Make and Makefiles

GNU Make
http://www.gnu.org/software/make

Make Overview

- The make utility automatically determines which pieces of a program need to be recompiled, and issues commands to recompile them
- Make requires a makefile that describes the relationships between files in your program and provides commands for updating each file
- Once a suitable makefile exists, you only need to type ‘make’ or ‘make target’ on the command line in order to compile and link all files required for your program
- For more information, refer to the GNU make overview document (by Richard M. Stallman and Roland McGrath) available on the course web site. Parts of this presentation contain material from their GNU make overview document.
- Various make utilities are available for use in ECEN 5613. The free GNU make utility has many features, is widely used, and is standardized.

Makefile Structure

- A simple makefile consists of "rules" with the following structure:

  target ... : *prerequisites* ...

  command

  ...

- A **target** is usually the name of a file that is generated by a program; examples of targets are executable, object, or hex files. It can also be the name of an action to carry out, such as ‘clean'
- A **prerequisite** is a file that is used as input to create the target. A target often depends on several files.
- A **command** is an action that make carries out. A rule may have more than one command, each on its own line. **Note: for GNU make you need to put a tab character at the beginning of every command line!**
- Usually a command is in a rule with prerequisites and serves to create a target file if any of the prerequisites change. However, the rule that specifies commands for the target need not have prerequisites. For example, the rule containing the delete command associated with the target ‘clean' does not have prerequisites.
- A **rule**, then, explains how and when to remake certain files which are the targets of the particular rule. make carries out the commands on the prerequisites to create or update the target. A rule can also explain how and when to carry out an action.

Using Make

- GNU make processes only the specified/relevant targets in the makefile; if no target is specified, it processes the first target found in the makefile. You should place the default target first in your makefile.
- GNU make requires a tab character at the beginning of each command line.
- In the following examples, you would execute make as follows:
    - make
    - make all
    - make prog.hex

- Note: since the ‘all’ target is listed first in the makefile, it is the default target built when you type ‘make’ on the command line.

GNU Make Simple Example (SDCC)

```
################################################################################
# GNU makefile example for code compiled with SDCC
# File Name: Makefile or makefile or gnumakefile
# Only the targets which apply will be processed by GNU make.
# To create the file 'prog.hex' using GNU make,
#     execute 'make prog.hex', 'make all', or just 'make'.
# Note: GNU make requires a special format for the makefile:
#     A tab character must be at the beginning of every command line!
################################################################################

# Default target
all: prog.hex

# Compile phase
syntax.rel : syntax.c syntax.h
      sdcc -c -mmcs51 --std-sdcc99 --verbose --model-large syntax.c

# Link phase
prog.hex : syntax.rel
      sdcc --code-loc 0x0000 --code-size 0x8000 --xram-loc 0x0000 --xram-size
0x8000 --model-large --out-fmt-ihx syntax.rel

# Phony target
.PHONY: clean
clean:
      del *.rel *.lst *.rst *.hex *.mem *.map *.sym *.lnk *.ihx
```

GNU Make More Complex Example (SDCC)

```
################################################################################
# GNU makefile example for code compiled with SDCC
# File Name: Makefile or makefile or gnumakefile
# Only the targets which apply will be processed by GNU make.
# To create the file 'prog.hex' using GNU make,
#            execute 'make prog.hex', 'make all', or just 'make'.
# Note: GNU make requires a special format for the makefile:
#     A tab character must be at the beginning of every command line!
################################################################################

# Usually SDCC's large memory model is the best choice for ECEN 5613.
MEMORYMODEL = --model-large

# These settings control how the compiler will process the code
SDCCCFLAGS = -c -mmcs51 --std-sdcc99 --verbose $(MEMORYMODEL)

# These settings control where the linker will place the code
# and variables in memory.  The executable code will begin at 0000.
# External RAM usage for variables will begin at 0000.

ASLINKFLAGS = --code-loc 0x0000 --code-size 0x8000 --xram-loc 0x0000 --xram-size 0x8000 \
                  --out-fmt-ihx $(MEMORYMODEL)

# Default target
all: prog.hex

# Compile phase (the $< variable evaluates to the first prerequisite only)
syntax.rel: syntax.c syntax.h
             sdcc $(SDCCCFLAGS) $<

# Link phase (the $^ variable evaluates to all of the prerequisites for prog.hex)
prog.hex: syntax.rel
             sdcc $(ASLINKFLAGS) $^

# Phony target
.PHONY: clean
clean:
             del *.rel *.lst *.rst *.hex *.mem *.map *.sym *.lnk *.ihx
             del syntax.asm
```