

# Système de gestion de fichiers

## Appels Système

### Manipulation de répertoire

`opendir, readdir, closedir, stat`

## 1 `ls -l`

Ecrire une commande `ll` qui affiche sur la sortie standard le détail du contenu d'un répertoire passé en argument.

Le format de sortie doit être le suivant :

inode	[droit]	iud	taille	date	nom
64897	[750]	261	512	Oct 14 17:57	.
60213	[750]	261	512	Oct 14 17:50	..
64956	[750]	261	24576	Oct 14 17:57	a.out
64953	[640]	261	594	Oct 14 17:54	myls.c
64954	[640]	261	1355	Oct 14 17:52	myll.c

Utilisez les appels système Unix (`opendir`, `readdir`, `closedir`, `stat`, `ctime`). La fonction `ctime` dans `#include<sys/time.h>` traduit une date en chaîne de caractères.

Ou en Python `os.listdir, stat` et `time.ctime` pour la conversion de la date. Il faudra en Python ajouter à la main `.` et `..` dans la liste retournée par `os.listdir()` avant de faire le `os.stat`

Pour vous aider reprenez le code des exemples vus en cours du `ls.c` ou `ls.py` et `stat.c` ou `mstat.py`.

## 2 ls -l (version complete)

```
total 24
drwxr-x---  2 courtrai prof      4096 Mar  4  1998 ./
drwxr-x---  3 courtrai prof      4096 Nov 16  1998 ../
-rw-r----- 1 courtrai prof     1241 Dec 10  1997 find.c
-rw-r----- 1 courtrai prof     2125 Dec 17  1997 myll.c
-rw-r----- 1 courtrai prof       568 Dec 10  1997 myls.c
-rw-r----- 1 courtrai prof       619 Dec 10  1997 mylsDir.c
```

Le premier nombre sur chaque ligne est le nombre de liens matériels sur l'entrée.

Pour cet exercice, utilisez les fonctions `getpwuid` `getgrgid` pour lire le fichier `/etc/passwd` et `/etc/group`

En Python `pwd.getpwuid` et `grp.getgrgid`

## 3 ls -lR

la version récursive du `ls -l`