

# Systeme d'Exploitation

## Travaux Dirigés (1), Licence 2 Informatique

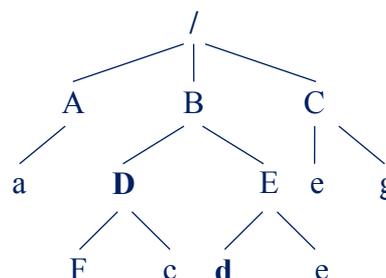
### Fichiers et Droits Utilisateurs

#### 1. Chemins absolus et relatifs

a. Rappeler les notions de chemin absolu et de chemin relatif d'un fichier. Comment reconnaître qu'un fichier est absolu ou relatif ?

b. Pour chaque fichier (ou répertoire), il existe toujours plusieurs chemins relatifs de ce fichier à partir d'un même répertoire courant. Vrai ou faux ?

Soit l'arborescence ci-contre où les lettres majuscules correspondent à des répertoires et les minuscules à des fichiers. Si D est le répertoire courant, exhiber si possible plusieurs chemins relatifs du fichier d.



c. Pour chaque fichier (ou répertoire), il existe toujours un seul chemin absolu. Vrai ou faux ? Donner un exemple d'arborescence et de chemin(s) qui justifient la réponse.

d. Supposons que le répertoire de travail est celui de la racine. Pour chaque fichier, son chemin relatif est donc donné par rapport à la racine. Ainsi, dans ce cas particulier, le chemin relatif est identique au chemin absolu. Vrai ou faux ? Donner un exemple pour justifier la réponse.

e. Choisir un répertoire de l'arborescence donnée dans la question b en tant que répertoire de travail. Donner un/des chemins relatifs de ce même répertoire.

#### 2. Liens symboliques et physiques

L'exemple de session Linux joint en annexe manipule une arborescence de fichiers concernant une matière d'enseignement (avec une séparation en cours, TD, TP, etc.). Analysez cette session afin de comprendre le sens et l'utilité des commandes utilisées et répondez aux questions suivantes.

a. Rappeler ce qu'est un inode. Quelle commande shell permet de connaître l'inode d'un fichier ou d'un répertoire ?

b. Rappeler la différence entre un lien symbolique et un lien physique. Quelle commande shell permet de reconnaître des liens durs et symboliques dans une arborescence de fichiers ?

c. Un lien physique vers un fichier ou répertoire F est une entrée d'un répertoire quelconque qui **pointe** vers l'inode de F. Dans l'exemple de la session donnée en annexe, expliquer pourquoi le répertoire TP possède 4 liens alors qu'il n'a qu'un seul répertoire père.

d. Supposons que deux utilisateurs U1 et U2 partagent le même fichier F à travers un lien dur. L'utilisateur U1 n'ayant plus besoin de F le supprime de son répertoire (en supposons qu'il ait le droit nécessaire pour le faire). L'utilisateur U2 a-t-il encore la possibilité d'utiliser F ? Pourquoi ?

e. Même question que plus haut, mais cette fois-ci U1 accède à F par un lien dur, tandis que U2 y accède par un lien symbolique.

f. Même question mais cette fois-ci U1 accède à F par un lien symbolique, tandis que U2 y accède par un lien dur.

g. Quel est l'intérêt des liens ? Quand est-il plus intéressant d'utiliser un lien dur plutôt qu'un lien symbolique et inversement ?

### 3. Droits sur les fichiers et répertoires

Lors de la création d'un fichier ou d'un répertoire par des programmes ou utilitaires, les droits souhaités (demandés) sont généralement `rw-rw-rw-` (666) pour un fichier et `rxwxrwxrwx` (777) pour un répertoire. Ces valeurs sont contrôlées par un masque, lui-même modifiable par la commande `umask`. Cette commande prend comme paramètre une valeur octale qui sera soustraite aux droits d'accès souhaités.

ATTENTION: Le calcul des droits définitifs (effectifs) n'est pas une simple soustraction de valeurs octales. Le masque retire des droits mais n'en ajoute pas.

- a. En prenant en considération ce qui précède, quel seront les droits d'un fichier créé dans un système où le `umask` est égal à 056 ?

Considérons la situation ci-dessous et répondez aux questions :

```
[pierre@home2 Test]$ ls -l
total 24
drwxrwxr-x 2 pierre pierre 4096 Sep 22 10:58 Temo
-rw-rw-r-- 1 pierre pierre  14 Sep 20 21:15 titi
-rw-rw-r-- 1 pierre pierre   69 Sep 20 21:18 toto
```

- b. En sachant qu'aucun changement de droit particulier n'a été fait pour les fichiers du répertoire `Test`, quelle est la valeur de `umask` ?
- c. Proposer une valeur de `umask` pour qu'à l'avenir les fichiers et répertoires à créer soient associés aux droits suivants: nous voulons supprimer pour les groupes le droit d'écriture sur les fichiers et sur les répertoires. Comment faire le changement ? Quelle sera la valeur numérique correspondant aux droits des fichiers et des répertoires ?
- d. Que se passe-t-il si nous décidons de mettre la valeur de `umask` à 0 ?

Soit l'exemple de session linux suivant:

```
[pierre@home2 Teste]$ ls -l
total 16
drwxrwxr-x 2 pierre pierre 4096 Sep 18 23:17 OS
-rw-rw-r-- 1 pierre pierre   0 Sep 18 23:17 titi
-rw-rw-r-- 1 pierre pierre   0 Sep 18 23:17 toto
```

- e. Expliquez ce que fait chacune des commandes ci-dessous, dans l'ordre présenté:
- ```
chmod 400 *
chmod a+x *
touch OS/tata
```
- f. Quelle est la version numérique de la commande `chmod u+x toto.txt` ?
- g. Quelle est la version symbolique de la commande `chmod 700 toto.txt` ?

## 4. Les droits d'endossement

Soient les deux nœuds du système de fichiers, tels que décrits dans la suite par la commande `ls -l`, où `a.out` est un exécutable qui ouvre en lecture/écriture (`O_RDWR`) le fichier `Donnees` :

```
-rwxrwxr-x    1 francois  groupe1    11687 déc  2 14:12 a.out
----rw-r--    1 francois  groupe1         44 déc  2 14:09 Donnees
```

On supposera que `francois` (respectivement `pierre`) appartient exclusivement au groupe `groupe1` (respectivement `groupe2`).

a. L'exécutable `a.out` peut-il ouvrir le fichier `Donnees` s'il est exécuté par :

1. `francois` ?
2. `Pierre` ?

On change le propriétaire et le groupe propriétaire du fichier `Donnees` ainsi :

```
-rwxrwxr-x    1 francois  groupe1    11687 déc  2 14:12 a.out
----rw-r--    1 pierre   groupe2         44 déc  2 14:09 Donnees
```

b. Idem question a.

Maintenant, on positionne le `set-uid` bit de l'exécutable `a.out` ainsi :

```
-rwsrwxr-x    1 francois  groupe1    11687 déc  2 14:12 a.out
----rw-r--    1 pierre   groupe2         44 déc  2 14:09 Donnees
```

c. Idem question a.

## 5. Le sticky bit

Soit l'exemple de session linux suivant :

```
[pierre@ufr0039 ~]$ ls -ld PartageLicence
drwxrwxrwx 2 pierre pierre 4096 sep  8 17:54 PartageLicence
[pierre@ufr0039 ~]$ ls -l PartageLicence
total 24
-rw-rw-r-- 1 bob      bob        83 sep  8 17:51 file1
-rw-r--r-- 1 paul     paul       813 sep  8 17:53 file2
-rw-rw-r-- 1 pierre   pierre     488 sep  8 17:56 file3
```

a. En supposant que tous les utilisateurs peuvent accéder au répertoire "`PartageLicence`", quels sont ceux (parmi `pierre`, `bob` et `paul`) ayant le droit de supprimer le fichier :

1. `file1` ?
2. `file2` ?
3. `file3` ?

Supposons maintenant que le `sticky` bit du répertoire "`PartageLicence`" est positionné, tel que montré ci-dessous :

```
[pierre@iut0039 ~]$ ls -ld PartageLicence
drwxrwxrwt 2 pierre pierre 4096 sep  8 17:54 PartageLicence
```

b. Répondez à la même question que précédemment.

## ANNEXE

```
[pierre@iut0039 CoursSE]$ pwd
/home/cours/pierre/CoursSE
[pierre@iut0039 CoursSE]$ ls -lia
total 56
16581882 drwxr-x--x  7 pierre Enseignants 4096 jan 11 15:19 .
16580611 drwx--x--x 25 pierre Enseignants 4096 jan 12 10:44 ..
16581873 drwxr-xr-x  2 pierre Enseignants 4096 jan 11 16:05 cours
16581876 drwxr-xr-x  2 pierre Enseignants 4096 jan 11 16:04 doc-linux
16581877 drwxr-x---  2 pierre Enseignants 4096 jan 11 15:17 notes-étudiants
16581875 drwxr-xr-x  2 pierre Enseignants 4096 jan 11 15:19 TD
16581874 drwxr-xr-x  4 pierre Enseignants 4096 jan 11 16:04 TP
[pierre@iut0039 CoursSE]$ ls -ial cours/
total 1236
16581873 drwxr-xr-x  2 pierre Enseignants  4096 jan 11 16:05 .
16581882 drwxr-x--x  7 pierre Enseignants  4096 jan 11 15:19 ..
16581878 -rw-r--r--  1 pierre Enseignants 329033 jan 11 10:56 cours1
16581879 -rw-r--r--  1 pierre Enseignants 849515 jan 11 10:56 cours2
16581880 -rw-r--r--  1 pierre Enseignants  37967 jan 11 10:56 cours3
16581899 lrwxrwxrwx  1 pierre Enseignants   13 jan 11 15:07 doc -> ../doc-linux/
[pierre@iut0039 CoursSE]$ ls -ali doc-linux/
total 108
16581876 drwxr-xr-x  2 pierre Enseignants  4096 jan 11 16:04 .
16581882 drwxr-x--x  7 pierre Enseignants  4096 jan 11 15:19 ..
16581893 -r--r--r--  1 pierre Enseignants 57108 jan 11 15:01 doc-NFS
16581895 -r--r--r--  1 pierre Enseignants 20880 jan 11 15:03 doc-sys
[pierre@iut0039 CoursSE]$ ls -ila notes-étudiants/
total 88
16581877 drwxr-x---  2 pierre Enseignants  4096 jan 11 15:17 .
16581882 drwxr-x--x  7 pierre Enseignants  4096 jan 11 15:19 ..
16581897 -rw-r-----  1 pierre Enseignants 15732 jan 11 15:05 notesDS
16581898 -rw-r-----  2 pierre Enseignants 48119 jan 11 15:06 notesTP
[pierre@iut0039 CoursSE]$ cd TD
[pierre@iut0039 TD]$ ls -lia
total 16
16581875 drwxr-xr-x  2 pierre Enseignants 4096 jan 11 15:19 .
16581882 drwxr-x--x  7 pierre Enseignants 4096 jan 11 15:19 ..
[pierre@iut0039 TD]$ cd ../TP
[pierre@iut0039 TP]$ ls -lia
total 84
16581874 drwxr-xr-x  4 pierre Enseignants  4096 jan 11 16:04 .
16581882 drwxr-x--x  7 pierre Enseignants  4096 jan 11 15:19 ..
16581898 -rw-r-----  2 pierre Enseignants 48119 jan 11 15:06 notes
16581884 drwxr-xr-x  2 pierre Enseignants  4096 jan 11 15:52 TP1
16581885 drwxr-xr-x  2 pierre Enseignants  4096 jan 11 15:53 TP2
[pierre@iut0039 TP]$ ls -lia TP1 TP2
TP1:
total 80
16581884 drwxr-xr-x  2 pierre Enseignants  4096 jan 11 15:52 .
16581874 drwxr-xr-x  4 pierre Enseignants  4096 jan 11 16:04 ..
16581883 lrwxrwxrwx  1 pierre Enseignants   23 jan 11 15:52 man-SE ->
../../doc-linux/doc-sys
16581871 -rw-r-----  1 pierre Enseignants  9884 jan 11 14:53 TP1_réponse
16581220 -rw-r--r--  1 pierre Enseignants 40757 jan 11 14:53 TP1_sujet
```

```
TP2:
total 528
16581885 drwxr-xr-x  2 pierre Enseignants  4096 jan 11 15:53 .
16581874 drwxr-xr-x  4 pierre Enseignants  4096 jan 11 16:04 ..
16581891 lrwxrwxrwx  1 pierre Enseignants    23 jan 11 15:53 man-NFS ->
../../doc-linux/doc-NFS
16581887 -rw-r-----  1 pierre Enseignants 324902 jan 11 15:25 TP2_réponse
16581886 -rw-r--r--  1 pierre Enseignants 174552 jan 11 14:56 TP2_sujet
[pierre@iut0039 TP]$ cd ../cours/
[pierre@iut0039 cours]$ ls doc
doc-NFS  doc-sys
[pierre@iut0039 cours]$
```