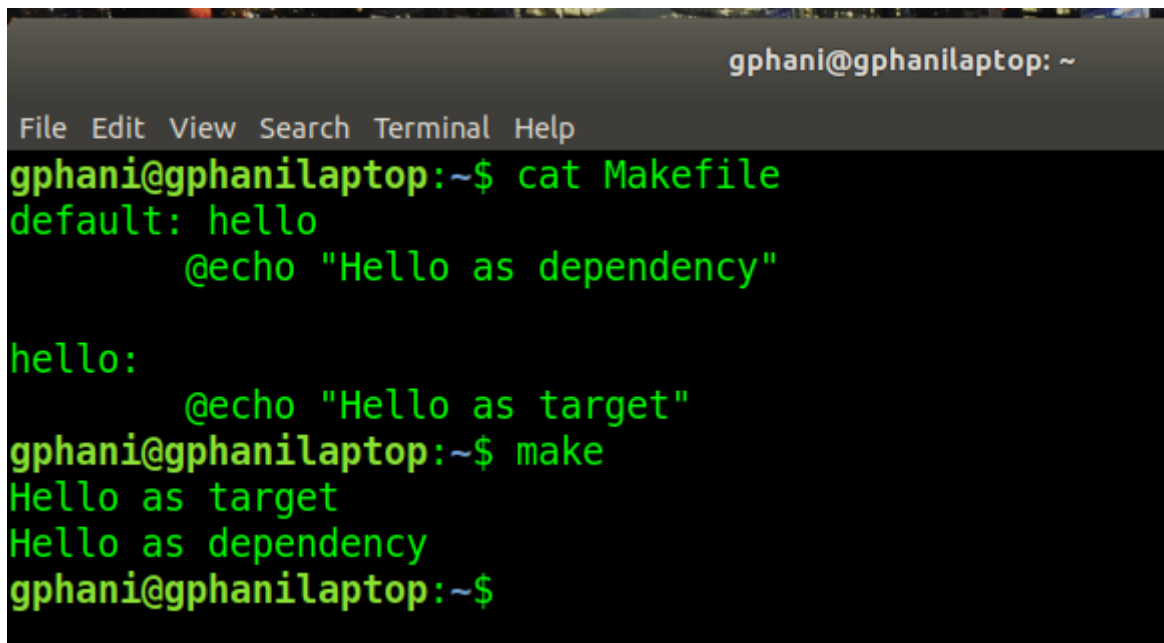


Session – 11 : The make command
30-08-2019, 08:00 – 09:00 Hrs, RJN 302

The projector was not switched on during this session. We did some discussion on the make command. Some comments based on the discussion are as follows.

1. Naming the file: the instructions for the make utility should be in the file called Makefile. Alternatively one can use `make -f filename` to force it to use instructions from another file.
2. The recipe for each target is to be given in lines that start with a tab. Sometimes your text editor may replace the tab character with say, 6 space characters. This will not work. Replace those to a tab using search-replace feature of vi editor.
3. The recipe can be in multiple lines. Each of these lines must start with a tab. Any line that skips the tab character after that implies that the recipe has completed.
4. Each command in the recipe will be first displayed to the screen before executing. This can be prevented by using a `@` character at the beginning of the command.
5. The colon character indicates the name of the target. What follows after the colon is a list of dependencies. If a file by the name in the list of dependencies is not found in the current directory, it will be deemed as a target. Make will look for instructions on making that target in the Makefile and go ahead to execute that recipe.

This is illustrated in the following example.



```
gphani@gphanilaptop: ~  
File Edit View Search Terminal Help  
gphani@gphanilaptop:~$ cat Makefile  
default: hello  
        @echo "Hello as dependency"  
  
hello:  
        @echo "Hello as target"  
gphani@gphanilaptop:~$ make  
Hello as target  
Hello as dependency  
gphani@gphanilaptop:~$
```

Now if we create an empty file called hello in the same folder, the output will be different. Once there is a file called hello in the folder, it is not looked as a target that needs to be made. So that section is not executed.

```
gphani@gphanilaptop: ~  
File Edit View Search Terminal Help  
gphani@gphanilaptop:~$ cat Makefile  
default: hello  
        @echo "Hello as dependency"  
  
hello:  
        @echo "Hello as target"  
gphani@gphanilaptop:~$ touch hello  
gphani@gphanilaptop:~$ ls -l hello  
-rw-r--r-- 1 gphani gphani 0 Sep 10 21:17 hello  
gphani@gphanilaptop:~$ make  
Hello as dependency  
gphani@gphanilaptop:~$
```

Homework:

- [1] Create a Makefile that uses your own bash shell scripts in each recipe.
- [2] Create a Makefile that can compile a code in three different ways namely (a) without any options, (b) with all options to provide warnings for non-compliance to ANSI standards etc., and (c) with options that help the final executable run fastest possible for the given architecture of the machine.