

Prof. Nilton César de Paula

ARQUIVO BINÁRIO



Arquivos

- São estruturas especiais que ficam armazenadas na memória secundária (fita magnética, disquete, HD, pendrive, ...)
- Principais operações sobre arquivos
 - ▣ Consultar
 - ▣ Acrescentar novos dados
 - ▣ Modificar dados
 - ▣ Eliminar dados
- Tipos de arquivos
 - ▣ Binário
 - ▣ Texto

Operações sobre arquivos

□ Consultar

1. Abrir o arquivo
2. Achar a ficha procurada
3. Copiar informações da ficha
4. Fechar o arquivo

□ Acrescentar novos dados

1. Abrir o arquivo
2. Achar posição de inserção
3. Guardar nova ficha
4. Fechar o arquivo

● Modificar dados

1. Abrir o arquivo
2. Achar ficha procurada
3. Alterar dados da ficha
4. Guardar ficha alterada
5. Fechar o arquivo

● Eliminar dados

1. Abrir o arquivo
2. Achar ficha procurada
3. Retirar a ficha do arquivo
4. Fechar o arquivo

Arquivo Binário

- É o tipo mais comum que armazena dados em forma de registros, conhecido também de arquivo de registros;

Tipo

CAD_ALUNO = registro

nome : caracter;

nota1 : real;

nota2 : real;

media : real;

Fim_registro;

- Cada registro é armazenado um após o outro no arquivo;
- Para o acesso aos registros do arquivo a aplicação deve conhecer o formato padrão do arquivo;
- O acesso a um registro no arquivo pode ser seqüencial ou direto;

Arquivo Binário

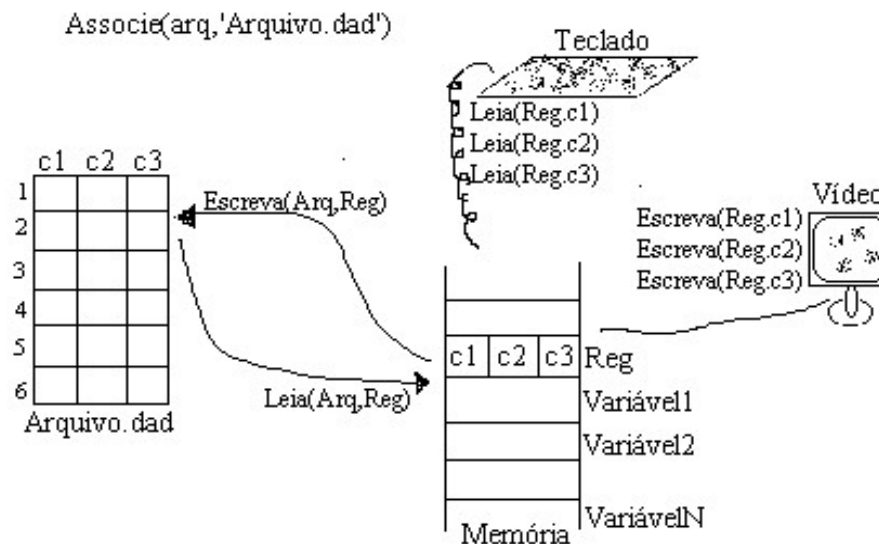
principais operações

- ▣ **Abre**(var_arquivo): comando que abre um arquivo existente e coloca o apontador no primeiro registro.
- ▣ **Fecha**(var_arquivo): comando que fecha um arquivo aberto, tornando-o indisponível para o acesso.
- ▣ **ReAbre**(var_arquivo): comando que: apaga um arquivo existente, cria um arquivo novo e coloca o apontador no início do arquivo.
- ▣ **FDA**(var_arquivo): comando que retorna “Verdadeiro” se o apontador estiver no final do arquivo e “Falso”, caso contrário. Para arquivo vazio, nenhum registro estará sendo apontado.

Arquivo Binário

principais operações

- ▣ **Leia**(var_arquivo, var_registro): comando que carrega o registro corrente do arquivo e o armazena no registro em memória.
- ▣ **Escreva**(var_arquivo, var_registro): comando que grava no registro corrente do arquivo o registro em memória.



Arquivo Binário

principais operações

- ▣ **Aponte**(*var_arquivo*, *número_registro*): comando que move o apontador para o registro de número especificado no arquivo.
- ▣ **TamanhoArquivo**(*var_arquivo*): comando que devolve a quantidade de registros de um arquivo.
- ▣ **PosiçãoRegistro**(*var_arquivo*): comando que retorna o número do registro corrente, ou seja, o qual o apontador estiver apontando.

Arquivo Binário

Exemplo

Lê os dados do teclado e grava-os no arquivo 'produto.dat'.

Algoritmo Exel;

Tipo

Produto = Registro
Nome : Caracter;
Qtidade : Inteiro
Preço : Inteiro;
FimRegistro;

Tipo_Arquivo = Arquivo de **Produto**;

Variáveis

Arq : Tipo_Arquivo;
Reg : **Produto**;
i, n : Inteiro;

Início

Associe(Arq, 'produto.dat');
ReAbre(Arq);
Leia n;
Para i de 1 Até n Faça
 Leia Reg.nome;
 Leia Reg.qtidade;
 Leia Reg.preço;
 Escreva(Arq, Reg);
FimPara;
Fecha(Arq);

Fim.

Arquivo Binário

Exemplo

Lê os registros armazenados no arquivo 'produto.dat' e mostra-os na tela.

Algoritmo Exe2;

Tipo

Produto = Registro

Nome : Caracter;

Qtidade : Inteiro;

Preço : Inteiro;

FimRegistro;

Tipo_Arquivo = Arquivo de Produto;

Variáveis

Arq : Tipo_Arquivo;

Reg : Produto;

Inicio

Associe(Arq, 'produto.dat');

Abre(Arq);

Repita

Leia(Arq, Reg);

Escreva Reg.nome;

Escreva Reg.qtidade;

Escreva Reg.preço;

Até (FDA(Arq));

Fecha(Arq);

Fim.

Arquivo Binário

Exemplo

Atualiza um registro no arquivo
'produto.dat' usando acesso
seqüencial.

Algoritmo Exe3;

Tipo

Produto = Registro

Nome : Caracter;

Qtidade : Inteiro;

Preço : Inteiro;

FimRegistro;

Tipo_Arquivo = Arquivo de Produto;

Variáveis

Arq : Tipo Arquivo;

Reg : Produto;

qt, nomreg : Inteiro;

achou : Lógico;

Inicio

Associe(Arq, 'produto.dat');

Abre(Arq);

Escreva 'Digite o nome do produto a ser alterado a quantidade';

Leia nomreg;

Escreva 'Digite a nova Quantidade em estoque';

Leia qt;

Achou \leftarrow F;

Repita

Leia(Arq, Reg);

Se (Reg.nome = nomreg)

Então | Aponte(Arq, PosiçãoRegistro(Arq) - 1);

| Reg.qtidade \leftarrow qt;

| Escreva(Arq, Reg);

| achou \leftarrow V;

Até (FDA(Arq)) ou (achou);

Fecha(Arq);

Fim.

Arquivo Binário

Exemplo

Atualiza um registro no arquivo 'produto.dat' usando acesso direto.

Algoritmo Exe4;

Tipo

Produto = Registro

Nome : Caracter;

Qtidade : Inteiro;

Preço : Inteiro;

FimRegistro;

Tipo_Arquivo = Arquivo de Produto;

Variáveis

Arq : Tipo Arquivo;

Reg : Produto;

Qt, numreg : Inteiro;

Inicio

Associe(Arq, 'produto.dat');

Abre(Arq);

Escreva 'Digite o número do registro a ser alterado a quantidade';

Leia numreg;

Escreva 'Digite a nova quantidade em estoque';

Leia qt;

Aponte(Arq, numreg);

Leia(Arq, Reg);

Reg.qtidade \leftarrow qt;

Aponte(Arq, PosiçãoRegistro(Arq) - 1);

Escreva(Arq, Reg);

Fecha(Arq);

Fim.

Arquivo Binário

Exemplo

Tipo

vp = Vetor[1..3] de inteiro;

Alunos = Registro

Nome : Caracter;

Notas : vp;

Media : real;

aprovado : Lógico;

FimRegistro;

Tipo_Arquivo = Arquivo de Alunos;

Variáveis

ind : Inteiro;

nota : Inteiro;

nome : Caracter;

arq : Tipo_Arquivo;

reg : Alunos;

A rotina atualiza campos do registro do arquivo a partir de outros campos do registro.

Procedimento Aprov_Reprov;

Inicio

Escreva 'Calculando a media dos alunos';

Abre(Arq);

Enquanto não (FDA(Arq)) Faça

Leia(Arq, Reg);

Reg.media ← Calcula_Media(Reg.notas);

Se (Reg.media ≥ 6.0)

Então Reg.aprovado ← V;

Senão Reg.aprovado ← F;

FimSe

Aponte(Arq, PosiçãoRegistro(Arq) - 1);

Escreva(Arq, Reg);

FimEnquanto;

Fecha(Arq);

Fim;

Exercícios

1. Faça uma rotina que copie os registros de um arquivo para outro arquivo. Declare todas as variáveis necessárias antes.
2. Faça uma rotina que copie os registros iguais de um arquivo para outro arquivo. Declare todas as variáveis necessárias antes.
3. Faça uma rotina que retire os registros repetidos de um arquivo. Declare todas as variáveis necessárias antes.
4. Faça uma rotina que mostre os registros iguais entre dois arquivos. Declare todas as variáveis necessárias antes.

Exercícios

5. Dado uma ficha de uma biblioteca com os seguintes campos: código livro, título, autor, assunto (ex. algoritmo, banco de dados, ...), editora, ano e edição, construa um algoritmo modularizado que:
- a) Permita cadastrar os livros em um arquivo;
 - b) Possibilite mostrar ao usuário todos os livros de um mesmo assunto que ele está procurando;
 - c) O usuário altere um dos campos do registro do arquivo usando “código livro”;
 - d) Crie um novo arquivo com todos os livros ordenados pelo nome do autor;
 - e) Mostre os registros do arquivo criado em ‘d’;
 - f) Mostre um relatório ordenado pela editora de todos os livros de um mesmo assunto fornecido pelo usuário.

Exercícios

6. Dado uma ficha de uma escola com os seguintes campos: código aluno, nome, nota 1, nota 2, nota 3 e nota 4, construa um algoritmo modularizado que:
 - a) Um professor possa executar as principais operações sobre o arquivo. Para a operação de consulta, a média aritmética deverá ser mostrada juntamente com a situação do aluno (≥ 6 aprovado; < 6 e ≥ 3 recuperação; < 3 reprovado);
 - b) Imagine que o professor montou 8 equipes de 5 alunos e precisa da média de cada equipe. A composição de cada equipe (códigos alunos) deverá ser armazenada em uma matriz 8×5 . Para encontrar a média de uma equipe use as notas dos alunos que compõe a equipe aplicando o cálculo da média aritmética.

Exercícios

7. Dado uma ficha de uma escola com os seguintes campos: número matrícula do aluno, rg, nome, data nascimento, sexo e curso, construa um algoritmo modularizado que:

A composição do registro deverá ser:

Tipo

aluno = registro

rg, mat, curso : inteiro; //para curso [1-computação 2-matemática 3-direito 4-pedagogia ...]

nome, dnasc, sexo : caracter; //para sexo ['m' 'f']

fimregistro;

- a) Possa executar as principais operações sobre o arquivo;
- b) Permita mostrar um relatório de todos os alunos organizados pelo nome para um determinado nome de curso consultado pelo usuário.
- c) Para todos os cursos, organizados pelo nome do curso, possa mostrar um relatório com seus respectivos alunos ordenados pelo nome do aluno.

Imagine que exista um outro arquivo com os códigos dos cursos e seus nomes. Assim, este arquivo deverá ser consultado para descobrir o nome de um curso a partir do código do curso.