

Exercises for 01-C Build Tools

C-Build-Tools

Preprocessor

1.

Copy the following C program into a file `preproc-test.c`

```
#include <stdio.h>

int main() {

    int nr_students=10;

    printf("Nr of students: %d\n",
          nr_students);

}
```

Compile it and execute it.

2.

Make GCC halt after the preprocessing is done.

3.

Make GCC halt after the preprocessing is done and store the output in a file called `preproc-test.E`

4.

Make some changes to the file `preproc-test.c`:

- Add `#include <limits.h>` at the top of the file `preproc-test.c`.
- Add `printf("max in value: %d\n", INT_MAX);` at the end of the main function in the file `preproc-test.c`.

Invoke GCC and make it halt after preprocessing.

Look at the line we just added to `preproc-test.c`. Has it changed? Why?

5.

Compile and execute the program.

6.

Make some changes to the file `preproc-test.c`:

- Add `#define NR_STUDENTS 1234` after the `#include` statements at top of the file `preproc-test.c`.
- Assign `nr_students` like this `int nr_students=NR_STUDENTS;` in the file `preproc-test.c`.

Invoke GCC and make it halt after preprocessing.

Look at the line we just added to `preproc-test.c`. Has it changed? Why?

7.

Compile and execute the program.

Generate Assembler Code

8.

Make GCC halt after the compiler has produced assembler code.

What is the name of the output file.

Open the output file and try to understand what's going on. Give it only some few minutes. This will not be examined.

9.

Make GCC halt after the compiler has produced assembler code and make it store the output in a file called `apa.s`.

Makefile

Make sure that the program make (or GNU Make) is installed.

10.

Create a file called Makefile. In the Makefile you should add, using Makefile syntax (videos are available):

- a variable with a good name. The variable shall be assigned `preproc-test.c`

Invoke the program make in the following two ways:

- `make -f Makefile`
- `make`

Hmm, nothing worked. Why?

11.

To the Makefile above add:

- a rule called `all`. Make sure to add a colon (`:`) after `all`

Invoke the program make in the following two ways:

- `make -f Makefile`
- `make`
- `make -f Makefile all`
- `make all`

12.

To the Makefile above add:

- a rule called `11a`. Make sure to add a colon (`:`) after `all`

Invoke the program make in the following two ways:

- `make -f Makefile`
- `make`
- `make -f Makefile lla`
- `make lla`

13.

To the Makefile above add:

- the rule `all` shall depend on the variable in (10).

14.

To the Makefile above add a compilation directive to the `all` rule, like this:

```
gcc $(C_FILES)
```

15.

Invoke the program `make`.

You should now see a file `a.out` created.

16.

Change the compilation directive in the `add` rule to store the output file in a file called `preproc-test`: