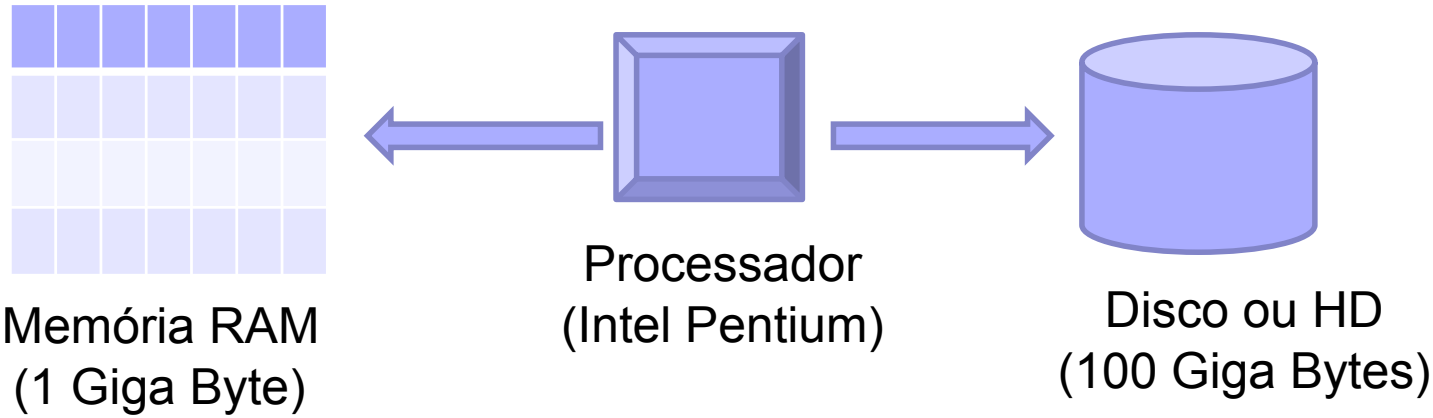
# Computação Eletrônica

Arquivo

# Arquivo




- Por que temos 2 memórias?
  - HD: permanente (pode desligar o computador), barato e lento
    - O HD é representado por um cilindro 
  - RAM: volátil (ao desligar o computador, perde tudo), caro e rápido

# Arquivo

- Até o momento, todos os programas Pascal usavam apenas a memória RAM
- As variáveis são armazenadas na memória RAM

# Arquivo

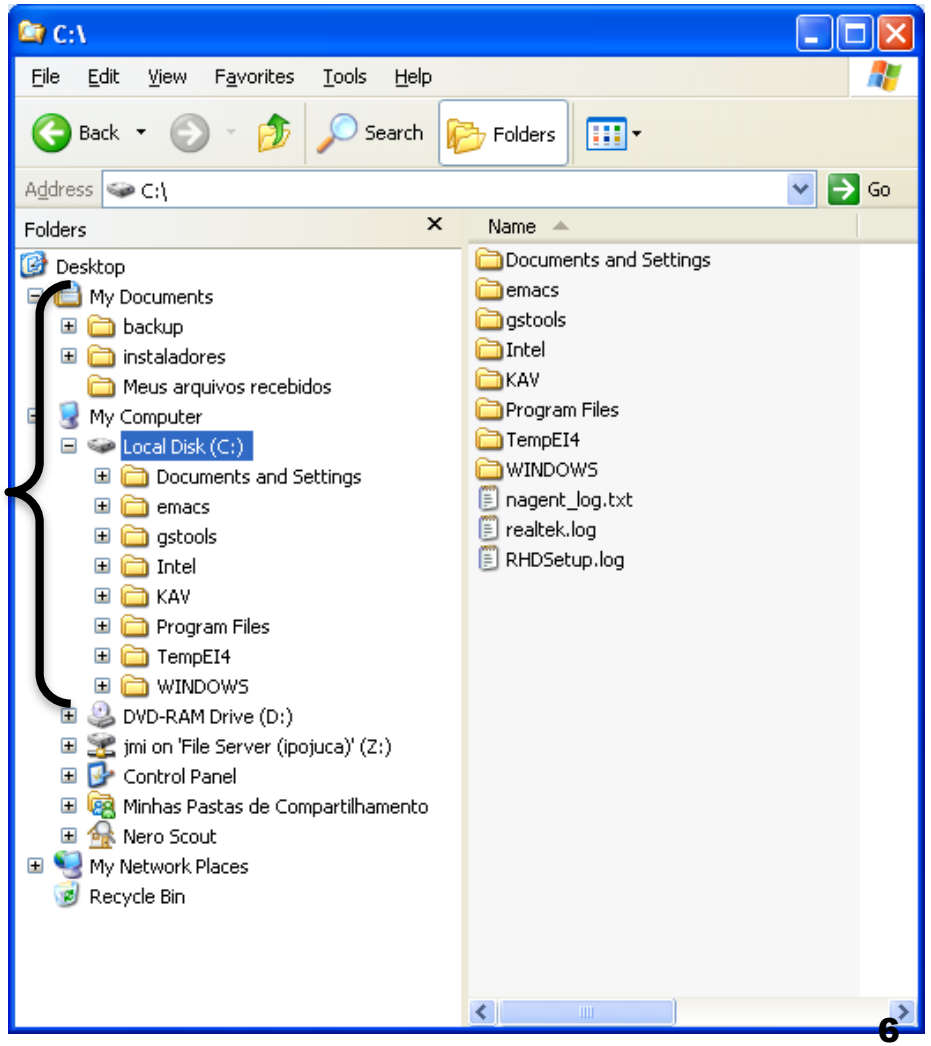
- Suponha um programa Pascal que lê o nome e a nota de 70 alunos armazenando em 2 vetores.
  - Se o computador for desligado no meio da digitação, todos os dados são perdidos
  - Pior: assim que o programa terminar, todos os dados são perdidos também
  - Como fazer para salvar estas informações no disco?
  - Ou seja, como manipular arquivos em Pascal?

# Arquivo

- Não confunda:
  - Arquivos que contém um programa Pascal
    - Este arquivos, por exemplo, helloworld.pas, são compilados (traduzidos) para helloworld.exe. Se você der 2 cliques em helloworld.exe o programa executará. Chamaremos estes arquivos de ***arquivos Pascal***
  - Arquivos *criados* por um programa Pascal
    - São arquivos que contém dados como inteiros, reais, nome de aluno, notas, número de matrícula, sócios de clubes, etc.
    - Não podem ser lidos pelo Windows. Duplo clique neles não resultam em nada. Apenas programas Pascal conseguem ler estes arquivos (acessar suas informações). Chamaremos estes arquivos de ***arquivos Arq***

# Arquivo

- No disco (ou HD) é onde ficam armazenados os arquivos




Todas estas pastas e estes arquivos estão armazenados no disco. Ou seja, você pode desligar o computador sem perder esses dados.

# Arquivo

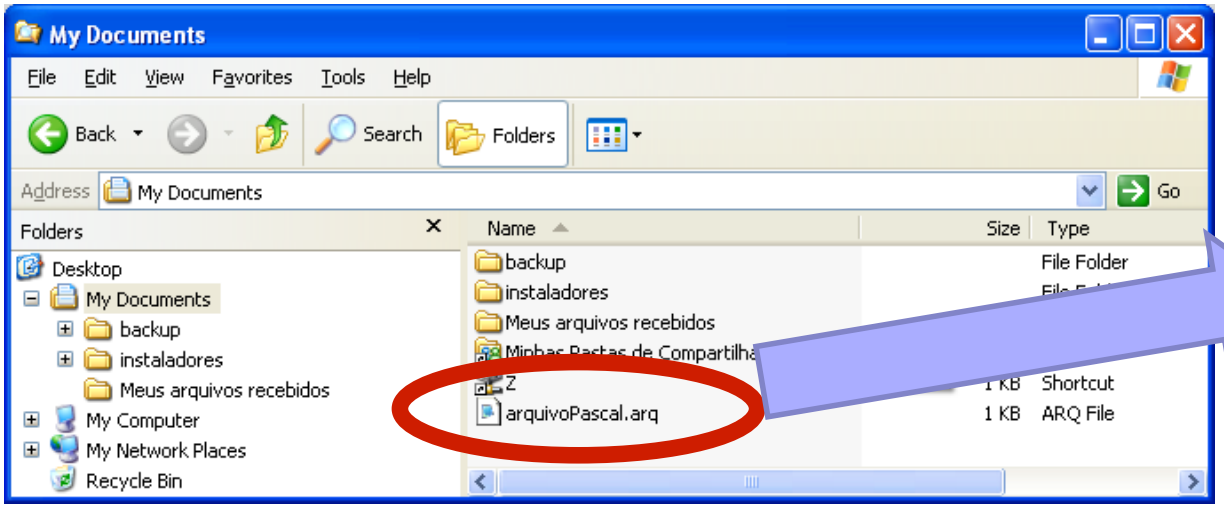
- Pascal consegue criar arquivos
- Todo arquivo *Arq* possui um tipo
  - Por exemplo, arquivo do tipo inteiro, ou arquivo do tipo real, etc.
  - Um arquivo do tipo inteiro, armazena **vários** números inteiros
  - Um arquivo do tipo real, armazena **vários** números reais
  - Um arquivo do tipo aluno armazena **vários** alunos

# Arquivo

- Todo arquivo *Arq* também possui um nome (assim como todo arquivo de qualquer computador)
- Todo arquivo *Arq* possui, no final, um símbolo de fim de arquivo (eof)

# Arquivo

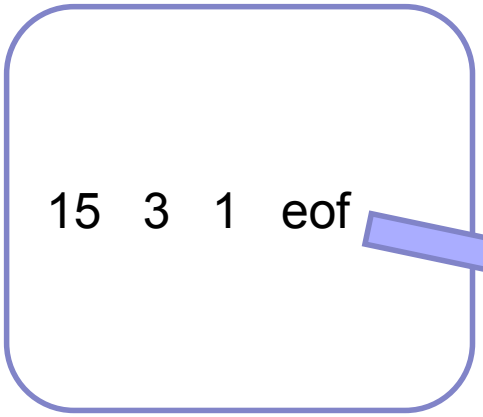
- Exemplo de um arquivo *Arq* cujo nome é `arquivoPascal.arq`
- Este arquivo pode ser salvo em qualquer pasta do seu Windows. No caso abaixo, a pasta é My Documents




Como este arquivo foi parar aí? Calma... Aguarde os próximos slides.

# Arquivo

- O que contém um arquivo de inteiros?
- Se pudéssemos ver seu conteúdo, seria mais ou menos assim:




arquivoPascal.arq

Arquivo de inteiros que contém os números 15, 3 e 1.

**eof**: Símbolo especial que indica o fim do arquivo. Todo arquivo Pascal tem este símbolo no final

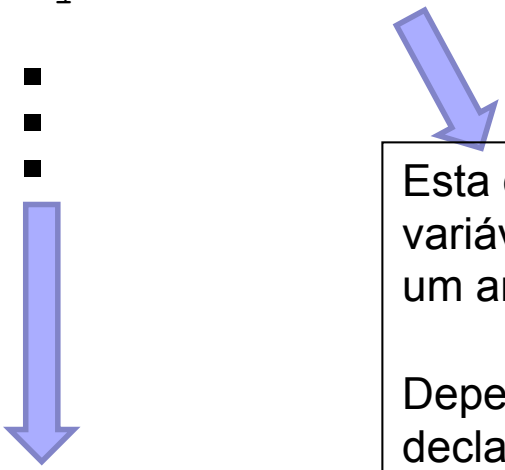
# Arquivo

- Como manipular arquivos *Arq* em Pascal?
  1. Declare uma variável do tipo *file of <tipo>*
     - *Por exemplo, um arquivo de inteiros é uma variável do tipo "file of integer"*
  2. Associe o nome do arquivo no HD à variável declarada
  3. *Prepare* o arquivo para ser usado
  4. Leia ou escreva dados na variável declarada

# Arquivo

## 1. Declare uma variável do tipo *file of <tipo>*

```
Program arquivo1;
var xyz: file of integer;
```



O programa continua nos próximos slides.

Esta declaração diz ao Pascal: a variável **`xyz`** será usada para manipular um arquivo de inteiros.

Dependendo do problema, podemos declarar variáveis do tipo “file of real”, “file of boolean”, “file of String”, “file of socio” (onde “socio” é um registro), etc.

# Arquivo

## 2. Associe o nome do arquivo no HD à variável declarada

```
Program arquivo1;
var xyz: file of integer;
    n: integer;
begin
   Assign(xyz,'arquivoPascal.arq');
```

Este comando apenas diz ao Pascal: a variável **`xyz`** será usada para manipular um arquivo **cujo nome no HD é arquivoPascal.arq.**

Não confunda! O nome da **variável do tipo file** é diferente do **nome do arquivo no HD**. Pascal precisa saber como estes nomes se relacionam.

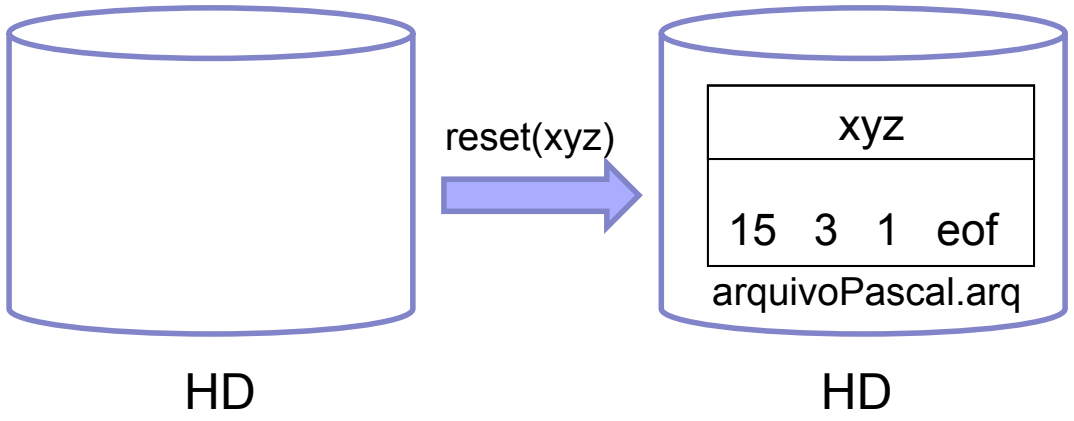
# Arquivo

Este slide assume que o arquivo `arquivoPascal.arq` está no HD

## 3. Prepare o arquivo para ser usado

```
Program arquivo1;
var xyz: file of integer;
    n: integer;
begin
   Assign(xyz,'arquivoPascal.arq');
   Reset(xyz);
   ▪
   ▪
   ▪
```

`reset(xyz)` prepara o arquivo `arquivoPascal.arq` para ser utilizado. Pascal localiza o arquivo no HD e deixa seu conteúdo disponível para manipularmos. Dizemos que `reset` **abre** o arquivo.

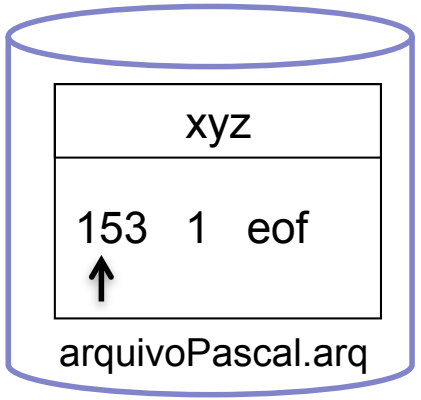


# Arquivo

Este slide assume que o arquivo `arquivoPascal.arq` está no HD

## 4. Leia ou escreva dados na variável declarada

```
Program arquivo1;
var xyz: file of integer;
    n: integer;
begin
   Assign(xyz,'arquivoPascal.arq');
   Reset(xyz);
```




HD

Um arquivo é uma **lista** de elementos. No nosso exemplo, um arquivo é uma lista de inteiros.
Para manipular arquivos, Pascal automaticamente define a **posição** nessa lista em que o arquivo se encontra. Após o `reset`, a posição é a zero. No exemplo, a posição zero contém o número `15`.

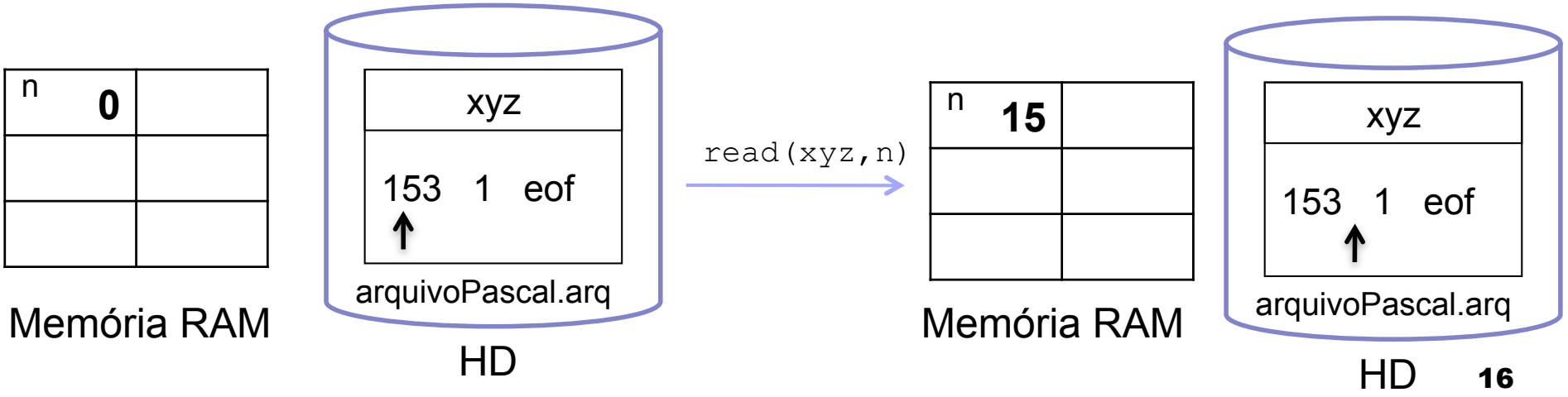
# Arquivo

Este slide assume que o arquivo `arquivoPascal.arq` está no HD

## 4. Leia ou escreva dados na variável declarada

```
Program arquivo1;
var xyz: file of integer;
    n: integer;
begin
   Assign(xyz,'arquivoPascal.arq');
   Reset(xyz);
   read(xyz,n);
```

`read(xyz,n)` lê o número na posição corrente de `xyz` (no caso, `15`) e copia este valor para a variável `n`. Pascal automaticamente adianta a posição do arquivo.



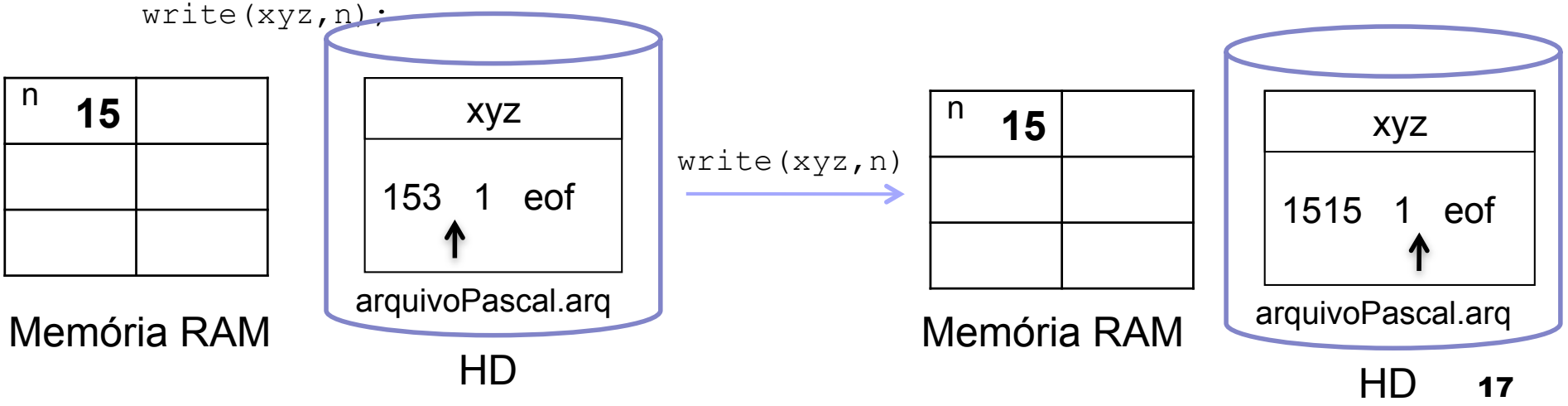
# Arquivo

Este slide assume que o arquivo `arquivoPascal.arq` está no HD

## 4. Leia ou escreva dados na variável declarada

```
Program arquivo1;
var xyz: file of integer;
    n: integer;
begin
   Assign(xyz,'arquivoPascal.arq');
   Reset(xyz);
   read(xyz,n);
   write(xyz,n);
```

`write(xyz,n)` grava o valor de `n` na posição corrente de `xyz`. Pascal automaticamente adianta a posição do arquivo.



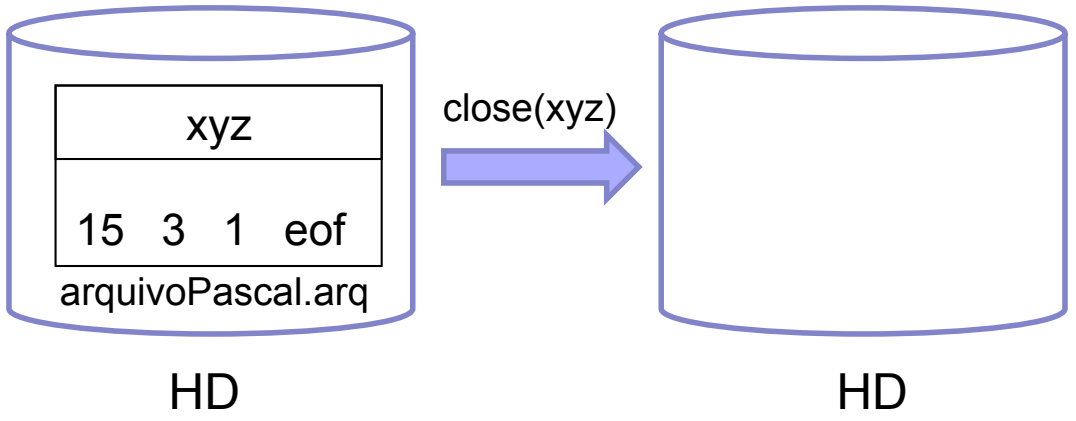
# Arquivo

Este slide assume que o arquivo `arquivoPascal.arq` está no HD

Ao final do programa, **feche os arquivos.**

```
Program arquivo1;
var xyz: file of integer;
    n: integer;
begin
   Assign(xyz,'arquivoPascal.arq');
   Reset(xyz);
   read(xyz,n);
   write(xyz,n);
   close(xyz);
end.
```

`close(xyz)` diz a Pascal que o arquivo não será mais manipulado.

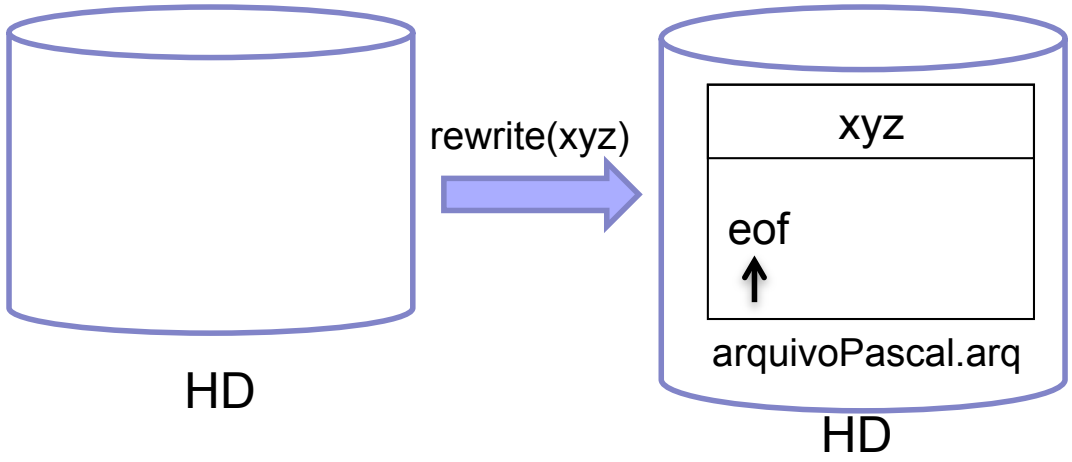


# Arquivo

## E se quisermos criar um novo arquivo?

```
var xyz: file of integer;
begin
   Assign(xyz,'arquivoPascal.arq');
   Rewrite(xyz);
```

- ...
- ...
- ...



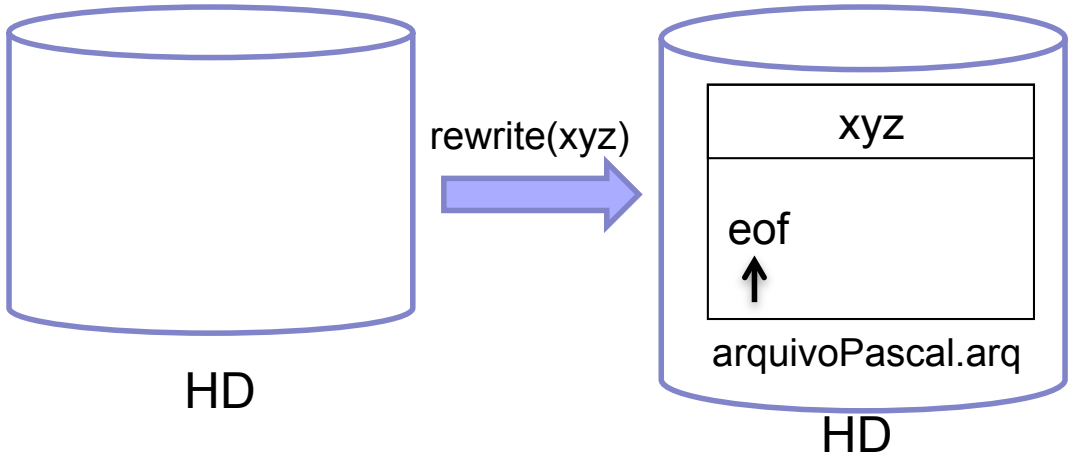

`Rewrite(xyz)` cria no HD um arquivo **vazio (só com o eof)** cujo nome é `arquivoPascal.arq`. A posição do arquivo é a zero, que aponta para eof.

# Arquivo

## E se quisermos criar um novo arquivo?

```
var xyz: file of integer;
begin
   Assign(xyz,'arquivoPascal.arq');
   Rewrite(xyz);
```

- ...
- ...
- ...

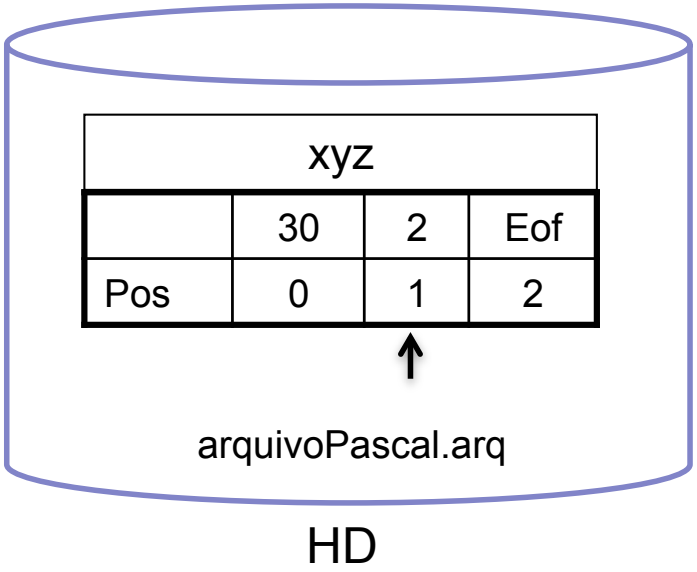



`Rewrite(xyz)` cria no HD um arquivo **vazio** (já com o eof) cujo A a a

**CUIDADO:** Se um arquivo chamado `arquivoPascal.arq` já existir no HD, ele será apagado e substituído pelo novo `arquivoPascal.arq` vazio.

# Arquivo

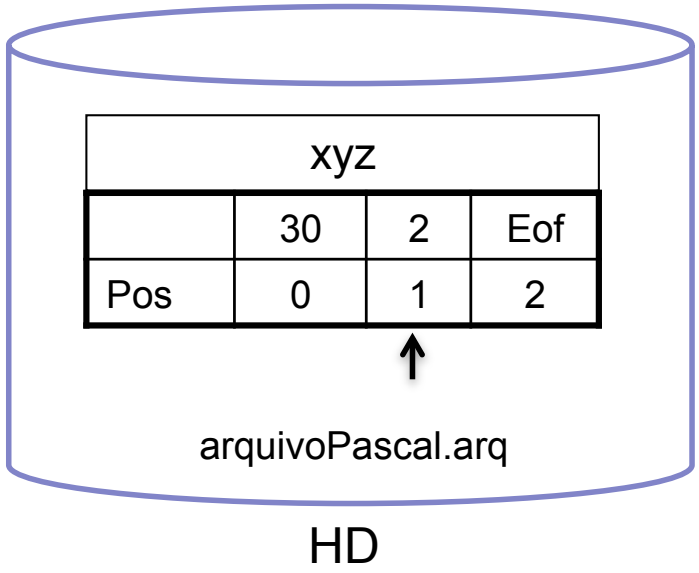
A posição de um arquivo é um número que vai de 0 (início) à N (eof).



# Arquivo

A função `filepos(arquivo)` retorna a posição corrente do arquivo (sem alterá-la).

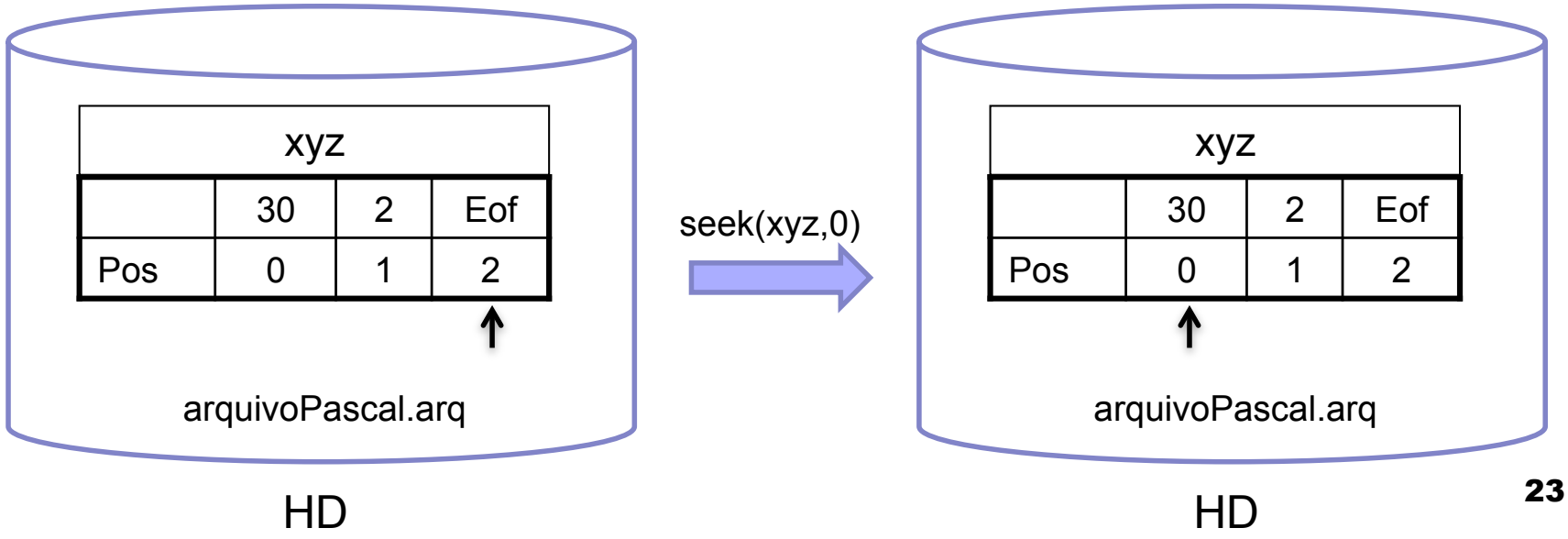
No exemplo abaixo, `filepos(xyz)` retorna o número 1.



# Arquivo

A função `seek(arquivo,n)` posiciona o arquivo na posição n.

O exemplo abaixo `seek(xyz,0)` leva o arquivo para a posição zero.



# Arquivo

Devemos ter cuidado ao atualizar um arquivo. Atualizar significa ler um dado do arquivo, modificá-lo e gravar a modificação. Como `read` automaticamente avança a posição corrente do arquivo, temos que sempre voltar uma posição antes de usar `write`.

# Arquivo

## Programa que lê o primeiro número de um arquivo e atualiza-o com seu valor mais 5.

```
Program atualiza;
var arquivo: file of integer;
    n, pos: Integer;
begin
   Assign(arquivo, 'arquivoDeInteiros.dad');
   Reset(arquivo);
   Read(arquivo, n); {n recebe o primeiro numero do arquivo.}
   {Porem, read avancou a posicao do arquivo. Temos que voltar 1 posicao}
   pos := filepos(arquivo); {guarda em pos a posicao atual do arquivo}
   pos := pos - 1; {calcula a posicao anterior}
   seek(arquivo,pos); {posiciona o arquivo na posicao anterior}
   n := n+5; {calcula o novo valor de n}
   write(arquivo,n); {atualiza n}
   close(arquivo);
end.
```

# Arquivo

```
{Criar um arquivo de inteiros.}
Program arquivo1;
var arquivo: file of integer;
begin
   Assign(arquivo,'meuPrimeiroArquivo.arq');
   Rewrite(arquivo);
   Close(arquivo);
end.
```

# Arquivo

```
{Cria e grava um arquivo de inteiros.}
Program arquivo2;
var arquivo: file of integer;
    i,n: integer;
begin
   Assign(arquivo,'quadrados.arq');
   Rewrite(arquivo);
   i := 1;
   while (i <= 50) do
      begin
          n := i * i;
          write(arquivo,n);
          i := i + 1;
      end;
   close(arquivo);
end.
```

# Arquivo

```
{Lê um arquivo de inteiros.}
Program arquivo3;
var arquivo: file of integer;
    i,n: integer;
begin
   Assign(arquivo,'quadrados.arq');
   Reset(arquivo);
   while not(eof(arquivo)) do
      begin
          read(arquivo,n);
          writeln(n);
      end;
   readln;
   Close(arquivo);
end.
```

# Arquivo

```
{Atualiza um arquivo de inteiros.}
Program arquivo3_5;
var arquivo: file of integer;
    i,n,dobro: integer;
begin
   Assign(arquivo,'quadrados.arq');
   Reset(arquivo);
   while not(Eof(arquivo)) do
      begin
          read(arquivo,n);
          dobro := 2 * n;
           seek(arquivo,filepos(arquivo)-1);
          write(arquivo,dobro);
      end;
   Close(arquivo);
end.
```

# Arquivo

```
{Cria um arquivo com registros}
{
Program arquivo4;
type aluno = record mat: integer;
                    cpf: integer;
             end;
var arquivo: file of aluno;
    a: aluno;
    i: integer;
begin
   Assign(arquivo,'alunos.arq');
   Rewrite(arquivo);
   i := 1;
   while (i <= 5) do
      begin
          write('Entre matricula e cpf: ');
          readln(a.mat,a.cpf);
          write(arquivo,a);
      end;
   close(arquivo);
end.
```

# Arquivo

```
{Consulta um arquivo pela mat,
   retornando a mat e cpf}
Program arquivo6;

type aluno = record mat: integer;
                    cpf: integer;
             end;
var arquivo: file of aluno;
    a: aluno;
    matricula: integer;
begin
   Assign(arquivo,'alunos.arq');
   Reset(arquivo);

   write('Entre a matricula: ');
   readln(matricula);

   while not(eof(arquivo)) do
      begin
         read(arquivo,a);
         if (a.mat = matricula) then
            begin
               writeln('Matricula:
',a.mat,'  CPF: ',a.cpf);
               writeln('Sua posicao no
arquivo eh: ',filepos(arquivo)-1);
            end;
      end;
   close(arquivo);
end.
```

# Arquivo

```
{
arquivos e arrays
}
Program arquivo_array;

var arquivo: file of integer;
    vetor: array[0..50] of integer;
    n,i,pos:integer;
begin
    Assign(arquivo,'arquivo.arq');
    Rewrite(arquivo);          { i := 1; }

    write(arquivo,5);          { vetor[i] := 5;  i := i + 1; }
    write(arquivo,8);          { vetor[i] := 8;  i := i + 1; }
    write(arquivo,10);         { vetor[i] := 10; i := i + 1; }

    Reset(arquivo);            { i := 0; }

    read(arquivo,n);           { n := vetor[i]; i := i + 1; }
    read(arquivo,n);           { n := vetor[i]; i := i + 1; }
    read(arquivo,n);           { n := vetor[i]; i := i + 1; }

    pos := filepos(arquivo); { pos := i; }
    seek(arquivo,3);           { i := 3; }

    read(arquivo,n);           { n := vetor[i]; i := i + 1; }
    pos := filepos(arquivo); { pos := i; }
    n := n + 1;                { n := n + 1; }
    write(arquivo,n);          { vetor[i] := n; i := i + 1; }

    close(arquivo);
end.
```