



**Université Yahia Farès de Médéa**  
**Département: GEI**  
**2<sup>ème</sup> année Informatique (L.M.D)**  
**Option : RSI/SIIA**

# **Systeme de Gestion de Fichiers (SGF)**

**A. KHELDOUN**

**Émail:** [univ.mede@gmail.com](mailto:univ.mede@gmail.com)

**Année Universitaire: 2012-2013**

# Fichiers

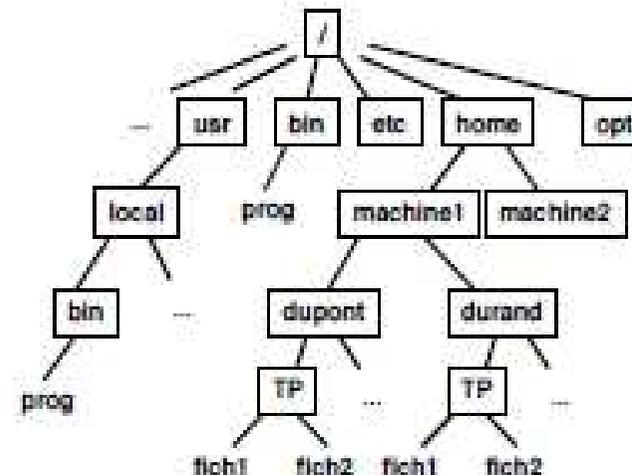
- ❑ **Définition**: ensemble d'informations regroupées en vue de leur conservation et de leur utilisation dans un système informatique.
- ❑ **Système de gestion de fichiers (SGF)** : partie du système d'exploitation qui conserve les fichiers et permet d'y accéder.
  
- ❑ **Fonctions d'un SGF**
  - **Conservation permanente des fichiers** (permanente : indépendamment de L'exécution des programmes et de l'intégrité de la mémoire principale)
    - conservation en mémoire secondaire (disque),
  - **Organisation logique et désignation** des fichiers,
  - **Partage et protection** des fichiers,
  - Réalisation des **fonctions d'accès** aux fichiers.
- ❑ **Exemples SGF**
  - **Windows** : FAT16, FAT32, NTFS,...
  - **Unix** : ufs (unix BSD) , ext2fs,...
  - **MacOS** : HFS

# Désignation des fichiers

- Principe de la désignation symbolique (*organisation hiérarchique*)
  - Les noms forment une arborescence,
  - Noeuds intermédiaires = catalogues, ou répertoires – en anglais *directory* (ce sont aussi des fichiers),
  - Noeuds terminaux (feuilles) = fichiers simples,
  - Nom (universel ou absolu) d'un fichier = le chemin d'accès depuis la racine (en anglais": *path*)

Exemples de noms universels :

```
/
/bin
/usr/local/bin/prog
/home/machine1/dupont/TP/fich1
/home/machine1/durand/TP/fich1
```



## Désignation des fichiers

### □ répertoire courant

- par défaut, tout usager a un répertoire courant de base (home directory), par exemple /home/machine/user4,
- on peut changer de répertoire courant au moyen de la commande **cd <nom du répertoire>**
- on peut connaître le nom absolu du répertoire courant par **pwd**
- on peut connaître le contenu du répertoire courant par **ls** (**ls -l** est plus complet)

```
$ ls -l
-rwxr-xr-x 1 root admin 1000 2003-17-02 20:05 TP.c
drwxr-xr-x 1 root admin 1000 2013-17-02 20:00 rep/
```

## Utilisations courantes des fichiers

□ Dans Unix, le contenu d'un fichier est simplement une **suite d'octets**, sans autre structure. L'interprétation de ce contenu dépend de l'utilisation.

➤ **Programmes exécutables**

✓ Commandes du système ou programmes créés par un usager

➤ **Fichiers de données**

✓ Documents, images, programmes sources, etc.

➤ **Fichiers temporaires servant pour la communication**

✓ Ne pas oublier de les supprimer après usage

## Interface système pour l'utilisation des fichiers

❑ Au niveau de l'interface des appels système, un fichier est représenté par un **descripteur**.

❑ Primitive fopen

FILE \* fopen(const char \* *fname*, const char \* *mode*);

*fname* : le nom du fichier à ouvrir

*mode* : le mode d'ouverture : "r", "w", "a", "r+", "w+", "a+";

Retourne : pointeur sur struct FILE, NULL si erreur.

❑ Primitive fclose

int fclose(FILE \* *fic*);

*fic* : pointeur sur le fichier ouvert qu'on veut fermer

Retourne : 0 ou EOF si erreur

## Interface système pour l'utilisation des fichiers

### ❑ Primitive fprintf

int fprintf (FILE \* *flot, const char format, ...*);

Retourne : nombre d'octets écrits, ou valeur négatif si erreur.

### ❑ Primitive fscanf

int fscanf (FILE \* *flot, const char format, ...*);

Retourne : nombre de paramètres reconnus, ou EOF si erreur de flot.

### ❑ Primitive chmod

int chmod(char \* *fname, mode\_t mode*);

*fname* : le nom du fichier

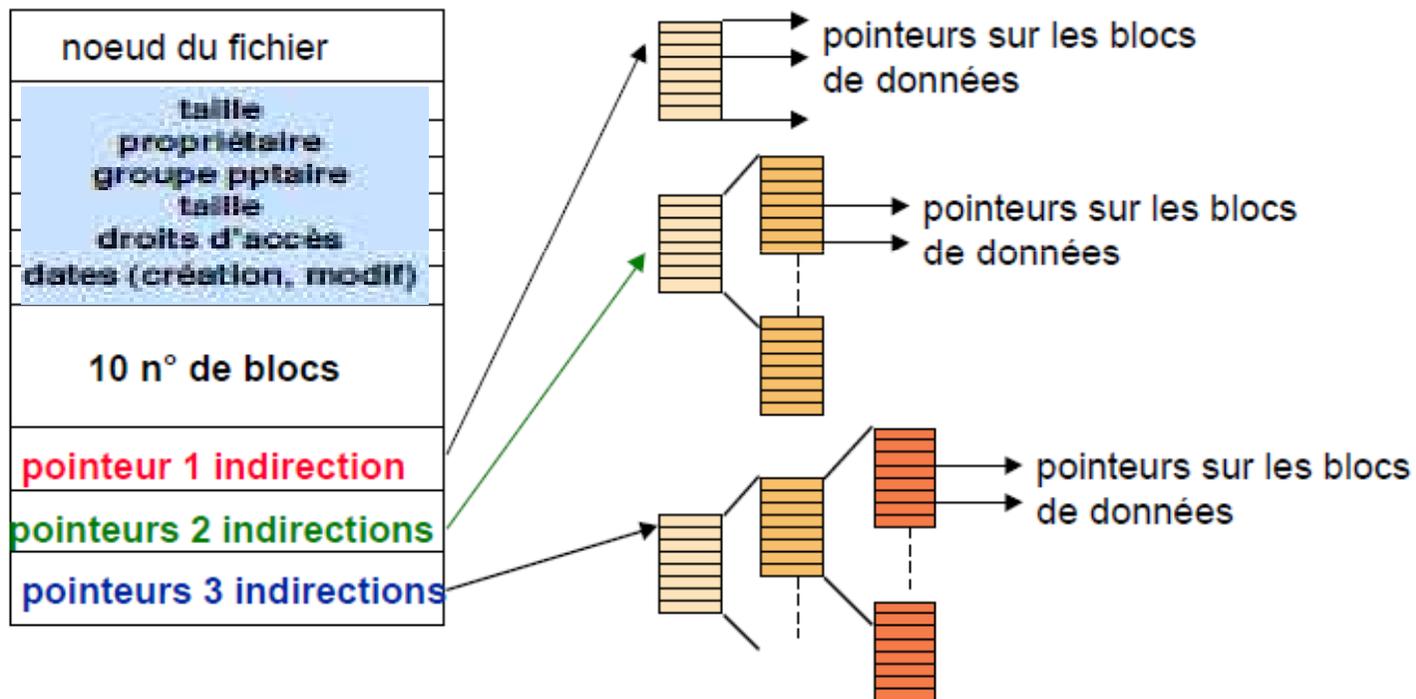
*mode* : mode d'accès (<sys/stat.h>)

## Interface système pour l'utilisation des fichiers

- ❑ **Représentation physique d'un fichier**
  - Un fichier est représenté physiquement par un ensemble de blocs (suite d'octets de taille fixe) sur disque. Les blocs ne sont pas nécessairement contigus.
  - Typiquement, la taille d'un bloc est 8 Ko.
  
- ❑ **Structures de données pour la gestion interne des fichiers**
  - La structure principale (invisible aux usagers) associée à un fichier est un descripteur appelé **i-node** du fichier. Les i-nodes sont contenus dans une table globale.

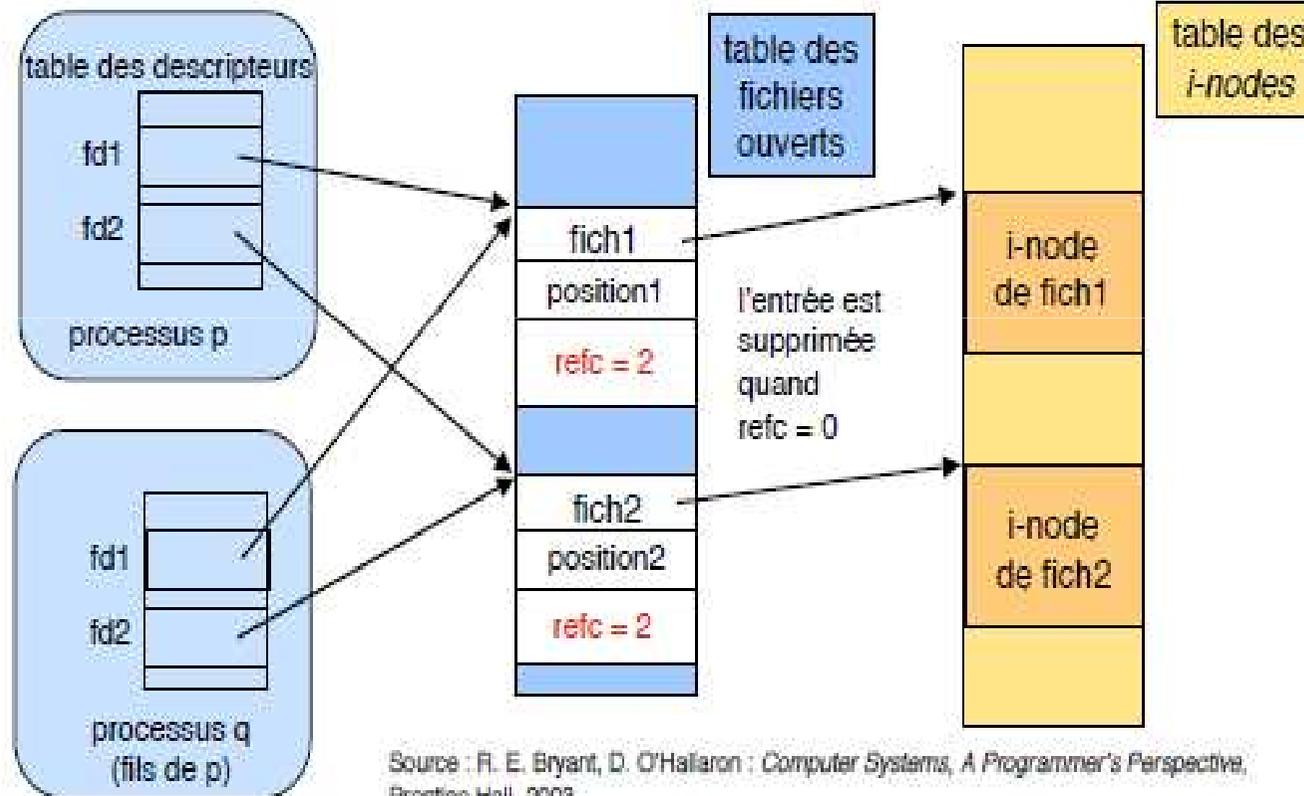
# Implantation des fichiers : par i-nodes

Une table i-node par fichier sur le disque



# Partage de fichiers entre père et fils

tables communes du système



# Implantation des répertoires

répertoire racine

1	.
1	..
4	bin
7	dev
5	lib
3	etc
9	usr
8	tmp

↓ ss-répertoire

n° i-node

i-node 9

info
125

/usr

bloc 125

9	.
1	..
17	claudio
57	luc
44	prog

i-node 44

info
688

/usr/prog

## Protection des fichiers

### ❑ Fichiers ordinaires

r W X	r W X	r W X
propriétaire (u)	groupe (g)	autres (o)

➤ exemple (fichier fich) : **rwX r-- r--** : tout accès pour le propriétaire, lecture seule pour tous les autres.

➤ **chmod go+w fich** : donne le droit **w** au groupe et aux autres

➤ **chmod o-w fich** : retire le droit **w** aux autres

❑ **Répertoire**: même chose, mais le droit **x** signifie “recherche dans le répertoire”

## Conclusion

- ❑ **Le SGF du S.E est une partie complexe mais nécessaire:**
  - **Il permet de stocker des données sur un support externe**
  - **Il utilise des concepts logiques faciles à mettre en œuvre**
  - **Il peut être utilisé en mode commande**
  
- ❑ **Chaque S.E. à son propre SGF avec ses caractéristiques:**
  - **Organisations**
  - **Modes d'accès**
  - **Multiutilisateurs**
  - **Niveau de protection et sécurité**