**Make-File Basics:**

* Automate compile process
* System program: *make*
  + A rule interpreter
  + Executes instructions to compile and link a program
  + Instructions are written into a Makefile
* Format of rules:

Target : prerequisites

Command

…

**Target:** a name of a file or program to be built or an action to be carried out.

**Command:** an instruction to be executed.

**Prerequisites:**  a list of files on which the target depends.

**Ex.**

1. Compiles a program consisting of a single source code file (hello.c) and no header files.

Hello : hello.c

gcc -o hello hello.c

2. Compiles a program consisting of two source code files (test.c, sort.c) and a header file (sort.h)

test : test.o sort.o

gcc -o test test.o sort.o

test.o : test.c sort.h

gcc -c test.c

sort.o : sort.c sort.h

gcc -c sort.c

clean :

rm test test.o sort.o

3. The previous Makefile with variables to reference all object files.

objects = test.o sort.o

test : $(objects)

gcc -o test $(objects)

test.o : test.c sort.h

gcc -c test.c

sort.o : sort.c sort.h

gcc -c sort.c

clean :

rm test $(objects)

4. Letting make deduce the commands using an implicit rule for updating .o files.

CFLAGS = -g -Wall

CC = gcc

objects = test.o sort.o

test : $(objects)

$(CC) -o test $(objects)

test.o : test.c sort.h

sort.o : sort.c sort.h

.PHONY : clean

clean :

rm test $(objects)