

# **Fichiers**

**NF01**

**Philippe TRIGANO**



# Problème

- **Pour l'instant,**
  - impossible de conserver les données et résultats
  - après exécution, tout est perdu
  
- **Solution : les fichiers**

# Principe général d'accès aux fichiers

1. Association entre le nom externe du fichier et une variable de type fichier. Ex :  

```
assign(f , 'c:\Mes Documents\mon_fichier');
```
2. Ouverture du fichier en « mise à jour » ou en « création »
3. Instruction(s) de lecture ou d'écriture (ou de positionnement)
4. Fermeture du fichier

# Ouverture d'un fichier

## ➤ En mise à jour :

reset (ID\_Fichier);

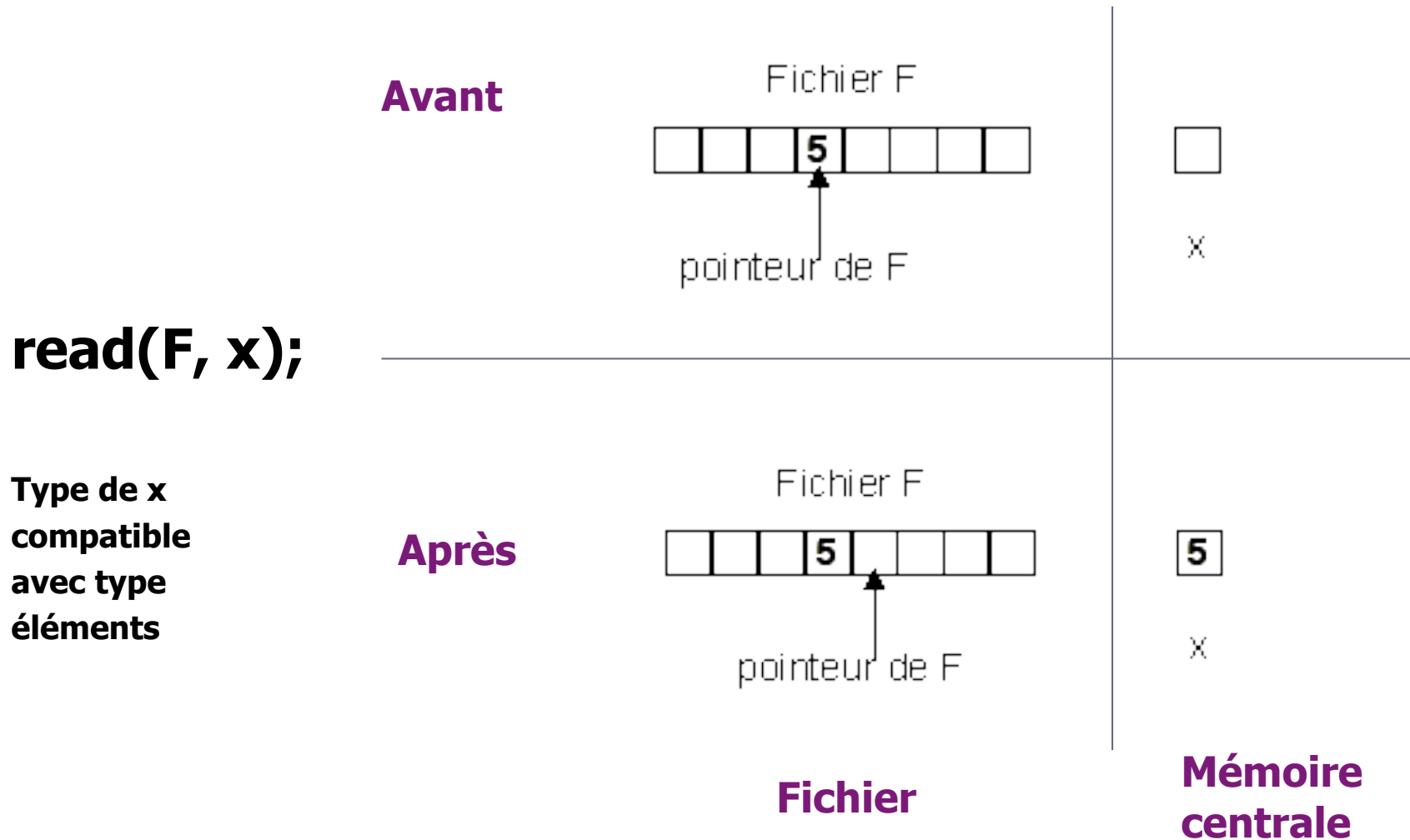
- ✓ Instruction exécutée une fois
- ✓ Après, on peut réaliser des opérations de lecture (ou d'écriture sous certaines conditions)
- ✓ Le pointeur de lecture est positionné au début du fichier

## ➤ En création :

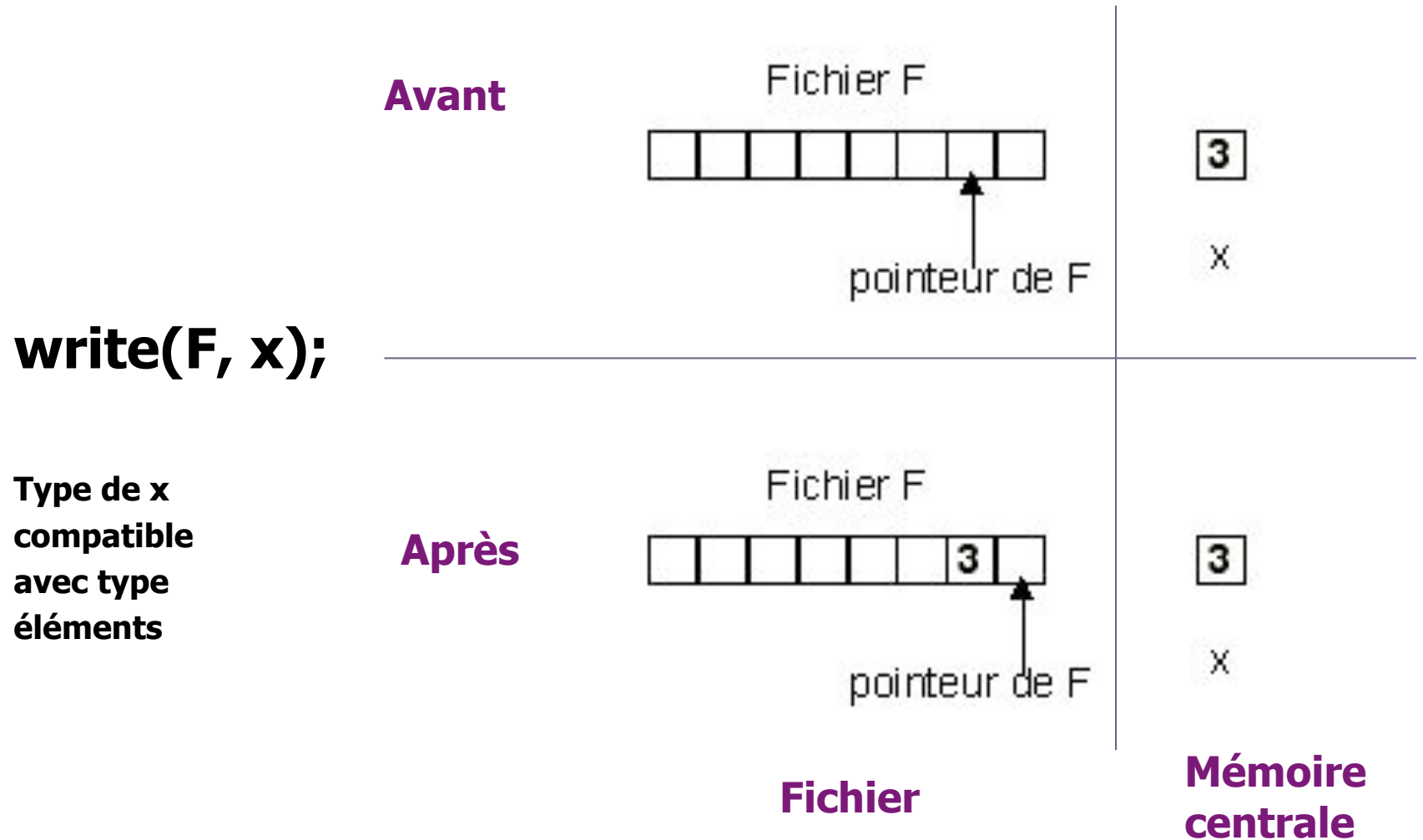
rewrite(ID\_Fichier);

- ✓ Instruction exécutée une fois
- ✓ Après, on peut réaliser des opérations d'écriture
- ✓ Si le fichier existe déjà, son contenu est « écrasé »
- ✓ Si le fichier n'existe pas, il est créé

# Lecture d'un élément du fichier



# Ecriture dans un fichier



# Code Pascal

## ➤ Lecture

```
assign(f,'Mon_Fichier');  
reset(f);  
...  
read(f, v1,v2);  
...  
read(f, v3);  
...  
close(f)
```

## ➤ Ecriture

```
assign(f,'Mon_Fichier');  
rewrite(f);  
...  
write(f, v1);  
...  
write(f, v2, v3);  
...  
close(f)
```



# Fin de fichier

## ➤ Problème

Comment savoir qu'on a atteint la fin du fichier (en lecture) ?

## ➤ Solution

La fonction **eof** (end-of-file) retourne

- ✓ true si la fin de fichier est atteinte
- ✓ false sinon

## ➤ Utilisation

```
while not eof(f) do  
    read(f, ...);  
    ...  
end
```





# Exemple

## ➤ *Affichage du contenu d'un carnet d'adresses*

type

```
Adresse = record  
    nom, rue, ville : string;  
    numero : integer;  
end;
```

```
Fichier = file of Adresse ;
```

var

```
carnet : Fichier;  
client : Adresse;
```



# Lecture dans un fichier

```
begin
  assign(carnet, 'carnet_adresses') ;
  reset(carnet) ;

  while not eof (carnet) do
    begin
      read(carnet, client);

      with client do
        begin
          writeln ('NOM :', nom) ;
          writeln ('NUMERO :', numero) ;
          writeln ('RUE :', rue) ;
          writeln ('VILLE :', ville) ;
        end;
      end ;

      close (carnet) ;
    end.
  end.
```

**Lecture d'un enregistrement  
du fichier *carnet* dans le  
record *client***

# Ecriture dans un fichier

```
begin
  assign (carnet, 'carnet_adresses') ;
  rewrite (carnet) ;
  c := 'O' ;
  while c <> 'N' do
    with client do
      begin
        write ('NOM :') ; readln (nom) ;
        write ('NUMERO :') ; readln (numero) ;
        write ('RUE :') ; readln (rue) ;
        write ('VILLE :') ; readln (ville) ;
        write(carnet,client) ;
        write ('Autre adresse ?') ; readln (c) ;
      end ;
    end ;
  close (carnet) ;
end.
```

**Ecriture du record *client*  
dans un enregistrement  
du fichier *carnet***

# Fichiers Texte

## ➤ Texte

- Suite de caractères
- Ensemble de lignes

## ➤ Fin de ligne

- Un ou deux caractères spéciaux
- Windows : 0D 0A

## ➤ En Pascal

- Type prédéfini : **Text**
- Utilisation possible de **readln** et **writeln**



# Exemple

- **Saisie d'un texte par un utilisateur et écriture dans un fichier**
- **Algorithme ?**

Lire une ligne

Tant que ligne non vide

    Ecrire la ligne dans le fichier

    Lire ligne suivante

# Ecriture dans un fichier texte

```
program MON_TEXTE;  
var  
    f: Text;  
    s: string;  
begin  
    assign(F, 'montexte.txt' );  
    rewrite(f);  
    writeln(' Tapez un texte et terminez par une ligne vide' );  
    readln(s) ;  
    while (s <> "") do  
        begin  
            writeln(f,s);  
            readln(s);  
        end;  
    close(f);  
end.
```

# Lecture d'un fichier texte

## Algorithme

Tant que la fin de fichier n'est pas atteinte

- ✓ Lire une ligne
- ✓ Afficher la ligne

# Lecture dans un fichier texte

```
program Lecture;  
var  
    f: Text;  
    s: string;  
begin  
    assign(f, 'montexte.txt' );  
    reset(f);  
    while not eof(f) do  
        begin  
            readln(f,s);  
            writeln(s);  
        end;  
    close(f);  
end.
```