
TD 7
semaine du 15/5/00
les fichiers, niveau utilisateur

1. Types de fichier

Il existe différents types de fichier sous UNIX, découvrez les en utilisant la commande `ls -l` dans votre répertoire et dans les répertoires `/dev` et `/usr/bin`.

Listez le répertoire `/dev`. Faites un `ps` et regardez la colonne TTY, que remarquez vous ?

Lancez une deuxième fenêtre `Nxterm`. Exécutez la commande `ttty`. Placez-vous dans la première fenêtre `Nxterm` et tapez une commande qui écrit dans l'autre fenêtre.

2. Droits d'accès fichiers, répertoires

Créez un fichier quelconque de nom `essai1` (par exemple commande `touch essai1`). Quels sont les droits par défaut du propriétaire, du groupe et du reste du monde. ?

Faites en sorte que le groupe ait un accès en lecture.

Rendez le fichier `essai1` exécutable pour le propriétaire, le groupe et le reste du monde.

Créez un répertoire `essaid` et vérifiez les droits par défaut.

Ce répertoire est un répertoire qui est destiné à recevoir des informations confidentielles. Vous ne voulez pas que les membres du groupe ni le reste du monde puissent lister les fichiers de ce répertoire ni en supprimer. Vérifiez les droits et modifiez les si nécessaire. Quelqu'un d'autre que vous peut-il renommer un fichier de ce répertoire ? En créer un ?

Créez un fichier `essai2` dans ce répertoire et donnez des droits de lecture au groupe et au reste du monde. Vérifiez le résultat, un membre du groupe peut-il lister les fichiers du répertoire, afficher le contenu d'un fichier de ce répertoire ou copier ce fichier sur son compte (faites l'essai avec un voisin) ?

Finalement le répertoire est tellement confidentiel que vous ne voulez pas que le groupe ou le reste du monde puisse y accéder. Modifier les droits en conséquence.

Créer un autre répertoire `essaid2` et donner des droits de lecture et des droits d'accès au répertoire mais aucun droit d'écriture au groupe et au reste du monde.

Créez un fichier `essai3` dans ce répertoire et donner des droits en écriture au fichier au groupe. Demandez à un membre du groupe d'essayer de modifier le fichier, puis d'effacer le fichier. Expliquez.

Finalement vous voulez que par défaut, tout nouveau fichier créé ait au maximum tous les droits pour le propriétaire, un droit de lecture et d'exécution pour le groupe et un droit d'exécution pour les autres. Quelle commande faut-il faire? Quels seront les droits pour les nouveaux fichiers texte? pour les nouveaux répertoires?

3. Liens sur un fichier

Créez un répertoire `essailn` et placez-y un fichier quelconque `essai`. Créer quatre liens différents `lessai1`, `lessai2`, `lessai3`, `lessai4` sur `essai` dans `essailn` (commande `ln`). Faites un `ls -il`, que remarquez vous ?

Changez les droits d'accès de `lessai1`, faites un `ls -l`, qu'en déduisez-vous ?

Quel est la taille de chacun des ces fichiers (utilisez la commande `du`), quel est l'espace occupé par le répertoire ?

Détruire `lessai3`, faites un `ls -il`, expliquez.

Détruire *essai*, faites un *ls -il*, expliquez.

Créez un lien *essai6* de votre répertoire de connexion vers *essailn/lessai2*.

Supprimez tous les droits pour vous, le groupe et les autres sur le répertoire *essailn*.

Essayez d'afficher le contenu de *essai6*, et de le modifier en écriture. Qu'en déduisez vous ?

A quoi peuvent servir les liens d'après vous ?

A l'aide de la commande *df*, visualiser les différents systèmes de fichier et leur point de montage.

Repérez votre système de fichier et essayer de faire un lien vers un fichier d'une autre partition. Expliquez.

Essayez à nouveau en utilisant l'option *-s* de *ln*. Faites un *ls -il*. Que remarquez vous. Expliquez.

4. Redirection des entrées et sorties standards.

Redirigez la sortie de la commande *ls -l* vers un fichier quelconque.

Concaténez à ce fichier le résultat de la commande *ps -ef*.

Recherchez dans ce fichier toutes les lignes contenant votre nom de login en redirigeant l'entrée.

On donne le code d'un mini-interpréteur (c'est une version un peu améliorée de celui écrit au td4) :

```
#include <stdio.h>
#include <string.h>

main(int argc, char* argv[], char* arge[]) {
    char command[128];
    char* sptr, * PROMPT = "? ";

    printf("%s", PROMPT);
    sptr=gets(command);
    while(strcmp(command,"exit") && (sptr != NULL)) {
        if ((sptr=strstr(command,"cd ")) == command)
            chdir(sptr+3);
        else
            system(command);
        printf("%s", PROMPT);
        sptr=gets(command);
    }
}
```

Chargez et compilez le fichier *inter.c* (c'est une version du code ci-dessus qui utilise *fgets* au lieu de *gets*). Créez un fichier de commandes et faites le exécuter par votre mini-interpréteur *inter*.

Ajoutez une commande spécifique à *inter* (par exemple *interprete <fichier>*) pour exécuter toutes les commandes d'un fichier sans lancer un processus spécial pour cela.