

## Lista de Comandos: Arquivo texto e Binário

---

### Declaração de arquivo do tipo binário

**program** sequencial;

**type**

endereco = **record**

    rua : string[40];  
    numero : integer;  
    CEP : real;

**end;**

cadastro = **record**

    nome : string[30];  
    ender : endereco;  
    CPF: real;  
    Sexo : char;  
    nascimento : real;  
    rg : real;

**end;**

arqcad = **file of** cadastro; { **file of** é uma palavra chave para definir arquivo binário }

**var**

    arq\_in, arq\_out : arqcad;  
    reg: cadastro;

---

### Declaração de arquivo do tipo texto

**Var**

**ArqEnt : text;** { **text** é uma palavra-chave que caracteriza os arquivos tipo texto. }

---

**Texto/Binário**

**assign()** - Associa o nome do arquivo a ser utilizado no programa com o nome do arquivo a ser utilizado no disco. Funciona direcionando o ponteiro para o arquivo, ou melhor, dando um caminho virtual para ele.

Ex: `assign(arquivo, 'texto.txt');`

---

### **Texto/Binário**

**rewrite()** - Este comando se comporta de duas formas:

Quando um arquivo que já existe for aberto com **rewrite**, **seu conteúdo será apagado** e o "pointer" será posicionado no início do arquivo. Quando o arquivo não existir, o comando **rewrite** criará um novo arquivo com o nome especificado na instrução **assign**, deixando-o aberto.

Ex: `rewrite(arquivo);`

---

### **Texto/Binário**

**reset()** - Abre um arquivo já existente e posiciona o "pointer" no início do arquivo. Se o arquivo não existir, ocorrerá um erro de **I/O**. No contexto de arquivo do tipo texto **reset** abre o arquivo exclusivamente para leitura. No caso de arquivo do tipo binário, **reset** abre o arquivo para leitura e/ou gravação.

Ex: `reset(arquivo);`

---

### **Texto**

**append()** - Abre um arquivo do tipo texto já existente exclusivamente para a escrita, prolongando o arquivo a partir do último carácter já gravado.

Ex: `append(arquivo);`

---

### **Texto/Binário**

**close()** - Esvazia o buffer e fecha o arquivo. Faz com que todos os dados contidos no buffier temporário sejam gravados no disco (descarrega o buffer). Além disso, também fecha o arquivo.

Ex: close(arquivo);

---

### **Texto/Binário**

**eof()** 'end of file' (fim de arquivo) Função que devolve TRUE se o final do arquivo for encontrado e FALSE caso contrário.

eof(arquivo);

---

### **Binário**

**seek()** - Com esse comando é possível acessar o n-ésimo registro do arquivo associado à "arquivo", caso n seja menor que o número de registros do arquivo. Em Pascal, o primeiro registro é o de número 0 (zero).

Ex: seek(arquivo, n);

---

### **Binário**

**read** - Esse comando lê o conteúdo do registro corrente do arquivo associado à variável "arquivo" e armazena-o na variável registro, que deve ser do mesmo tipo do arquivo. Após a execução desse comando, o registro corrente é lido, e o próximo passa a ser o registro corrente.

Ex: read(arquivo,registro);

---

## Binário

**write()** - Esse comando, quando executado, grava no registro corrente do arquivo associado à variável “arquivo” o conteúdo da variável registro.

Ex: write(arquivo, registro);

---

## Texto

**Read() e Readln(), Write() e Writeln()** – em relação a arquivos do tipo texto e binário, uma diferença com relação às operações é a possibilidade de utilização dos comandos readln e writeln sobre arquivos texto, o que não é possível sobre arquivos binários, visto que estes não possuem marca de final de linha. A diferença entre os comandos read e readln, write e writeln é basicamente a mesma quando utilizados como comandos de entrada e saída. Os comandos read e write operam sobre campos, enquanto que os comandos readln e writeln operam sobre linhas.

Ex: read(arquivo,lista-de-variaveis); readln(arquivo,lista-de-variaveis); write(arquivo,lista-de-variaveis); writeln(arquivo,lista-de-variaveis)

---

## Binário

**filesize()** - Informa a quantidade de registros do arquivo.

Ex: seek(arquivo, filesize(arquivo)); --> Esta instrução posiciona o "pointer" no final do arquivo “arquivo” para executar inclusão de novos registros. Como o primeiro registro de um arquivo em Pascal possui número zero, a instrução acima posiciona o "pointer" no próximo registro livre para inclusão (após o último ocupado).

---

## Binário

**filepos()**- Informa a posição atual do "pointer" no arquivo. Indica qual é o número do registro corrente.

Ex: filepos(arquivo)

