

## Exercise 9: Creating a Makefile

Winter Term 2018/19

---

The lecture was on how to compile and build a program that has been split up into more than one C-file. The make-tool can support this task in a very efficient way.

**Unit 1:** Write a simple “Hello World” program. Move the part that outputs the text (i.e., the `printf()` function call), into a separate file `hello.c`. Edit an appropriate interface file `hello.h` in order to use the subroutine in the `main.c` program. Write a Makefile that maintains the program structure. Use a variable `CC` in the makefile to define the compiler, e.g., `CC=gcc`.

**Unit 2:** A “real” makefile is often a bit more complicated, especially if you are using Qt-files. Go to the directory of tutorial 7. The source files are `lcdrange.cpp`, `lcdrange.h`, and `main.cpp`. Take a look at the Makefile, and answer the following questions:

1. What is the program structure, i.e., which file depends on which one?
2. Which commands are used to create one file type from another one?

In order to validate the Makefile, do the following tasks:

1. Touch the file `lcdrange.cpp` (by using the command `touch`) and execute `make`. What does the command `touch`? Browse the command’s description `man touch`.
2. Do the same with the file `lcdrange.h`.

Which files need to be compiled in the two cases? What have you observed?

Have fun, Theo and Ralf