PROJECT Nº1

INTRODUCTION

Backups protect file systems from user errors, disk or other hardware failures, software errors that may corrupt the file system and natural disasters. The most common uses of backups are to restore files accidentally deleted by users and to recover from disk failures.

The simplest way to protect a file system against disk failures or file corruption is to copy the entire contents of the file system to a backup device. The resulting archive is called a full backup. If a file system is later lost due to a disk failure, it can be reconstructed from the full backup onto a replacement disk. Individual lost files can also be retrieved. Full backups have two disadvantages: reading and writing the entire file system is slow, and storing a copy of the file system consumes significant capacity on the backup medium.

Faster and smaller backups can be achieved using an incremental backup scheme, which copies only those files that have been created or modified since a previous backup. Incremental schemes reduce the size of backups, since only a small percentage of files change on a given period of time. A typical incremental scheme performs occasional full backups supplemented by frequent incremental backups. Restoring a deleted file in an incremental backup system may require consulting a chain of backup files, beginning with the last full backup and applying changes recorded in one or more incremental backups.

In this project, a variant of the usual backup/restore scheme is proposed. For simplification, the files are backed up in the same disk, which makes this scheme vulnerable to file system corruption and other disasters. However, it may still be useful in many situations, when it is necessary to recover an old file.

During the development, students will have the opportunity to practice with file and directory manipulations, process management and signal handling.

REQUIREMENTS

In this project, two applications must be developed:

- Backup - This application keeps a backup of the regular files of a specified directory. It starts by doing a full backup and after that, at specified intervals, checks for files that have been created or modified and makes an incremental backup of those files.
- Restore – Using this application it must be possible to view the changes that have been made to the files in the directory and to restore a given file or the whole directory, at a specified data.

More specific details about each application are given in the following. An illustration is presented at the end of this document.

BACKUP PROGRAM

- This program is launched through the command **`bckp dir1 dir2 dt &`**
  where
  - **`dir1`** is the name of the directory to be monitored;
  - **`dir2`** is the directory where are saved the files in **`dir1`** that have been modified/created or any other information necessary to restore a previous state of **`dir1`**;
  - **`dt`** is the interval between two consecutive scannings of **`dir1`**.
- Only regular files must be monitored; subdirectories, hard links and symbolic links must not be taken into account.

- At the beginning, the program must do a full backup of **dir1**; then, do scannings at **dt** intervals; whenever a creation/modification/deletion of any file is detected, the list of existing files at the moment of the scanning must be stored in a text file, named **__bckpinfo__**, in a subdirectory of **dir2**, as specified in the following point; if any file modification/creation is detected, an incremental backup must be done. The file **__bckpinfo__** may contain other information about the files that may be needed by the backup program. The modified files must be enterily saved.
- All the data related with backups (the files of the full backup, the list of existing files after file creation/modification/deletion, the modified/created files of the incremental backup, or other necessary information) must be saved in a subdirectory of **dir2**, the name of which is formed, taking into account the date/time at which the backup occurred, as follows: **year_month_day_hours_minutes_seconds** (example: **2013_03_15_12_30_00**).
- A different process must be launched for each file in **dir1** that needs to be copied to **dir2**.
- The **bckp** process ends when the user sends it **SIGUSR1** signal. It must be guaranteed that any process that was created by **bckp** completes its task before ending.

RESTORE PROGRAM

- This program is launched through the command **rstr dir2 dir3**
  where
  - **dir2** is the directory that was used to backup the files;
  - **dir3** is the directory where the backed up files are going to be restored.
- The program must allow the restoring of all the files contained in **dir1** in a specified date/time, using the full/incremental backups. For this, a list of all the times when the contents of **dir1** was modified (files were created/modified/deleted) must be shown to the user, so that he/she chooses the instance of **dir1** to be restored.
- A different process must be launched for each file in **dir2** that needs to be copied to **dir3**.
- An efficient restore algorithm will be appreciated.

GRADING

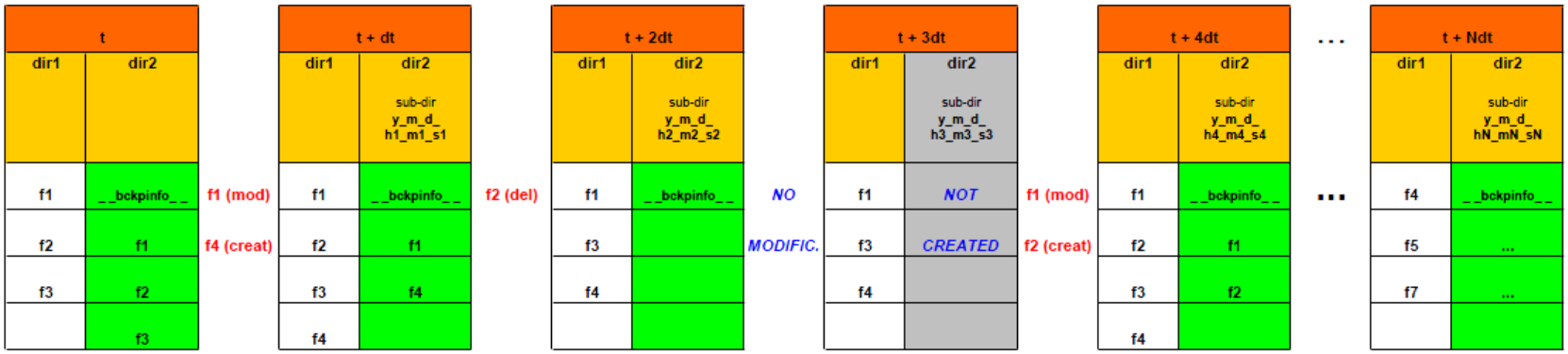
The grading for this project will take into account several criteria, namely:

- Code readability in general, namely, identifier names, indentation, ...
- Comments: function header comments and in-line comments.
- Adequate use of data structures.
- Code structure and modularity.
- Input validation.
- Operation of the applications according with the requirements.

SUBMISSION

- Create a directory named **TxGyy** (where **x** and **yy** represent, respectively, the number of the class/*turma* and the team number (for example, T2G07, for team 7 of class 2).
- Create two sub-directories, named **backup** and **restore**. Copy to each folder the source code of the corresponding program (**only the files with extensions .c** or **.h**) and the **makefiles**.
- Compact the files into a file named **TxGyy.tar** and submit the file through the link available at the course page, at Moodle/FEUP.
- **Each working group must submit only one project.** The team may resubmit the project several times, but only the last submitted version will be graded and will be used to determine if the project is late.
- **Submission must be done before 2013/04/21, at 23:55.**

BACKUP - command > bckp dir1 dir2 dt &



__bckpinfo__
f1
f2
f3

__bckpinfo__
f1
f2
f3
f4

__bckpinfo__
f1
f3
f4

__bckpinfo__
f1
f2
f3
f4

__bckpinfo__
f4
f5
f7

NOTE: other info is necessary, beyond the filenames

RESTORE - command > rstr dir2 dir3

List of available restore points (in the format yyyy_mm_dd_hh_mm_ss, not t+Nt):

t
t+dt
t+2dt
t+4dt NOTE: not "t+3dt"
...

Which restore point (time) ? t+4dt

=> restore f1 (at t+4dt)
restore f2 (at t+4dt)
restore f3 (at t)
restore f4 (at t+dt)