

 **UNIVERSIDADE DO VALE DO RIO DOS SINOS**
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS (C6/6) – Curso: Informática

LABORATÓRIO II – AULA 01

Disciplina: Linguagem de Programação PASCAL
Professor responsável: Fernando Santos Osório
Semestre: 99/2
Horário: 21 e 41

E-mail: osorio@exatas.unisinos.br
Web: <http://www.inf.unisinos.br/~osorio/lab2.html>
Xerox : Pasta 54 – LAB. II (Xerox do C6/6)

1. ARQUIVO TEXTO:

Os arquivos texto são um tipo de dados pré-definido da linguagem Pascal (Turbo Pascal), sendo necessária a sua declaração na seção VAR de um programa, da seguinte forma:

Var
ArqTxt: TEXT;

A declaração de uma variável do tipo arquivo “TEXT” (*File of strings*) vai nos permitir abrir um arquivo em disco e associá-lo a esta variável, usando o comando **ASSIGN**. Em seguida, para que possamos escrever no arquivo vamos inicialmente abrir ele com o comando **REWRITE**. O programa listado abaixo mostra como escrever uma linha de texto em um arquivo em disco:

```
Program File_Hello;
{
  Este programa cria um arquivo texto em disco - hello.txt
  cujo conteúdo é a mensagem "Hello world!".
}

Var
  ArqTxt:Text; { Define uma variável ArqTxt do tipo arquivo texto=string }

Begin

  assign(ArqTxt,'hello.txt');
  { Associa o nome do arquivo a variável ArqTxt }

  rewrite(ArqTxt);
  { Cria um novo arquivo em disco, vazio e pronto para ser escrito }

  { Usa os comandos Write/WriteLn para escrever no arquivo... }
  writeln(ArqTxt);      { Escreve uma linha em branco no arquivo }
  writeln(ArqTxt,'HELLO WORLD!'); { Escreve o texto no arquivo }
  writeln(ArqTxt);      { Escreve uma linha em branco no arquivo }

  close(ArqTxt);      { Fecha o arquivo. Atualiza informações no disco }

end.
```

Os comandos write/writeln/read/readln podem ser usados para escrever e ler dados de arquivos de maneira muito similar ao modo como usamos estes comandos para escrever e ler dados no vídeo/teclado. A única diferença é que indicamos como primeiro parâmetro da chamada da procedure writeln/readln o nome da variável ao qual o arquivo em disco foi associado. Desta forma os dados são escritos no arquivo como se estivessem sendo escritos na tela, ou, são lidos do arquivo como se alguém estivesse digitando no teclado as informações. Tudo é feito em modo TEXTO.

* Os principais comandos usados para manipular arquivos em Pascal são:

ASSIGN – Associa o nome de um arquivo físico existente em disco ao nome de uma variável do programa usada para se referir ao arquivo em disco. Exemplo: assign(MeuArq,'C:\XPTO.TXT');

REWRITE – Abre o arquivo em disco, zerando o seu conteúdo, e preparando ele para que se possa escrever à partir do início do mesmo. O comando rewrite é usado para criar um arquivo e escrever neste arquivo. Se o arquivo já existir e possuir informações, ao executar o comando rewrite, *todo o seu conteúdo será destruído!* Exemplo: rewrite(MeuArq);

RESET – Abre um arquivo em disco, já existente, preparando-se para ler o seu conteúdo à partir do início. O arquivo deve obrigatoriamente existir no disco, caso contrário ocorrerá um erro de execução. O comando reset é usado para ler um arquivo. Exemplo: reset(MeuArq);

APPEND – Abre um arquivo em disco, já existente, preparando-se para escrever ao final deste. O arquivo deve obrigatoriamente existir no disco. O comando append é usado para adicionar (escrever) novas informações no final do arquivo. Exemplo: append(MeuArq);

CLOSE – Fecha o arquivo, atualizando as informações em disco. O comando close permite que este arquivo seja atualizado, esvaziando buffers, ajustando tamanho final, etc, e disponibilizando (liberando) o arquivo para outros usos. Exemplo: close(MeuArq);

* Para escrever e ler nos arquivos tipo texto:

WRITELN – Escreve uma linha de texto no arquivo. Similar ao comando writeln para o vídeo.

READLN – Lê uma linha de texto de um arquivo. Similar ao comando readln para o teclado.

Exemplos:

```
writeln(MeuArq,'Idade = ', Reg_Funcionario.Idade);  
writeln(MeuArq,Nota1,' ',Nota2,' ',Media);  
readln(ArqFunc,Nro_Func,Salario,Reajuste);  
readln(ArqTexto,Linha);
```

* Para que se possa saber a posição em que nos encontramos em um arquivo, ou para que possamos nos posicionar, temos os seguintes comandos:

EOF – Função que indica se chegamos ao fim do arquivo (End-Of-File).

SEEK – Procedure que permite avançar até uma posição específica do arquivo.

FILEPOS – Função que retorna a posição atual em que estamos lendo o arquivo.

FLUSH – Permite “esvaziar” o buffer de leitura/escrita. Usada para ter certeza que o que está sendo lido ou escrito realmente está no arquivo e não em um buffer de memória.

* Para manipular com diretórios e com arquivos de modo geral:

ERASE – Apaga um arquivo existente no disco.

RENAME – Renomeia um arquivo existente em disco.

CHDIR – Troca de diretório (Change Dir).

MKDIR – Cria um novo diretório (Make Dir).

RMDIR – Apaga um diretório do disco (Remove Dir).

GETDIR – Retorna qual é o diretório corrente de trabalho.

O Turbo Pascal permite também o uso de uma função de retorno de erros relativos ao acesso a arquivos através de uma variável chamada *IOResult*. Para os arquivos texto, é possível que através desta variável possamos saber se um arquivo já existe ou não no disco. Exemplo da implementação de uma função do tipo “existe_arquivo” (vide *help* do comando *reset*):

```
Function Existe_Arquivo(NomeArq: String): Boolean;
{
  Função que retorna um valor boolean indicando se o arquivo existe (true) ou não (false)
}
Var
  Arquivo: Text; {Também aceita => Arquivo: File}
Begin
  {$I-}
  Assign(Arquivo, NomeArq);
  FileMode := 0; { Selecciona o modo de acesso do tipo “read-only” }
  Reset(Arquivo);
  Close(Arquivo);
  {$I+}
  Existe_Arquivo:= (IOResult = 0) and (NomeArq <> "");
end; { Existe_Arquivo }

Begin
  if Existe_Arquivo ('C:\Arq.txt')
  then
    Writeln('O arquivo existe!')
  else
    Writeln('O arquivo não existe!');
End.
```

Observações Importantes:

- Lembre-se que o `ReadLn` e `WriteLn` em arquivos se comportam de modo similar a escrita na tela e leitura do teclado. Sendo assim, quando usamos o `readln`, se lermos uma variável numérica, o espaço em branco serve como separação entre um dado e outro a ser lido, mas se lermos uma variável do tipo `string`, toda a linha será lida até chegar no “enter” do final da linha;
- Cuidado ao usar o comando `read` (no lugar do `readln`) pois ele não lê o “enter”;
- Os comando de posicionamento de cursor na telas, `gotoxy` e `clrscr`, não tem efeito no que se refere aos arquivos. Para escrever uma linha em um arquivo, onde o texto se encontra no meio da linha, a única maneira é escrever vários espaços em branco, deslocando o texto mais para adiante na linha.

EXERCÍCIOS

1. Faça um programa que escreva em disco uma contagem que vá de 1 até 100, com um número em cada linha. Escreva na última linha do arquivo o seu nome.
2. Faça um programa que leia um arquivo texto, lendo linha a linha, e exibindo as linhas numeradas na tela. A idéia é podermos pegar um arquivo de programa Pascal e mostrar na tela com as linhas numeradas.
3. Faça um programa que imite um editor de textos. Inicialmente você irá criar um vetor em memória onde será armazenado o texto escrito pelo usuário (texto de 1 até no máximo 50 linhas). O usuário vai escrever um texto, terminando por uma linha onde ele irá escrever apenas a palavra 'FIM'. Salvar o conteúdo deste vetor que está na memória, em um arquivo em disco.
4. Faça um programa que leia (do teclado) um cadastro de 10 alunos, indicando o nome, nota1, nota2. Calcule a média simples dos 10 alunos e depois escreva um arquivo contendo os dados de cada aluno: nome, nota1, nota2 e média.
5. Faça um programa que leia os dados gerados pelo programa anterior e mostre na tela as informações referentes aos alunos, calculando também a média geral da turma.
6. Faça um programa de criptografia de dados, ou seja, um programa capaz de ler um arquivo texto, codificar este arquivo através de alguma técnica de alteração do código ASCII (exemplo: somar 1 ao valor ASCII de cada caracter), e escrever em disco o arquivo codificado. Crie um outro programa que descriptografe um arquivo criado pelo programa de criptografia, realizando a operação inversa: ler o arquivo do disco, decodificar e escrever o novo arquivo em disco descriptografado.
7. Faça um programa do tipo "grep". Dado um arquivo texto e uma palavra qualquer, este programa deve procurar esta palavra no arquivo texto e devolver a linha onde esta palavra foi encontrada.
8. Faça um programa do tipo "sed ou awk" (Unix). Dado um arquivo texto e duas palavras (original e substituída), substitua todas as ocorrências da palavra original colocando no seu lugar a palavra a ser substituída.