Recitation 1: Makefile and C

Department of Computer Science and Engineering
University of Minnesota

June 9, 2025

Recitation Guidilines

- Recitation materials will be released every Sunday at 11:59 PM.
- Yuo will have till corresponding Friday 11:59 pm to submit the **exercises.tar.gz** to **Canvas** as a single tar.gz file.
- Collaborations allowed.
- Submit individual work.

Overview

- GCC
- Makefile
- C programming
- GDB
- Valgrind
- Exercises

GCC

- GNU Compiler Collection
- Compiles C code to object file which may be executable

GCC Examples

```
# -o generates executable object file
gcc -o p1.o p1.c
# -c generates unlinked object file
gcc -o p1.o -c p1.c
```

make

- Controls the generation of executables and other non-source files of a program.
- Build and install your package without knowing the details
- Figures out automatically which files it needs to update, based on which source files have changed.

Makefile Guidelines

```
target: dependencies
    rules...
    rules...
```

Makefile

```
# Execute the object file p1.o
p1: p1.o
    ./p1.o
# Compile p1.o
p1.o: p1.c
    gcc -o p1.o p1.c
```

Makefile(cont.)

- Make runs the first target by default if no target is provided explicitly.
- Target `p1` depends on `p1.o`, and `p1.o` depends on `p1.c`
- Make errors if `p1.c` is not present in the current folder
- When typing `make` or `make p1` in terminal
    - Make runs `gcc -o p1.o p1.c` and created `p1.o`
    - Make then executes `p1.o`

Automatic variables

- Variables have values computed afresh for each rule that is executed. Examples(also check target `p2` in the sample):
    - `$@`: target name
    - `$<`: first dependency
    - `$?`: names of dependencies newer than target
    - `$^`: names of all the dependencies, with spaces between them
- Makefiles with a different name than `Makefile` could be executed using make as follows:
    - `make -f Makefilename.mk target`

Wildcard and automatic variables

- Wildcard: `*` and `%`
    - `*` searches file system for the matching name
    - `%`: replaced with a string, depends on the usage
- `CC`, `CFLAGS`, and `CPPFLAGS` are **implicit variables**.

Dynamic Memory Management in C

- Memory allocation and deallocation(check man page)
    - `malloc`
    - `free`
- Projects will have both static and dynamic memory allocation. Ensure to allocate and deallocate correctly.

gdb

- GNU debugger
- What is going on inside another program while it executes
- For C, use the −g flag while compiling the code to enable debugging
- Once the executable is run using gdb, you could step through the code workflow, add watch variables, put breakpoints.

gdb Example

```
# Compile with debugging symbols
gcc -g -o p3.o p3.c
# Invoke gdb
gdb ./p3.o
# Set breakpoints
(gdb) b p3.c:5
# Start the program
(gdb) r
```

gdb Example(cont.)

```
# Print variable
(gdb) p variable_name
# Next step/instruction
(gdb) n/s
# Continue
(gdb) c
# Check memory layout
(gdb) info proc mappings
```

Valgrind

- Tool to detect memory management, threading bugs, and for profiling code.
- Run Valgrind only after compiling code with `-g` flag to get the line numbers
- Use Valgrind to check for memory leaks: `valgrind --leak-check=full target_program`

Exercises

- Complete `Makefile1.mk` in the **exercise** folder
    - `root.c` depends on `dep1.c` and `dep2.c`
    - Complete the targets `root`, `dep1` and `dep2`
    - `dep1` and `dep2` targets should generate unlinked object files `dep1.o` and `dep2.o`
    - `root` target should generate the executable object file `root.o`
- Run gdb and Valgrind on `root.o`
    - Put a breakpoint at the for loop line in gdb
    - Observe the memory layout when the execution reaches the breakpiont(no need to report the result)
    - Run Valgrind and report the memory leak result in a text file, **memleak.txt**
    - Resolve the memory leak in root.c

Submission

- Submit the **tar.gz** of the completed **exercises** folder to **Canvas**. Add the **memleak.txt** to the tarball as well.
- Please do not submit any other files.
- The submission deadline is **June 13th, 11:59 PM**