CIS*2520 F24 - Briefing on Makefiles

A Makefile is:

- A type of file that handles and – in some ways – automates the building process of software.
- In the context of C programming, makefiles help streamline the compilation and linking of your C and header files. This is especially useful when working on bigger projects involving multiple source files.

Makefiles typically have 3 key components:

1) Targets
    - The targets generally include the files that the make automation tool is expected to generate (e.g., an object file such as `main.o`). The most common – and also the default – target is `all`, which builds the entire project.
2) Dependencies
    - The dependencies specify the files that the target depends on. For example, a target object file (e.g., `main.o`) may depend on a C and header file (e.g., `main.c header.h`).
3) Rules
    - The rules define how targets are built using its dependencies. They usually consist of command(s) that the make automation tool executes (e.g., `gcc -Wall -c main.c`).

Structure of a Makefile

```
target: dependencies
    Command rule
          ↓
main.o: main.c header.h
    gcc -Wall -c main.c
```

Some Makefiles may also choose to have:

- Variables `CC` for the compiler (e.g., `CC = gcc`) and/or `CFLAGS` for compilation flags (e.g., `CFLAGS = -Wall -c`). Variables are typically assigned at the top of the file.
- A target called `clean` that removes all generated object files from the working directory.

  ```
  clean:
      rm -f *.o main
  ```
- Developer comments for clarity.

Makefiles are useful for:

- Automating the compilation and linking process. By doing so, they save time and reduce errors.
- Only recompiling the files that have changed or are affected by changes, speeding up the entire build process. It does this by comparing the modification times of the targets to its dependencies.

Using Makefiles:

- Makefiles are run using the `make` command. Running this command processes all dependencies and associated command rules to build each target.

Any questions?