

REPORT : ASSIGNMENT 1 : Makefile

By
Manoj Kumar
2018CS50411
Group – 4

In the Top directory, there are 4 subdirectories – 1,2,3,4.

Each sub-directory contains a simple C file and a corresponding Makefile. here are total 5 source code files, each in different directory.

Here is some description about each sub-directory:

./1/ : it consists of a simple square function of an integer in C and a makefile which has two targets-all and clean. 'All' target creates an object file (mySqr.o) in a subdirectory of Top named obj while 'clean' target removes that already created file.

./2/ : it consists of a simple addition function of two integers in C and a makefile which has two targets-all and clean. 'All' target creates an object file(myPythonInC.o) in a subdirectory of Top named obj while 'clean' target removes that already created file.

./3/ : it consists of a simple sumseries function of four integers in C and a makefile which has two targets-all and clean. This function uses the addition function. 'All' target creates an object file(sumseries.o) in a subdirectory of Top named obj while 'clean' target removes that already created file.

./4/ : it consists of a simple addition function of two integers in C and a makefile which has two targets-all and clean. 'All' target creates an object file(modulo.o) in a subdirectory of Top named obj while 'clean' target removes that already created file.

In Top Directory :

here is the main c file named myApp.c which keeps use of all these 4 files. It's a main function that calls above defined functions . It creates object file myApp.o in obj subdirectory.

COMMANDS TO FOLLOW:

To unzip the submission:

```
sudo apt-get install unzip  
unzip Top.zip
```

change directory:

```
cd Top
```

Sub-directories 1,2,3,4 are already created. Directory Top contains files myApp.c, Makefile, Report.pdf

compilation of the parent makefile:

```
make 'or' make all
```

It compiles the Makefile in Top directory as well as all Makefiles of sub-directories 1,2,3,4. It compile entire code and creates object files in obj sub directory and executable file in exe subdirectory.

Execution of executable file:

```
make RUN
```

execute executable file and prints the final answer on screen.

Output is like this:

Value of a is : 3

Manoj Kumar

value of a : 3

value of b : 9

value of c : 12
value of d : 24
sum of a,b,c and e : 28

Clean everything:

make clean

deletes all the object, executable files and subdirectories obj and exe.

Note - For printing final answer on the screen, follow these steps -

1. Go into Top directory, and open Terminal in the directory.
2. Type **make** or **make all** and press enter.
3. After that type **make RUN** and press enter, answer will be on your screen.
4. After that if you want to clear the object files then type **make clear** and press enter.

Detailed Explanation:

The main task was to create makefiles in each of the sub-directories and a main makefile which is stored in the parent directory.

The makefiles in the sub-directories check the dependencies of the .c file and recompiles if it detects any change in the .c file and it ensures that the .o file created is stored in the obj directory inside Top instead of the current one.

The description of the main makefile is given below:

->The all target checks the dependencies of directories and all the .o files which are stored in the obj sub-directory and creates a final named executable in the exe sub-directory inside Top directory.

->The directories target makes the directories exe and obj. the commands used for the purpose are `mkdir -p <dir_name>` instead of `mkdir <dir_name>` so that the 'directory already exists error' doesn't occur.

->The <name>.o targets check the dependencies and then changes the directory and recompiles accordingly.

->make clean: The clean target deletes all the contents of the exe and obj directory. There is no need to delete the directory here because we have already used -p command to make the directories.

->make RUN: The RUN target executes the final executable named final.

Makefile FORMAT >>

Target : Dependencies

Commands

Target refers to the function being executed.

Dependencies refers to the files in which any change taken place forces the command to execute.

Commands are the ones executed in linux once the target file is read.

The makefile in these directories does multiple tasks >>

1. The first task which is done whenever make command or make all command is written without even writing any specific command is the conversion of the high level language code file into object file which can be used by the linker to convert it into an executable file which can be loaded on the software with the help of a loader.

2. Whenever the command `make clean` is typed then it removes the object file made by the command `make` in the directory named `obj`.