Introduction to Makefile

1. Makefile is basically a shell script, except for the following characteristics: it automatically detects the file changed and re-runs the commands with dependency on changed files.
2. Makefile is executed by “run” command within the directory containing Makefile.
3. Each command in Makefile is specified in the following format:

target_file: dependency_files

[tab]action lines

4. The target file can also be a command name instead of the filename of a target file.
5. The command is run by “make target_file” where target_file is the target file in the command of Makefile, then the action lines will be run.
6. If “make” is called, then the first command in Makefile is run.
7. When “make target_file” is called, then corresponding dependency_files is checked to see if any of them is changed. If any of dependency_files are changed and there is corresponding command with changed dependency file as target_file, then that command will be run prior to current command.
8. Due to check of change in dependency file, Makefile is useful for specifying compilation rules, but it is almost equivalent to shell script.
9. Makefile can also define variables just like in shell script. Wildcard matching is also used for general rules.
10. The “clean:” can be used as a command with corresponding actions, to remove intermediate compilation files. Such command will be called by “make clean”.