

Fichiers

Dr Khadim DRAME
kdrame@univ-zig.sn

Département d'Informatique
UFR des Sciences et Technologies
Université Assane Seck de Ziguinchor

20 novembre 2021



Plan

- 1 Introduction
- 2 Organisation et accès aux fichiers
- 3 Manipulation de fichiers



Introduction

- Données utilisées dans un programme sont jusqu'à maintenant
 - soit incluses dans le programme.
 - soit saisies par l'utilisateur.
- Ces données sont stockées dans la mémoire vive (RAM) de l'ordinateur.
- Ces données sont perdues à la fin du programme ou lorsque l'ordinateur est éteint.
- Besoin de conserver certaines données → utiliser les fichiers.



C'est quoi un fichier ?

- Un **fichier** est une structure de données destinée à contenir des données homogènes.
- Il permet de stocker des données sur un support non volatile (disque dur, clé USB, etc.) pour une durée indéterminée.
- Un fichier possède
 - un identificateur (nom du fichier) ;
 - un emplacement ;
 - un contenu (des données).



Types de fichiers

- On distingue trois types de fichiers en Pascal
 - fichiers **texte** : le contenu est une suite de **caractères** regroupés en lignes ;
 - fichiers **typés** : les données sont de type simple (entier, réel) ou structuré (**enregistrement**) ;
 - fichiers **non typés** : le contenu n'est pas spécifié (son, image,...).
- Les fichiers typés sont les plus communément utilisés.



Plan

- 1 Introduction
- 2 Organisation et accès aux fichiers
- 3 Manipulation de fichiers



Organisation des données dans un fichier

- L'accès à un fichier dépend de l'organisation des données dans le fichier.
- Cette organisation est spécifiée au **moment de la création** du fichier.
- On distingue 3 formes d'organisation
 - organisation **séquentielle** : les données sont stockés dans l'ordre dans lequel elles sont entrées ;
 - organisation **relative** : le fichier est composé d'enregistrements numérotés ;
 - organisation **indexée** : chaque enregistrement est associé à une clé.



Méthodes d'accès aux fichiers

- La manière de lire ou écrire des données dans un fichier est dite **méthode d'accès**.
- Cette méthode doit être spécifiée au **moment de l'ouverture** du fichier.
- On distingue 3 méthodes d'accès
 - accès **séquentiel** : les données sont traitées en séquence ;
 - accès **direct** : on peut accéder directement à un enregistrement via son numéro ;
 - accès **indexé** : on accède aux données suivant l'ordre de la clé.
- NB : la méthode d'accès séquentielle est applicable quelque soit l'organisation.



Plan

- 1 Introduction
- 2 Organisation et accès aux fichiers
- 3 Manipulation de fichiers**



Déclaration de variables de type fichier

Syntaxe

- Fichier texte

var <nom_fichier> : **text** ;

- Fichier non typé

var <nom_fichier> : **file** ;

- Fichier typé

var <nom_fichier> : **file** of <type> ;

- Exemple

```
1 var ftext : text ;  
2   fpers : file of personne ;  
3   fblock : file ;
```



Assignment

- L'assignation permet d'associer la variable fichier au chemin du fichier.
- Elle se fait après la déclaration de la variable fichier.
- **assign** : associer le nom logique au nom physique du fichier.
 - Syntaxe
`assign(<nom_fichier>, <nom_physique>);`
 - Exemple
`assign(fpers, 'C: \Documents\doc1.dat');`



Ouverture de fichier

- **reset** : **ouvrir** un fichier en **lecture** sans écraser son contenu.
 - Syntaxe
`reset(<nom_fichier>);`
 - Exemple
`reset(fpers);`
- **rewrite** : **ouvrir** un fichier en **écriture**; si le fichier existe, il sera écrasé et son contenu perdu.
 - Syntaxe
`rewrite(<nom_fichier>);`
 - Exemple
`rewrite(fpers);`



- **read** : **lire** le contenu du composant courant du fichier dans une variable de même type.
 - Syntaxe
`read(<nom_fichier>, <nom_variable>);`
 - Exemple
`read(ftext, ligne);`
- **write** : **écrire** à la position courante du fichier le contenu d'une variable.
 - Syntaxe
`write(<nom_fichier>, <nom_variable>);`
 - Exemple
`ligne := 'Bienvenue à l'UASZ';`
`write(ftext, ligne);`



Vérification de la fin et fermeture

- **eof** : fonction booléenne pour **vérifier** qu'il n'y a plus de données à lire (fin du fichier).
 - Syntaxe
`eof(<nom_fichier>);`
 - Exemple
`eof(fpers);`
- **close** : **fermer** le fichier.
 - Syntaxe
`close(<nom_fichier>);`
 - Exemple
`close(fpers);`



Autres opérations sur les fichiers

- **filepos**(<nom_fichier>) : donne la position courante dans le fichier.
- **filesize**(<nom_fichier>) : détermine la taille du fichier en enregistrements.
- **erase**(<nom_fichier>) : efface le fichier en paramètre.
- **rename**(<nom_fichier>, <nouveau_nom>) : renomme le fichier.
- **seek**(<nom_fichier>, <position>) : positionne le pointeur sur l'enregistrement correspondant à la position spécifiée.
- **ioresult**(<nom_fichier>) : fonction entière renvoyant l'état de la dernière opération d'E/S (0 opération s'est bien déroulée).



Exemples d'utilisation de fichier typé

Création d'un fichier de nombres réels

```
1 program creerFichier;  
2 Type fichierReel = file of real;  
3 var fs: fichierReel;  
4     X : real;  
5 begin  
6     assign(fs, 'C:\Users\HP\Documents\reels.dat');  
7     rewrite(fs); {ouverture du fichier en mode écriture}  
8     repeat  
9         write('Donner un réel non nul : ');  
10        readln(X);  
11        write(fs, X);  
12    until X=0;  
13    close(fs);  
14 end.
```



Exemples d'utilisation de fichier typé

Lecture d'un fichier de réels et calcul de leur somme

```
1 program lireFichier;
2 Type fichierReel = file of real;
3 var fs: fichierReel;
4     X, S: real;
5 begin
6     assign(fs, 'C:\Users\HP\Documents\reels.dat');
7     reset(fs); {ouverture du fichier en mode lecture}
8     S:=0;
9     while not eof(fs) do
10    begin
11        read(fs, X);
12        S:= S + X;
13    end;
14    writeln(S); (* somme des nombres lus *)
15    close(fs);
16 end.
```

Exemples d'utilisation de fichier typé

Copie d'un fichier dans un autre

```
1 program copierFichier;
2 Type fichierReel = file of real;
3 var fsource, fcopie: fichierReel;
4     X: real; {ou type enregistrement dans le fichier}
5 begin
6     assign(fsource, 'C:\Users\HP\Documents\reels.dat');
7     assign(fcopie, 'C:\Users\HP\Documents\copie.dat');
8     reset(fsource);
9     rewrite(fcopie);
10    while not eof(fsource) do
11    begin
12        read(fsource, X);
13        write(fcopie, X);
14    end;
15    close(fsource);
16    close(fcopie);
17 end.
```



Manipulation de fichiers texte

- Les opérations vues précédemment sont valables pour les fichiers texte.
- D'autres sont spécifiques aux fichiers texte
 - **eoln**(<nom_fichier>) : indique si on est en fin de ligne.
 - **read**(<nom_fichier>, <nom_variable>) : lit une chaîne dans le fichier.
 - **readln**(<nom_fichier>, <nom_variable>) : lit une chaîne dans le fichier et passe à la ligne suivante.
 - **write**(<nom_fichier>, <nom_variable>) : écrit une chaîne dans le fichier.
 - **writeln**(<nom_fichier>, <nom_variable>) : écrit une chaîne dans le fichier et passe à la ligne suivante.



Exemple d'utilisation de fichier texte

Copie d'un fichier texte dans un autre

```
1 program copierFichierTexte;  
2 var fsource, fcopie: text;  
3     ligne: string;  
4 begin  
5     assign(fsource, 'C:\Users\HP\Documents\doc1.txt');  
6     assign(fcopie, 'C:\Users\HP\Documents\copie.txt');  
7     reset(fsource); (* ouverture en mode lecture *)  
8     rewrite(fcopie); (* ouverture en mode écriture *)  
9     while not eof(fsource) do  
10        begin  
11            readln(fsource, ligne);  
12            writeln(fcopie, ligne);  
13        end;  
14        close(fsource);  
15        close(fcopie);  
16 end.
```

Exercices d'application

- 1 Écrire un programme qui crée un fichier typé contenant les informations des apprenants d'une formation professionnelle. Chaque apprenant possède un numéro, un nom, un prénom et une moyenne.
- 2 Écrire un programme qui parcourt le fichier créé précédemment et affiche toutes les informations qu'il contient.
- 3 Écrire un programme qui parcourt le fichier créé précédemment et affiche les informations d'un apprenant dont le numéro est lu à partir du clavier.



Exercices d'application

Correction Exercice 1

```
1 program creerFichierApprenant;  
2 {partie déclarative}  
3 Type apprenant = Record  
4   num : integer;  
5   prenoms : string[30];  
6   nom : string[20];  
7   moy : real;  
8 end;  
9 var f: file of apprenant;  
10    app: apprenant;  
11    i, nb: integer;
```



Exercices d'application

Correction Exercice 1

```
1 begin {Corps du programme}
2     writeln('Donner le nombre d''apprenants : ');
3     readln(nb);
4     assign(f, 'C:\Users\HP\Documents\apprenats.dat');
5     rewrite(f); (* ouverture en mode écriture *)
6     for i:=1 to nb do
7         begin
8             write('Donner le numéro de l''apprenant : ');
9             readln(app.num);
10            write('Donner ses nom et prénoms : ');
11            readln(app.nom, app.prenoms);
12            write('Donner la moyenne de l''apprenant : ');
13            readln(app.moy);
14            write(f, app);
15        end;
16    close(f);
17 end.
```



Exercices d'application

Correction Exercice 2

```
1 program lireFichierApprenant;  
2 Type apprenant = Record  
3   num : integer; prenom : string[30];  
4   nom : string[20]; moy : real;  
5 end;  
6 var f: file of apprenant;  
7     app: apprenant;  
8 begin  
9   assign(f, 'C:\Users\HP\Documents\apprenats.dat');  
10  reset(f); (* ouverture en mode lecture *)  
11  while not eof(f) do  
12    begin  
13      read(f, app);  
14      writeln(app.prenom, ' ', app.nom, ' ', app.moy);  
15    end;  
16    close(f); readln;  
17 end.
```



Insertion, suppression et tri dans un fichier

- Utilisation de tableaux
 - charger les données dans un tableau (ou plusieurs) ;
 - effectuer les opérations sur le tableau ;
 - vider le fichier original ;
 - sauvegarder les données à jour dans le fichier original.
- Utilisation de fichiers temporaires
 - lire les données à partir du fichier ;
 - copier les données modifiées dans un autre fichier temporaire ;
 - supprimer le fichier original ;
 - renommer le fichier temporaire par le nom du fichier original.



- Les fichiers permettent de sauvegarder des données de **manière permanente**.
- En Pascal, il y a trois types de fichiers : **fichiers typés**, **fichiers texte** et **fichiers non typés**.
- Il y a plusieurs **méthodes d'accès** aux données d'un fichier (séquentielle, directe, indexée).
- Un ensemble de **fonctions et procédures** est défini pour **manipuler** les fichiers.

