

Cours d'algorithmique

Gestion des fichiers - EISTI - ING 1

Ecole Internationale des Sciences du Traitement de l'Information

Définition

Un fichier informatique peut être défini comme une collection de données qui est stockée et conservée sur une mémoire secondaire comme un disque, une bande ou une disquette par exemple.

Propriétés

- ▶ Disponible en permanence (consultation, utilisation, modification)
- ▶ Supprimable par son propriétaire
- ▶ Élément clé de tout système informatique depuis les années 70 (transfert des fichiers papiers vers fichiers informatiques)

Pourquoi ?

- ▶ Contenu persistant
- ▶ Enregistrer sur un support externe (disque, CD, ...)
- ▶ Associé à un nom
 - ▶ Facilité d'utilisation
 - ▶ Impression
 - ▶ Archivage

Définitions et principes

- ▶ Toute donnée à traiter par un ordinateur doit être présente dans sa mémoire centrale.
- ▶ Cela nécessite donc un transfert des données stockées dans la mémoire secondaire.
- ▶ Problème d'organisation des données (physiques vs logiques), géré par le système de fichiers du système d'exploitation.
- ▶ Son rôle est de masquer les différences pour offrir à l'utilisateur un modèle logique algorithmiquement manipulable

Gestion des fichiers

Manipulation
algorithmique

Utilisation

Flots particuliers :
clavier

Flots particuliers :
écran

Fonctionnalités

- ▶ Fonctions importantes pour l'utilisateur : nommage des fichiers, protection des fichiers, opérations permises sur les fichiers.
- ▶ Fonctions à la charge du système : organisation physique en mémoire centrale.
- ▶ Solutions différentes entre les systèmes, donc incompatibilité générale entre les OS concernant la gestion des fichiers

Fichier

- ▶ suite finie d'informations qui se termine par un indicateur de fin (eof).
- ▶ unité logique stockée et gérée par le système de fichiers
- ▶ création par attribution d'un *nom externe*
- ▶ manipulation par association d'un *nom interne* au *nom externe*.

Noms et répertoires de fichiers

Répertoire

- ▶ gestion de l'ensemble des fichiers
- ▶ accès à l'emplacement des fichiers sur le support physique
- ▶ en général gestion par arborescence à plusieurs niveaux
- ▶ racine : répertoire principal.
- ▶ référencement d'un fichier : spécification du chemin d'accès de la racine de l'arborescence jusqu'au fichier désiré.

Exemple UNIX

Soit le fichier `cours.pdf` appartenant au sous répertoire `algo` qui est lui même dans le répertoire `ing1`. Chemin d'accès correspondant (nom externe) :

`/ing1/algo/cours.pdf`

Gestion des fichiers

Manipulation
algorithmique

Utilisation

Flots particuliers :
clavier

Flots particuliers :
écran

Liens logiques/physique

Structure physique

- ▶ suite de caractères
- ▶ suite d'enregistrements de taille fixe ou variable (dictionnaires)
- ▶ arborescence d'enregistrements de taille fixe ou variable (xml)

Organisation et accès

- ▶ Avantage : fichiers dynamiques (taille variable tant qu'il reste de la place sur le support physique)
- ▶ Inconvénient : lenteur d'accès (10^6 fois plus lent que mémoire centrale)
- ▶ L'accès aux enregistrements dépend de leur organisation à l'intérieur d'un fichier

Gestion des fichiers

Manipulation
algorithmique

Utilisation

Flots particuliers :
clavier

Flots particuliers :
écran

Différents types d'organisation

- ▶ organisation séquentielle :
 - ▶ enregistrements séparés par des espaces et stockés dans l'ordre d'entrée
 - ▶ correspondance entre ordre physique et logique
 - ▶ avantage : données de type et de taille différents
- ▶ organisation relative :
 - ▶ enregistrements de même taille
 - ▶ identifiés par un numéro qui indique l'emplacement relativement au début du fichier

Différents types d'accès aux enregistrements

- ▶ accès séquentiel :
 - ▶ traitement en séquence selon l'ordre physique (fichiers séquentiels) ou selon l'ordre des numéros d'enregistrement (fichiers relatifs)
 - ▶ nécessite de lire tous les éléments précédents l'élément recherché
- ▶ accès direct ou aléatoire :
 - ▶ nécessite un fichier en organisation relative
 - ▶ gestion similaire à un tableau en mémoire centrale
 - ▶ tous l'es éléments ont la même taille et sont accessibles via un indice sans devoir lire les précédents
 - ▶ insertion ou suppression ailleurs qu'en fin de fichier impossible sans recopie.

Généralités

- ▶ Opérations de manipulations via les langages de programmation
- ▶ Nécessite une syntaxe et une sémantique particulière
- ▶ Opérations à deux niveaux : fichier et enregistrement

Niveau fichier

- ▶ Création d'un fichier : fixe certains paramètres (nom, emplacement sur mémoire secondaire,...)
- ▶ Suppression d'un fichier : libération de l'espace sur mémoire secondaire (automatique sur certains systèmes selon date)
- ▶ Ouverture d'un fichier : chargement en mémoire centrale des attributs du fichier (date de création, taille, taille max, longueur d'enregistrement,...)
- ▶ Fermeture d'un fichier : libération de l'espace dans les tables du système (certains systèmes n'autorisent qu'un nombre limité de fichiers ouverts par processus)
- ▶ Position courante dans un fichier : directe et modifiable.

Niveau enregistrement

- ▶ Lecture d'un enregistrement : lecture par le système de fichiers à partir de la position courante sur la mémoire externe, puis transfert dans une zone dédiée de la mémoire centrale.
- ▶ Ecriture d'un enregistrement : écriture dans le fichier à partir de la position courante, avec possible augmentation de la taille (fin de fichier) ou remplacement de l'enregistrement (déjà présent)
- ▶ Ajout d'un enregistrement : ajout en fin de fichier avec augmentation de la taille
- ▶ Autres opérations non élémentaires : tri, mise à jour, fusion ...

Gestion des fichiers

Manipulation
algorithmique

Utilisation

Flots particuliers :
clavier

Flots particuliers :
écran

Définition

- ▶ Suite de caractères, chiffres, lettres et autres
- ▶ Contient des informations de fin de ligne et de fin de fichier
- ▶ Flots connus (standard)
 - ▶ Clavier : ligne de flot en entrée
 - ▶ Écran : ligne de flot en sortie

Contenu

- ▶ Flot : suite d'éléments parmi
 - ▶ nombres entiers ou réels, séparés par des espaces
 - ▶ lignes d'un texte
 - ▶ caractères
 - ▶ mélange des trois précités

Exemple

Différents flots

- ▶ Flot de nombres

- ▶ 12 145 -34 -3.324 0 ...

- ▶ Flot de lignes

- ▶ Programme

Pourquoi ?

- ▶ Obligation d'ouvrir un flot
 - ▶ lecture \implies impossibilité d'écrire
 - ▶ écriture \implies impossibilité de lire

Ouverture

- ▶ lecture : `ouvrir1(flott,chemin)`
- ▶ écriture : `ouvrir(flott,chemin)`

Fermeture

`fermer(flott)`

Fonctions associées

- ▶ Prédicat de fin de flot : `eof(flott)`
- ▶ Positionnement au début du flot : `debut(flott)`

Lecture

- ▶ simple : `lire(flot, elt1[, elt2])`
- ▶ avec changement de ligne : `lireln(flot, elt1[, elt2])`
- ▶ tout une ligne :
`lireligne(f :flot[,len :entier]) :chaine`

Écriture

- ▶ simple : `ecrire(flot, elt1[, elt2])`
- ▶ avec changement de ligne : `ecrireln(flot, elt1[, elt2])`

Entrée standard

- ▶ Flot d'entrée standard
- ▶ Toujours ouvert lors de l'exécution d'un programme
- ▶ Attention utilisation parfois délicate. . .

Lecture

lire(a)	Lecture d'une valeur
lire(a, b, c, ...)	Lecture de plusieurs valeurs
lireln(a)	Lecture et passage à la ligne
lireln(a, b, ...)	Idem

Sortie standard

- ▶ Flot de sortie standard
- ▶ Toujours ouvert lors de l'exécution d'un programme
- ▶ Divers possibilité de formatage

Écriture

ecrire(a)	Écriture d'une valeur
ecrire(a, b, c, ...)	Écriture de plusieurs valeurs
ecrireln(a)	Écriture et passage à la ligne
ecrireln(a, b, ...)	Idem

Autres possibilités de présentation

écriture	effet
ecrire(a)	12
ecrire(a : 4)	12
ecrire(b)	5.123
ecrire(b : 5 :2)	5.12
ecrire(b : 6 :4)	5.1230
ecrire(s)	toto
ecrire(s : 5)	toto

Gestion des fichiers

Manipulation
algorithmique

Utilisation

Flots particuliers :
clavier

Flots particuliers :
écran