R1.08

TP 1 : Gestion des fichiers

R&T Béthune

Résumé

Il est fortement conseillé de vous faire un compte rendu de travail détaillé. Celui ci est personnel. Mettez-y toutes vos remarques.

1 Rappels

1.1 Les chemins absolus

Pour accéder à un fichier, il faut connaître son nom. Le nom d'un fichier dans l'arborescence correspond au chemin que l'on prend pour y arriver. Par exemple, pour accéder au fichier cpp.c du schéma, il faut partir de la racine /, puis aller dans le répertoire des utilisateurs (users), puis dans le répertoire du groupe auquel appartient l'utilisateur (cours), puis aller dans le répertoire de l'utilisateur (moi). Enfin, ce dernier a créé deux sous-répertoires pour ses besoins : il faut choisir le dossier contenant les sources des programmes (src), puis désigner enfin le fichier (cpp.c).

Le nom complet est donc la mise bout à bout des noms constituant le chemin. La barre oblique / a été choisie comme convention pour séparer ces noms (sauf le premier / qui symbolise la racine). Le nom complet du fichier, c'est-à-dire son chemin absolu, est : /users/cours/moi/src/cpp.c

De même, le chemin absolu du fichier calendar est /users/cours/moi/calendar

1.2 Le répertoire courant

À tout moment de sa session, chaque utilisateur se trouve dans un répertoire de l'arborescence du système. Ce répertoire peut être représenté comme un nom absolu. Par exemple /users/cours/moi

On peut ainsi désigner un fichier dans ce répertoire, il suffit de donner le nom du fichier. Par exemple, si le catalogue de travail est /users/cours/moi, le fichier calendar peut être désigné simplement par calendar.

On désigne donc le fichier directement par son nom. De même, si le catalogue de travail est le répertoire /users/-cours/moi/src, on peut désigner plus simplement les deux fichiers qui y résident par cpp.c et pstree.c.

~ (tilde) : Ce répertoire a une signification particulière, c'est le répertoire de connexion. C'est celui sur lequel on se retrouve en début de session. Il est souvent appelé aussi le home directory (HOME).

1.3 Les chemins relatifs

Les chemins absolus sont complexes et longs à taper. Pour simplifier, on utilise souvent les chemins relatifs. Cette notion repose sur celle de catalogue de travail, c'est-à-dire du répertoire courant. En effet, si nous sommes dans le répertoire /users/cours/moi et que nous voulions désigner le fichier pstree.c situé dans le répertoire src, le chemin sera simplement (on décrit uniquement le chemin restant à effectuer) src/pstree.c

Si le catalogue de travail est /users, le chemin relatif pour désigner le fichier calendar est cours/moi/calendar

Très important : les chemins relatifs ne commencent jamais par le caractère barre oblique /. C'est la seule manière de distinguer un chemin relatif d'un chemin absolu. Remarque : lorsqu'un nom absolu est utilisé pour désigner un fichier, il y a unicité de ce fichier (on suit le chemin depuis la racine, on est donc bien sûr d'atteindre le bon fichier). En revanche, lorsqu'on utilise des noms relatifs, on ne peut plus affirmer que le fichier est unique. Par exemple le fichier src/cpp.c peut représenter un fichier depuis /users/cours/moi ou depuis /users/cours/lui ou encore /users/ens/u1/src/cpp.c. . .

1.4 Les conventions . et ..

Le symbole . (point) représente le répertoire de travail. Ceci évite de taper le nom absolu du répertoire courant dans certaines commandes.

Le symbole .. (deux points) représente le nom du répertoire situé juste au-dessus (répertoire père). Ainsi, si le répertoire courant est /users/cours/moi alors .. représente le répertoire /users/cours.

1.5 Autocomplétion

La touche tab permet de compléter automatiquement une commande ou un nom de fichier.

— Dans le cas d'une commande, si une seule complétion est possible, la commande est affichée complétée. Par exemple, si on tape clea suivi de tab, la commande est complétée en clear.

— Toujours pour une commande, si plusieurs complétions sont possibles, un bip retentit ; un second appui sur tab affiche alors toutes les commandes possibles.

— Le même mécanisme peut être utilisé pour les noms de fichiers : tab complète un nom de fichier (utilisé dans une commande) s'il n'y a qu'une possibilité et sinon deux appuis permettent de lister toutes les complétions.

Exercice : Tapez pw suivi de deux fois tab pour voir toutes les complétions possible de la commande.

2 Exercices

2.1 L'aide

Pour avoir de l'aide sur une commande, il faut utiliser l'une des trois solutions suivantes :

```
commande –h
commande help
man commande
```

Les options *-h* et *–help* sont moins complètes que la commande *man*.

Exercice : Testez les 3 méthodes sur la commande *ls* et comparez les résultats.

2.2 Gestion et manipulation de fichiers

1. Rappelez les commandes de base de navigation dans le système de fichiers. Utilisez ces commandes afin d'explorer l'arborescence du système. Testez les différents modes d'affichage.
2. Créez un fichier *essai* contenant le texte « ceci est un essai » à la racine de votre répertoire de travail.
3. Créez un fichier *mon fichier a moi.txt* (avec les espaces dans le nom du fichier)
4. Donnez une commande qui permet d'afficher le contenu du fichier *essai*.
5. Affichez le nom du répertoire de travail
6. Donnez une commande qui permet de créer dans */tmp* un répertoire *moduleR3* qui contient un répertoire nommé *tp1*, depuis votre répertoire de travail
7. *En étant dans le répertoire /tmp* , créez un répertoire *moduleR3bis*
8. Faites les questions suivantes depuis votre répertoire de travail et depuis le répertoire */tmp* :
   (a) Copiez le fichier *~/essai* dans le répertoire */tmp/moduleR3/tp1*
   (b) Déplacez le fichier */tmp/moduleR3/tp1/essai* dans le répertoire */tmp/moduleR3bis*
   (c) Effacez les fichiers */tmp/moduleR3/essai*
9. Dans le répertoire /tmp*/moduleR3bis*
   (a) Créez un lien synbolic du fichier *essai* nommé *essai.ln*
   (b) Créez un lien physique du fichier *essai* nommé *essai.ph*
   (c) En utilisant la commande *man ls* cherchez comment visualiser le numéro i-node d'un fichier. Ensuite, comparer les numéros i-node des trois fichiers crées précédemment. Que constate-t-on ?
   (d) Ajoutez une ligne dans le fichier *essai* une ligne de texte. Visualisez ensuite le contenu des fichiers *essai.ln* et *essai.ph*
   (e) Refaites l'expérience précédente avec le fichier *essai.ph*
   (f) En utilisant la commande *ls -l* , affichez les détails des fichiers *essai**
   (g) En utilisant la commande *file*, affichez la nature des trois fichiers essai. Commentez.
   (h) Supprimez le fichier essai , puis affichez les contenus des fichiers *essai.ln* et *essai.ph* . Expliquez les résultats !
   (i) Testez avec un répertoire !
10. Donnez une commande qui retourne le contenu du répertoire « *~/* » trié en ordre alphabétique inversée.
11. Que fait la commande *touch* ? et la commande *file* ?
12. Créez des fichiers vides (de tailles 0 octets) dans */tmp*.
13. Donnez une commande qui permet d'effacer le contenu du */tmp*. (y compris les sous répertoires de */tmp*)

2.3 Utilisation des méta-caractères

Placez-vous dans votre répertoire personnel pour exécuter les commandes suivantes.

1. Créez un répertoire /tmp/*tp_intro_sys* et déplacez vous dedans.
2. Copiez tous les fichiers qui se trouvent dans */usr/include* dont le nom commence par ***a*** et qui ont l'extension ***.h***
3. Effacez du répertoire les fichiers ayant l'extension ***.h***
4. Affichez la liste de tous les fichiers du répertoire */usr/include* qui ont une *extension*.
5. Copiez dans *le répertoire* les fichiers qui se trouvent dans */usr/include* et dont le nom commence par ***a c*** ou ***i***.
6. Exécutez la commande *rmdir /tmp/tp_intro_sys*. Quel est le résultat de cette commande ?
7. Expliquer ce que fait la commande *ls **
8. Créer, de façon optimale, dans */tmp/tp_intro_sys* , les fichiers *a1.avi, a2.avi, a3.avi, b1.avi, b2.avi, b3.avi, c1.avi, c2.avi et c3.avi*
9. Effacer uniquement les fichiers commençant par ***a***