

TP3

Contact:

Marios Fanourakis - marios.fanourakis@unige.ch

Rose Defossez - rose.defossez@etu.unige.ch

Hugo Haldi - hugo.haldi@etu.unige.ch

October 2021

1 Objectifs

L'objectif général du TP est de créer un programme qui permet de faire des backups d'architecture de fichiers facilement, en s'inspirant du programme *rsync*.

Les objectifs pédagogiques sont de:

- manipuler des fichiers;
- obtenir et modifier les attributs d'un inode;
- manipuler des liens;
- parcourir des architectures de dossiers

Ce TP se fera sur deux séances. Dans le cas où vous terminez la première séance en avance nous vous recommandons fortement de continuer directement sur la deuxième qui est un peu plus longue.

2 Le programme *rsync*

Le programme *rsync* permet de faire des copies d'architectures de fichiers. Il est très utilisé notamment pour faire des backups incrémentaux à travers le réseau. En effet *rsync* peut également faire des copies d'architectures de fichiers à distance. Dans ce TP nous nous limiterons à la création d'un programme qui permettra de:

- lister les fichiers d'un dossier de manière récursive;
- copier le contenu d'un dossier dans un autre de manière récursive.

Il s'agit donc d'un *rsync* très simplifié et sans copies à distance. Nous vous recommandons toutefois de lire rapidement la documentation de *rsync* car ce programme pourra vous être utile dans le future.

Le programme que nous développerons s'appellera *ultra-cp*.

3 Listing de fichiers

Cette section fera l'objet du premier TP. Il s'agit ici de lister le contenu complet d'un dossier. Ainsi TOUS les fichiers, dossiers et sous-dossiers doivent être listés.

Cette liste sera effectuée lorsque le programme sera appelé avec un seul argument. Si cet argument est un fichier, uniquement le fichier est listé, si cet un dossier alors le programme se comportera comme mentionné ci-dessus.

Voici un exemple de sortie du programme:

```

1 > ultra-cp ../../TP5
2 drwxrwxr-x      4096 Mon Nov  9 10:09:47 2015 ../../TP5
3 drwxrwxr-x      4096 Thu Nov 19 14:56:59 2015 ../../TP5/doc
4 -rw-rw-r--      928 Thu Nov 19 14:56:24 2015 ../../TP5/doc/TP5.aux
5 -rw-rw-r--      152 Thu Nov 19 14:56:24 2015 ../../TP5/doc/TP5.out
6 -rw-rw-r--       85 Thu Nov 19 14:55:08 2015 ../../TP5/doc/Makefile
7 -rw-rw-r--    143314 Thu Nov 19 14:56:24 2015 ../../TP5/doc/TP5.pdf
8 -rw-rw-r--      6150 Thu Nov 19 14:56:59 2015 ../../TP5/doc/TP5.tex
9 -rw-rw-r--      6151 Thu Nov 19 14:55:48 2015 ../../TP5/doc/TP5.tex~
10 -rw-rw-r--     25261 Thu Nov 19 14:56:24 2015 ../../TP5/doc/TP5.log
11 drwxrwxr-x      4096 Thu Nov 19 14:17:51 2015 ../../TP5/src
12 -rw-rw-r--     8480 Thu Nov 19 14:02:33 2015 ../../TP5/src/main.o
13 -rw-rw-r--      358 Thu Nov 19 13:30:48 2015 ../../TP5/src/Makefile
14 -rw-rw-r--     1180 Thu Nov 19 11:11:29 2015 ../../TP5/src/options.h
15 -rw-rw-r--    10250 Thu Nov 19 14:17:27 2015 ../../TP5/src/proc_entries.c
16 -rw-rw-r--     1141 Thu Nov 19 14:14:04 2015 ../../TP5/src/proc_entries.h
17 -rw-rw-r--       395 Thu Nov 19 13:28:18 2015 ../../TP5/src/err.c
18 -rw-rw-r--       365 Thu Nov 19 13:28:55 2015 ../../TP5/src/err.h
19 -rw-rw-r--     6520 Thu Nov 19 14:01:27 2015 ../../TP5/src/err.o
20 -rw-rw-r--     2859 Thu Nov 19 14:02:30 2015 ../../TP5/src/main.c
21 -rw-rw-r--     7920 Thu Nov 19 14:01:27 2015 ../../TP5/src/options.o
22 -rw-rw-r--    14520 Thu Nov 19 14:17:51 2015 ../../TP5/src/proc_entries.o
23 -rw-rw-r--     2711 Thu Nov 19 13:24:45 2015 ../../TP5/src/options.c
24 -rwxrwxr-x     28259 Thu Nov 19 14:17:51 2015 ../../TP5/src/ultra-cp

```

La sortie est donc similaire à la fonction `ls` et elle contient:

- le type de fichier: `d` pour un dossier, `-` pour un fichier régulier, `l` pour un lien; d'autre peuvent être implémentés;
- les droits en lecture/écriture/exécution: les droits d'accès ne concernent que les droits "basiques" mentionnés; les bits `setuid/setgid`, sticky bit et autres ne seront pas représentés;
- la taille du fichier en octets;
- la date de modification: voir la fonction `strftime`;
- le nom du fichier: donné relativement au dossier/fichier passé en paramètre.

La plus part des informations sont disponibles par les commandes `lsstat/stat/fstat` vues en cours.

4 Backup de dossiers et fichiers

4.1 Utilisation

Cette section fait l'objet du deuxième TP. En se basant sur le code développé précédemment vous devrez désormais faire une copie récursive de l'architecture des dossiers/fichiers. Cela sera le cas lorsque `ultra-cp` sera utilisé avec au moins deux paramètres: des sources et une unique destination. Si la destination est un dossier existant, chaque source spécifiée sera recrée dans la destination.

Ainsi, la commande:

```
1 ultra-cp file1 folder1 folder2/ destination
```

aura pour effet de créer l'architecture suivante dans destination:

1	drwxrwxr-x	4096	Thu	Nov	19	15:09:28	2015	destination
2	drwxrwxr-x	4096	Thu	Nov	19	15:09:58	2015	destination/folder1
3	drwxrwxr-x	4096	Thu	Nov	19	15:09:58	2015	destination/folder1/emptydir
4	-rw-rw-r--	0	Thu	Nov	19	15:09:50	2015	destination/folder1/f2
5	-rw-rw-r--	0	Thu	Nov	19	15:09:48	2015	destination/folder1/f1
6	drwxrwxr-x	4096	Thu	Nov	19	15:10:06	2015	destination/folder2
7	-rw-rw-r--	0	Thu	Nov	19	15:10:04	2015	destination/folder2/f5
8	-rw-rw-r--	0	Thu	Nov	19	15:10:06	2015	destination/folder2/f4
9	-rw-rw-r--	14	Thu	Nov	19	15:09:21	2015	destination/file1

Si le programme est appelé avec deux fichiers dans ce cas le fichier source sera copié vers le fichier de destination. Si la destination n'existe pas alors elle sera considérée comme un fichier à créer (le programme fera alors une erreur dans le cas où il y a plusieurs sources).

Uniquement les fichiers réguliers, les liens et les dossiers devront être copiés (i.e. pas les sockets, block devices, etc.).

Les fichiers et dossiers créés/modifiés auront les mêmes droits (si possible) que les sources.

Question: dans quel cas les droits ne pourrions pas être les mêmes ?

4.2 Modification de fichiers existants

Le programme pourra remplacer des fichiers existants mais ne devra jamais supprimer des fichiers/dossiers. Donc si un fichier est présent dans la destination mais pas dans la source il ne devra PAS être supprimé de la destination.

Si le fichier/dossier de destination existe déjà il devra être remplacé UNIQUEMENT si la taille de la source et de la destination sont différentes ou si la date de modification de la source est plus récente que celle de la destination.

Si l'option '-a' est passée en paramètre par l'utilisateur, les droits des fichiers/dossiers devront être modifiés même si le fichier/dossier n'a pas été remplacé.

4.3 Gestion des liens

Par défaut les liens seront déréférencés (i.e. transformés en fichiers et dossiers lors de la copie), sauf si l'utilisateur a passé l'option '-f' en paramètre du programme. Dans ce cas les liens seront copiés en tant que lien. De plus les liens de destination seront ajustés pour garantir qu'ils pointent bien vers le même inode que le lien source (indice: éviter les chemins relatifs, la fonction *realpath* sera utile pour cela).