam um dado, instrução ou endereço).

Memória Cache:

- Criada para acelerar a velocidade de transferência das informações entre memória principal e processador.

Métricas mais comuns:

- Hit: Dado encontrado no nível procurado.

- Miss: Dado não encontrado no nível procurado.
- Hit-rate: Percentual de hits no nível.
- Miss-rate: Percentual de miss no nível.
- Hit-time: Tempo de acesso ao nível para verificar se é hit ou miss.
- Miss-penalty: Tempo médio gasto para acesso ao dado não encontrado(Tempo para transferir o dado para um nível mais inferior que o desejado).

Memória principal:

- Dispositivo onde o programa que vai ser executado é armazenado para que o processador leia as instruções.

Memória secundária:

- Alta capacidade de armazenamento.

Algoritmo de Levenshtein:

```

#include <stdio.h>
#include <string.h>
int min(int a, int b, int c)
{
    if (a <= b && a <= c)
    {
        return a;
    }
    else if (b <= a && b <= c)
    {
        return b;
    }
    else if (c <= a && c <= b)
    {
        return c;
    }
}
int levenshtein(char *s1, char *s2)
{
    unsigned int x, y, s1len, s2len;
    s1len = strlen(s1);
    s2len = strlen(s2);
    unsigned int matrix[s2len + 1][s1len + 1];
    matrix[0][0] = 0;
    for (x = 1; x <= s2len; x++)
        matrix[x][0] = matrix[x - 1][0] + 1;
    for (y = 1; y <= s1len; y++)
        matrix[0][y] = matrix[0][y - 1] + 1;
    for (x = 1; x <= s2len; x++)
        for (y = 1; y <= s1len; y++)
            matrix[x][y] = min(matrix[x - 1][y] + 1, matrix[x][y - 1] +
1, matrix[x - 1][y - 1] + (s1[y - 1] == s2[x - 1] ? 0 : 1));

    return (matrix[s2len][s1len]);
}
void main()
{
    char string1[] = "exercacio";
    char string2[] = "exarcito";
    printf("Palavra 1: %s\n", string1);
    printf("Palavra 2: %s\n", string2);
    int resposta = levenshtein(string1, string2);
    printf("Alteracoes necessarias: %d", resposta);
}

```

```
} _____
```