3R-IN3 – Systèmes d'exploitation
TD1

Xavier Hilaire
x.hilaire@esiee.fr

14 mars 2025

1 Droits d'accès

L'utilisateur `fred`, de groupe `fred`, lance une commande `ls -l` dans un terminal et obtient la sortie suivante :

```
$ ls -l
total 316
drwxr-xr-x 4 fred fred   4096 29 sept.  2020  Calltech101
-rwxrwxr-x 1 fred fred    531 28 nov.   2011  cent_moment.m
-rwxr--r-x 1 eric fred   1463 17 sept.  2018  evalBOF
-rw-r--r-- 1 fred fred   1713 26 sept.  2018  #evalBVF.m#
-rw-r-xr-- 1 fred info   1749 27 sept.  2018  evalBVF
-rw-rwxr-x 1 fred fred   1586 28 nov.   2011  feature_vec.m
-rwsrws--x 1 eric info  27331 28 nov.   2011  feature_vec
-rw-rw---- 1 fred fred   4660 28 nov.   2011  feature_res
-rw-r--r-- 1 fred fred   8358 18 sept.  2018  index.html
-rw-r--r-- 1 fred fred   7092 17 sept.  2018  index.html~
-rw-r-xr-x 1 fred root   2208 17 sept.  2018  LabStats
-r--r--r-- 1 fred fred    344 24 sept.  2018  myExtractor1.m
-r--rw-r-- 1 fred fred    344 26 sept.  2018  myExtractor2.m
-r--r--r-- 1 fred fred    427 27 sept.  2018  myExtractor3.m
----rw-r-- 1 fred fred    177 17 sept.  2018  myFisher.m
-rw-r--r-- 1 fred fred   1107 24 sept.  2018  ransac.m
-rw-r--r-- 1 fred fred   1277 24 sept.  2018  ransac.m~
drw-r-xr-- 2 fred info   4096 12 oct.   2018  Releves
-------r-- 1 fred info 242573 17 sept.  2018  toto.mat
$
```

Questions :

1. Sur quels fichiers fred n'a t'il pas le droit de lecture ? D'écriture ? D'exécution ?
2. Quelles commandes devrait-il lancer pour que les utilisateurs du groupe primaire `info` puissent lire le fichier `toto.mat`, et qu'il puisse lui-même lire et écrire sur ce fichier ?
3. Supposez qu'à partir du répertoire listé, il lance la commande `ls␣-l␣Releves`. Cette dernière réussira t'elle ?
4. Le programme binaire `features_vec` a besoin d'ouvrir le fichier `features_res` en écriture pour y ajouter des résultats. Le pourra t'il s'il est lancé par fred ?

2 Jouer avec les fichiers

2.1 La commande ls

Utilisée sans arguments, la commande ls vous permet de lister les fichiers du répertoire courant. Consultez sa page manuel ('q' pour la quitter) : man ls

1. Placez-vous dans votre répertoire par défaut, et appelez ls. Appelez ensuite ls avec les options -l et -l -a, soit ls -l et ls -la Quel est l'effet de -l ? De -a ?
2. Listez le contenu de la racine (/) avec les options -l et -a
3. A quoi pourrait correspondre le premier caractère 'd' précédent les droits ? Vérifier cela en appelant à nouveau ls avec le bon argument.

2.2 La commande cp

La commande cp vous permet de copier des fichiers. Il y a deux formes possibles :
— cp src dest : copie un seul fichier source src vers le fichier cible dest
— cp src1 src2 ... dest : copie plusieurs fichiers src1, src2, ... vers le répertoire dest
Consulter sa page manuel : man cp

1. Copiez le fichier `/etc/passwd` vers votre répertoire par défaut. Avec `ls␣-l`, listez `/etc/passwd` et `~/passwd` (en une ou deux commandes, peu importe) La datation est-elle la même ?
2. Supprimez le fichier `~/passwd` en faisant `rm␣~/passwd` Puis refaites la copie, mais avec l'option -p de cp. Et maintenant ?
3. La commande stat (status) vous permet d'obtenir toutes sortes d'informations sur un ou des fichiers. Faites un `stat␣/etc/passwd␣~/passwd` Les datations sont-elles toutes identiques ?

2.3 La commande ln

La commande ln (link) vous permet de créer un lien symbolique (avec l'option -s) ou matériel (sans l'option -s).

1. Dans le terminal, refaites un ls -l Puis lancez `ln␣~/passwd␣~/lien`, et refaites encore un ls -l Qu'est-ce qui a changé ? Vérifiez avec `stat`.
2. Gedit (Gnome editor) est un éditeur texte. Editez votre fichier passwd en faisant `gedit␣~/passwd` Supprimez une ou deux lignes, puis enregistrez-le. Retournez dans le shell, et listez à nouveau `~/passwd` et `~/lien` Conclusion ? (Si vous avez un doute : la commande diff fic1 fic2 compare les fichiers fic1 et fic2 et affiche les différences quand elle en trouve. Comparez donc `~/passwd` et `/etc/passwd`, puis `~/passwd` et `~/lien`)

3 Résolution des noms de fichiers

Cet exercice doit être fait sur machine

1. Ouvrez un terminal, et essayez-y les commandes suivantes :

```
cd
echo $PWD
echo ~
cd /tmp
echo $PWD
echo ~
```

Quels résultats du cours retrouvez-vous ici ?

2. En utilisant ls ou echo, ou mieux encore, le programme `args1.c` vu en cours, faire afficher les noms de fichiers ou de répertoires :
   (a) de 3 lettres exactement, situés au niveau 0 (ç-à-d rattachés directement à la racine)
   (b) qui se terminent par m, au niveau 1
   (c) qui se terminent par un chiffre, au niveau 1
   (d) d'au moins 3 lettres, figurant dans /etc, qui contiennent un caractère 'e' mais pas parmi ses 3 premières lettres
   (e) figurant dans /etc, et dont l'une des trois dernières lettres est un caractère 'e'
   (f) qui contiennent au moins un chiffre, au niveau 1
   (g) qui sont dans /etc ou ses descendants immédiats, et ne débutent pas par une lettre majuscule

4 Redirections

4.1 La commande find

Forme générale : `find␣répertoires␣prédicats`

Elle parcourt récursivement l'aborescence à partir d'un ou plusieurs répertoires spécifiés et affiche les fichiers qui concordent avec le ou les `prédicats`.

Par exemple `find␣/etc␣-type␣d` listera les entrées de type d = directoy = répertoire à partir de `/etc`.

1. Modifier la ligne précédente pour que la sortie standard de find termine dans un fichier `/tmp/sortie`. Vérifier que c'est bien correct en faisant cat `/tmp/sortie`.
2. Compléter la ligne de commande précédente pour que la sortie d'erreur standard termine dans un fichier `/tmp/erreurs`. Vérifier le fichier produit.
3. Modifier la ligne précédente pour que les deux sorties standard et d'erreur standard terminent un même fichier `/tmp/tout`. Vérifier le fichier produit là encore.

4.2 La commande wc

La commande `wc` compte combien de lignes, mots, et caractères contient son entrée standard. Elle ne peut terminer que lorsque l'entrée standard ne peut plus être alimentée, autrement dit, lorsqu'elle est fermée ; ce qui se fait par la combinaison de touches Ctrl+D lorsqu'elle est rattachée à un terminal.

1. Commencez par tester wc seule dans un terminal, en tapant quelques lignes/mots.
2. Faites maintenant en sorte que `wc` opère sur le fichier `/tmp/sortie` que vous avez produit à l'exercice précédent.

4.3 La commande cat

La commande cat concatène le contenu du ou des fichiers passés en arguments vers sa sortie standard.

1. Essayez cat `/etc/passwd`, puis `cat␣/etc/group`, et enfin `cat␣/etc/passwd␣/etc/group` pour vous en convaincre.
2. Concaténer les fichiers `tmp/sortie` et `/tmp/erreurs` en un seul fichier `/tmp/toto`. Les tailles des fichiers `/tmp/tout` et `/tmp/toto` sont-elles différentes ? Et leur contenu ?

3. Refaites la question 2 de l'exercice 4.2 en utilisant cat et une autre méthode de redirection.
4. Reproduire à nouveau le même résultat, mais sans utiliser de fichier /tmp/sortie

5 Variables ordinaires et d'environnement

5.1 Variable ordinaire

Supposez que `fic` contienne un nom de fichier avec chemin absolu – pour le seul besoin de l'exemple, disons `/etc/libssh/libssh_client.config`

En utilisant echo, faites afficher :

1. Le chemin seul de fic, terminé par un '/' (`/etc/libssh/` dans l'exemple). Cela fonctionne t'il toujours avec un fichier situé à la racine ?
2. Le nom du fichier seul (`libssh_client.config` dans l'exemple)
3. L'extension du fichier seule (`.config` dans l'exemple). Obtenez-vous bien une extension vide si le fichier n'en a pas ?

5.2 Variable d'environnement PATH

Dans le cours, nous avons utilisé deux programmes `args1` et `args2`, que nous avons toujours lancés à partir du répertoire courant, ce qui nécessite de les faire précéder soit par le chemin absolu, soit par ./

1. Créer un répertoire bin rattaché directement à votre répertoire par défaut (HOME)
2. Placez-vous dans bin, téléchargez les deux programmes `args1.c` et `args2.c` et recompilez-les, comme cela est fait dans le cours
3. Affichez le contenu de votre variable PATH. Que devez-vous lui ajouter pour que le système aille aussi examiner votre répertoire bin après tous les autres ?
4. Vérifiez que votre solution est bien opérationnelle : on doit pouvoir lancer `args1` et `args2` depuis n'importe quel répertoire, sans avoir à préciser de chemin.
5. Exécutez une ligne de commande qui ajoute la ligne précédente à votre fichier `~/.bashrc` En ouvrant un nouveau terminal, vous devriez obtenir la variable PATH étendue, et non celle d'origine.