B2- Synthesis Pool

B-ADM-144

automakefile

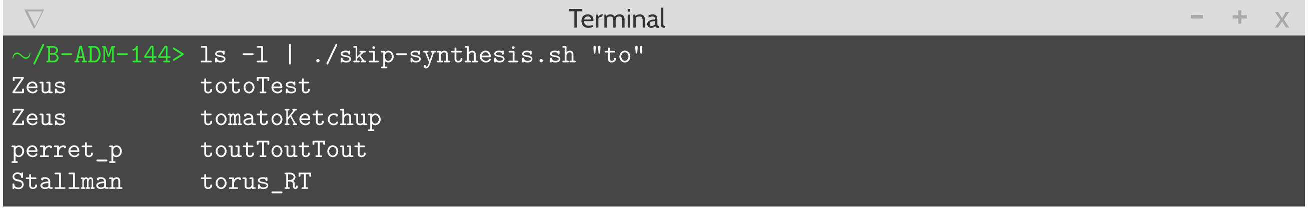
automake-like, but way better

v1.3

Requirement

skip-synthesis.sh

Write a shell script named *skip-synthesis.sh* that will both read the output of the command *ls -l*, and print the user/-name of the files starting with the string given as input.



```
~/B-ADM-144> ls -l | ./skip-synthesis.sh "to"
Zeus            totoTest
Zeus            tomatoKetchup
perret_p        toutToutTout
Stallman        torus_RT
```

The rest of the project will not be corrected unless this requirement is fully functional (and rewritten).

The file must be placed at the root of your git repository.

automakefile

Makefile Generator

binary name:	automakefile
repository name:	ADM_automakefile_$YEAR
repository rights:	ramassage-tek
language:	bash
group size:	1
compilation:	none

- Your repository must contain the totality of your source files, but no useless files (binary, temp files, obj files,...).
- All the bonus files (including a potential specific Makefile) should be in a directory named *bonus*.
- Error messages have to be written on the error output, and the program should then exit with the 84 error code (0 if there is no error).

Makefile Generation

Write a shell script named *automakefile*, that will generate a nice Makefile given a configuration file.

This configuration file can contain the following (potentially unordered) lines:

- source_filename;dependence1 dependence2 ...(specify the full names of the files, from the header subfolder below)
- PROJECT_DIR;name_of_the_project_root_folder
- SOURCES_DIR;subfolder_containing_the_source_files
- HEADERS_DIR;subfolder_containing_the_header_files
- LIBS_DIR;subfolder_containing_librairies
- EXEC;executable_name
- CC;compilator_binary
- CFLAGS;compilation_flag1 compilation_flag1...
- LDFLAGS;linking_flag1 linking_flag2 ...

The generated Makefile must contain at least the classic `clean`, `fclean` and `re` rules

PROJECT_DIR is the only mandatory line, along with at least one source file.

Version Manager

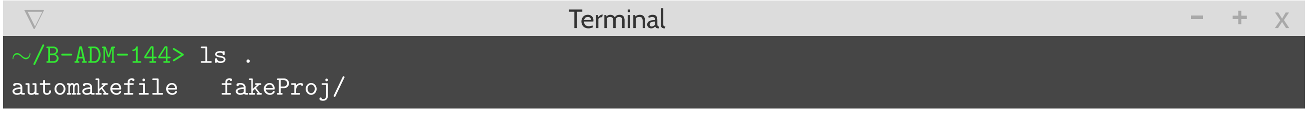
Add the following rules to the generated Makefile:

archive:	backup the source files in a compressed format, in the project folder
revert:	recreate the project from last backup
num:	print the current version number
delete:	delete last backup

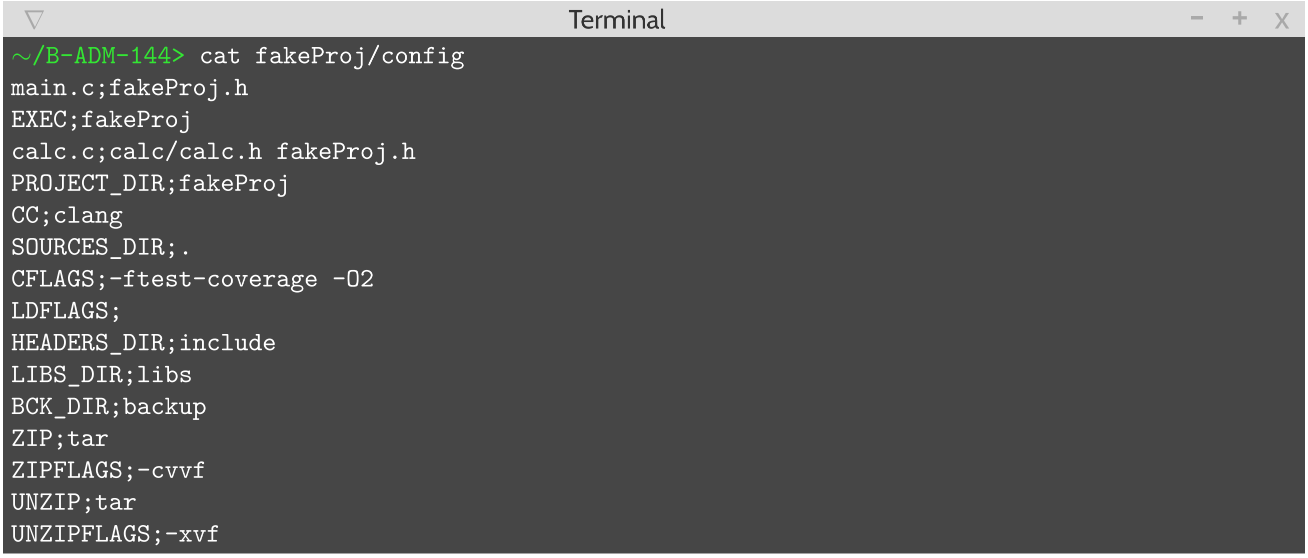
While archiving, the message *'xxx file generated.\n'* will be printed (*xxx* being the name of the archive).

5 more fields will be necessary in your configuration file :

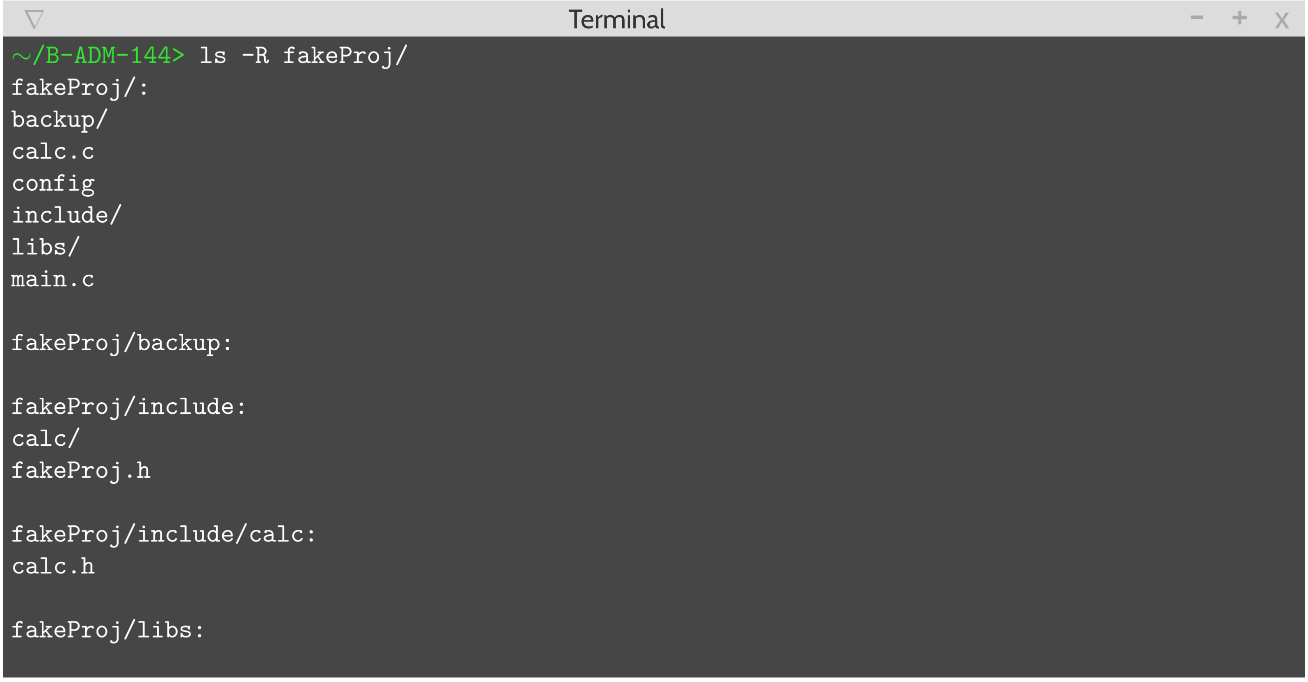
- BCK_DIR;name_of_the_backup_folder
- ZIP;compression_binary
- ZIPFLAGS;compression_options
- UNZIP;decompression_binary
- UNZIPFLAGS;decompression_options



```
~/B-ADM-144> ls .
automakefile   fakeProj/
```



```
~/B-ADM-144> cat fakeProj/config
main.c;fakeProj.h
EXEC;fakeProj
calc.c;calc/calc.h fakeProj.h
PROJECT_DIR;fakeProj
CC;clang
SOURCES_DIR;.
CFLAGS;-ftest-coverage -O2
LDFLAGS;
HEADERS_DIR;include
LIBS_DIR;libs
BCK_DIR;backup
ZIP;tar
ZIPFLAGS;-cvvf
UNZIP;tar
UNZIPFLAGS;-xvf
```



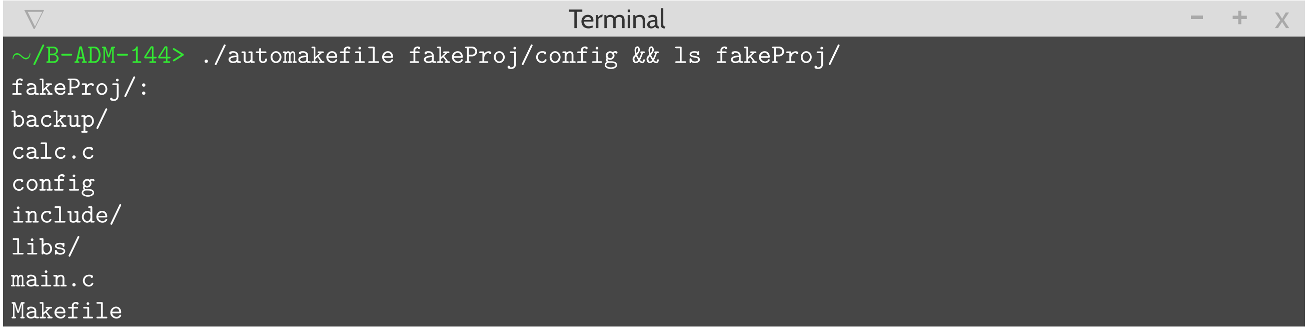
```
~/B-ADM-144> ls -R fakeProj/
fakeProj/:
backup/
calc.c
config
include/
libs/
main.c

fakeProj/backup:

fakeProj/include:
calc/
fakeProj.h

fakeProj/include/calc:
calc.h

fakeProj/libs:
```



```
~/B-ADM-144> ./automakefile fakeProj/config && ls fakeProj/
fakeProj/:
backup/
calc.c
config
include/
libs/
main.c
Makefile
```