

Redirections - Stockage

Cours 4

E. Paviot-Adet

Emmanuel.Paviot-Adet@parisdescartes.fr
Bureau B2-2

IUT Paris Descartes
DUT Informatique - 1^{ère} année

Sommaire

- 1 Redirections d'entrées/sorties
 - Généralités
 - Syntaxe
- 2 Stockage de fichiers
- 3 Liens matériels/symboliques
 - Lien matériel
 - Lien symbolique
 - Aspects pratiques

Plan

- 1 Redirections d'entrées/sorties
 - Généralités
 - Syntaxe
- 2 Stockage de fichiers
- 3 Liens matériels/symboliques
 - Lien matériel
 - Lien symbolique
 - Aspects pratiques

Flux associés à un processus

Programmes utilisables sous l'invite de commandes

Chaque processus possède :

`stdin` (n° 0) flux d'entrée standard

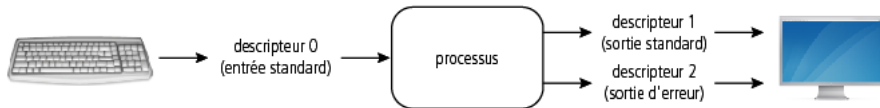
- Par défaut le clavier (réception d'information)

`stdout` (n° 1) Flux de sortie standard

- Par défaut l'écran (résultats, informations...)

`stderr` (n° 2) Flux de sortie des erreurs

- Par défaut l'écran (messages d'erreurs)



Nature et opérations sur les flux

Flux

- Nature : fichiers texte gérés par le système
 - Vision qu'en a le programmeur
 - Le système fait le lien entre ces fichiers et les périphériques
- Opérations : redirection des flux
 - Vers/depuis un fichier texte (pas de copie vers l'écran)
 - Un flux de sortie vers un flux d'entrée
 - Communication entre processus
- Noms emprunté au C
 - `stdin` STandarD INput
 - `stdout` STandarD OUTput
 - `stderr` STandarD ERRors
- En C++ : `cin`, `cout`, `cerr`

Redirection de stdout

Deux syntaxes

Redirection de *stdout* vers *fichier_texte*

- *commande* > *fichier_texte*
 - Si *fichier_texte* existe
 - Création d'un nouveau fichier ne contenant que les résultats de *commande*
 - Destruction des données précédemment contenues dans *fichier_texte*
- *commande* >> *fichier_texte*
 - Si *fichier_texte* existe
 - Ajout des résultats de *commande* après les données de *fichier_texte*

Redirection de stderr

Deux syntaxes

Redirection de *stderr* vers fichier_texte

- commande **2**> fichier_texte
 - Si fichier_texte existe
 - Création d'un nouveau fichier ne contenant que les erreurs de commande
 - Destruction des données précédemment contenues dans fichier_texte
- commande **2**>> fichier_texte
 - Si fichier_texte existe
 - Ajout des erreurs de commande après les données de fichier_texte
- Le flux *stderr* est numéroté 2, d'où les syntaxes **2**> et **2**>>

Redirection de stdin

Syntaxe

Redirection de fichier_texte vers *stdin*

- commande < fichier_texte
 - commande lit les informations dans fichier_texte

Mnémotechnique

- commande est toujours en début de ligne de commande
- < ou > indiquent la direction du flux de données
 - commande < fichier_texte
 - Les données vont du fichier vers la commande
 - Elles sont lues
 - commande > fichier_texte
 - Les données vont de la commande vers le fichier
 - Elles sont écrites

Combinaison des redirections

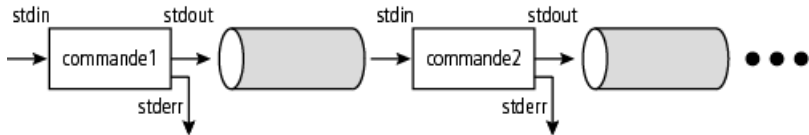
- Fichiers différents
 - Mettre les différentes syntaxes dans la commande
 - `sort < aTrier.txt > resultat.txt 2> err.txt`
- Même fichier pour les 2 flux de sortie
 - `mkdir rep > resultat.txt 2>&1`
 - `2>` on redirige `stderr`
 - `&1` vers le flux numéro 1 (stdout donc)
- Une curiosité
 - `mkdir rep 2> /dev/null`
 - `/dev/null` est un pilote pour un périphérique inexistant
 - Les messages sur `stderr` sont oubliés

Les tubes

Redirection vers un autre processus

La sortie de commande1 est utilisée comme entrée de commande2

- commande1 | commande2
 - Communication par l'intermédiaire d'un tampon (buffer)
- Exemples
 - `cat fic.txt | sort > trié.txt`
 - `env | grep HOME`



Commandes filtres

- sort** tri
- uniq** suppression des doublons de l'entrée (à coupler avec **sort**)
- grep** recherche d'une chaîne de caractères dans un ensemble de fichiers
 - tr** traduite, compresser et/ou éliminer des caractères de l'entrée
 - cut** supprimer une partie de chaque ligne d'un fichier
- paste** regrouper les lignes de différents fichiers
- sed** substitution de texte
- awk** appariement de chaînes dans des champs, opérateurs de sous-chaînes

Plan

- 1 Redirections d'entrées/sorties
 - Généralités
 - Syntaxe
- 2 Stockage de fichiers
- 3 Liens matériels/symboliques
 - Lien matériel
 - Lien symbolique
 - Aspects pratiques

Fichiers et système de gestion de fichier

Fichier physique

Blocs de données stockés sur une mémoire de masse

Système de gestion de fichiers (SGF)

Programme du système d'exploitation chargé de la gestion des fichiers physiques. Il stocke des informations relatives aux fichiers (*métadonnées*). Caractéristiques :

Réseau Accès à des fichiers à distance

Distribué Gestion de plusieurs nœuds du réseau

Journalisation Stockage des opérations d'écriture tant qu'elles ne sont pas terminées

- Permet de garantir l'intégrité des données en cas d'arrêt brutal

Fichiers et système de gestion de fichier

Fichier physique

Blocs de données stockés sur une mémoire de masse

Système de gestion de fichiers (SGF)

Programme du système d'exploitation chargé de la gestion des fichiers physiques. Il stocke des informations relatives aux fichiers (*métadonnées*). Caractéristiques :

Réseau Accès à des fichiers à distance

Distribué Gestion de plusieurs nœuds du réseau

Journalisation Stockage des opérations d'écriture tant qu'elles ne sont pas terminées

- Permet de garantir l'intégrité des données en cas d'arrêt brutal

Fonctions de SGF

- Manipulation des fichiers
 - Création
 - Destruction
 - Modification...
- Gestion des mémoires secondaires
 - Places libres, occupées...
- Localisation des fichiers
- Sécurité et contrôle
- Quelques noms :
 - FAT, NTFS, HPFS, ext2, ext3, ext4...

Métadonnées

Métadonnées courantes sous UNIX

- Droits d'accès
- Dates
 - Dernier accès, modification des données, modification des métadonnées
- Propriétaire
- Groupe d'appartenance du fichier
- Taille du fichier
- Nombre de noms désignant le fichier physique
- Nombre de blocs utilisés sur la mémoire secondaire
- Adresse disque du début du fichier physique
- Nature du fichier

Inodes

Inodes

Structure de données du SGF contenant les métadonnées relatives à un fichier

Signification Contraction de index-node (nœud d'index)

Stockage Dans une table au début de chaque volume

- *Table des inodes*

Numéro d'inode Numéro d'entrée dans la table des inodes

- Ce numéro est unique pour chaque volume
- Identification d'un fichier physique

Répertoire (sous UNIX)

Fichier contenant des informations permettant de lier chaque nom de fichier à son numéro d'inode

Inodes

Inodes

Structure de données du SGF contenant les métadonnées relatives à un fichier

Signification Contraction de index-node (nœud d'index)

Stockage Dans une table au début de chaque volume

- *Table des inodes*

Numéro d'inode Numéro d'entrée dans la table des inodes

- Ce numéro est unique pour chaque volume
- Identification d'un fichier physique

Répertoire (sous UNIX)

Fichier contenant des informations permettant de lier chaque nom de fichier à son numéro d'inode

Plan

- 1 Redirections d'entrées/sorties
 - Généralités
 - Syntaxe
- 2 Stockage de fichiers
- 3 Liens matériels/symboliques
 - Lien matériel
 - Lien symbolique
 - Aspects pratiques

Lien matériel

Lien matériel (*hard link*)

Référence à des données physiques sur un volume de stockage

- Notion introduite par UNIX
 - Possibilité d'associer plusieurs noms aux mêmes données
 - Autant de liens matériels que de noms
- Conséquence
 - Tous les liens symboliques d'un même fichier physique se trouvent sur le même volume
- Suppression (d'un nom) de fichier
 - Un compteur de liens matériels est associé à chaque fichier
 - Suppression \Rightarrow décrétement du compteur concerné
 - Si le compteur est nul, le fichier est physiquement supprimé

Liens matériels dans la commande `ls`

Terminal

```
DebianArchy:~# ls -l
total 36
drwxr-xr-x 2 root root 4096 2008-10-31 11:18 Desktop
-rw-r--r-- 1 philippe root 36 2008-08-25 15:22 essai
-rw-r--r-- 1 root root 63 2008-08-25 15:32 essai_grep
-rwxr-xr-x 1 philippe philippe 390 2008-08-25 16:06 script_grep
-rwxr-xr-x 1 root root 390 2008-08-25 15:58 script_grep~
-rwxr-xr-x 1 philippe root 454 2008-08-25 18:10 script_tant_que
-rwxr-xr-x 1 root root 452 2008-08-25 18:03 script_tant_que-
-rwxr-xr-x 1 philippe root 541 2008-08-25 18:54 script_while_fichier
-rwxr-xr-x 1 root root 560 2008-08-25 18:24 script_while_fichier-
```

Annotations:

- type
- permissions du
 - propriétaire
 - groupe
 - autres
- nombre de liens physiques
- propriétaire
- groupe
- taille (octets)
- date et heure de dernière modification
- nom du fichier

Lien symbolique

Lien symbolique (*symlink*)

Type particulier de fichier contenant une référence à un autre fichier ou répertoire

- La référence peut être un absolu ou relatif
 - Peut être considéré comme un alias
- Transparence aux opérations
 - Beaucoup d'opérations ne font pas la différence entre un lien matériel et la cible du lien symbolique
 - Il y a des exceptions comme *rm* par exemple
 - Les liens symboliques ne sont généralement pas suivis lors d'opérations récursives
 - Opérations sur une sous-arborescence complète

Liens symboliques dans la commande `ls`

```
Terminal — bash — ttys000
mbp-manu:tmp paviot$ ls -al
total 16
drwxr-xr-x  4 paviot paviot  136 11 oct 12:44 .
drwxr-xr-x@ 122 paviot paviot 4148 11 oct 06:31 ..
-rw-r--r--  1 paviot paviot   5 11 oct 12:43 file
lwxr-xr-x  1 paviot paviot   4 11 oct 12:44 fileSym -> file
mbp-manu:tmp paviot$
```

1 lien matériel

lien symbolique

Stockage des liens symboliques

Premières implémentations

La référence était du texte ordinaire. Un drapeau indiquait qu'il ne fallait pas ouvrir le fichier, mais la cible désignée par le contenu

- Lent
 - Chaque lien symbolique impose une ouverture de fichier
- Prend de la place sur le disque
 - Quelques octets utilisés pour une unité d'allocation sur le disque
 - Généralement 512 ou 1 024 octets
- Baptisés retrospectivement *slow symlink*

Stockage des liens symboliques

Implémentations actuelles

La référence de la cible est stockée dans une zone spéciale de l'inode.

- Le système maintient les tables des inodes en mémoire centrale
 - Accès rapide
 - On ne mobilise plus une unité d'allocation sur le disque
 - Place limitée
 - Retour à l'ancienne méthode si la référence est trop longue
- Baptisés *fast symlink*

Lien symbolique - Suppression de fichier

Suppression d'un lien symbolique

Le fichier cible existe toujours, seul le lien symbolique est supprimé

Suppression du fichier cible

Le lien symbolique existe toujours, mais il désigne un fichier inexistant. On dit que le lien est cassé. En anglais :

- *broken*
- *orphaned*
- *dangling*

Lien symbolique - Suppression de fichier

Suppression d'un lien symbolique

Le fichier cible existe toujours, seul le lien symbolique est supprimé

Suppression du fichier cible

Le lien symbolique existe toujours, mais il désigne un fichier inexistant. On dit que le lien est cassé. En anglais :

- *broken*
- *orphaned*
- *dangling*

Comparaisons

Lien matériel

- Ne désigne pas de répertoire
- Doit être dans le même volume que le fichier cible

Lien symbolique

- Désigne un fichier ou un répertoire
- Pas de contrainte sur les volumes contenant le lien et sa cible
- Mise à jour facile de la cible
- Possibilité d'adressage relatif

Comparaisons

Lien matériel

- Ne désigne pas de répertoire
- Doit être dans le même volume que le fichier cible

Lien symbolique

- Désigne un fichier ou un répertoire
- Pas de contrainte sur les volumes contenant le lien et sa cible
- Mise à jour facile de la cible
- Possibilité d'adressage relatif

Commande de base - ln

Commande *ln* : Création de liens (signification : LiNk)

Syntaxe simplifiée

ln [-s] Fichier cible Lien

- -s : Création d'un lien symbolique

En pratique

- Exemple :
 - *ls > file*
 - *ln -s file fileSym*
- Note : Par défaut, un lien matériel est créé