École normale supérieure
L3 geosciences
Tutorial on compilation

Lionel GUEZ

June 3, 2022

Contents

1 make, source file and executable file 1

2 Compilation error 2

3 Run-time error 3

4 Dépendances entre objets, bibliothèques liées 3

5 Conclusion 3

1 Basic use of make; source file and executable file

1. Create an empty directory, for example `Tutorial_compilation`. With a text editor, write a minimal Fortran program which prints some message, for example "Hello. I am a Fortran program.". Save this program into a file `hello.f`, in the directory you have just created.

2. In the terminal, display the content of `hello.f` with cat. Can you just run `hello.f` by typing its path? Like this:

   ```
   ./hello.f
   ```

   Display the file permissions of `hello.f`.

3. Would it make sense to give execution permission to `hello.f` ? (You can try.)

4. With a text editor, create in the directory `Tutorial_compilation` a file named GNUmakefile containing the following line:

   ```
   FC = gfortran
   ```

5. In the terminal, try this command:

```
make
```

Explain what happens.

6. Try compiling the program hello with this command:

   ```
   make hello
   ```

   Explain what happens. Take a mental note of the type of error messages you get in this case, so you understand them if you encounter them again.

7. Add in GNUmakefile the following line:

   ```
   FFLAGS = -ffree-form
   ```

   (option for free source form). Try to compile again. Which file was created? Display its content with cat or with a text editor. Display its file permissions. Run it.

8. What happens if you type a second time the command make hello ?

9. In the terminal, go to another directory: any directory other than the one containing the executable file. Run the program from this other directory (without copying the executable file into this directory).

10. Remove the executable file hello. Move `GNUmakefile` to another directory (for example the parent directory of `Tutorial_compilation`). Go to the directory `Tutorial_compilation` in the terminal and type again the command:

    ```
    make hello
    ```

    What happens? Put back GNUmakefile in its rightful position: in the directory `Tutorial_compilation`.

2 Example of a bug signaled at compilation-time

1. Create a new file, say `test_compilation_error.f`, in the directory `Tutorial_compilation`, containing the following program:

   ```
   program test_compilation_error
     implicit none
     real a, b, c
     print *, "hello, I am a Fortran program."
     a = sqrt(-1.)
     b = 3
     c = a + b
     print *, " c = ", c
   end program test_compilation_error
   ```

2. This program has a bug. Compile it. Make sure you understand the error message from the compiler.

3. Correct the bug (using any positive value) and compile again.

3 Example of a bug appearing at run-time

1. Copy `test_compilation_error.f` to a new file, say `test_run_time_error.f`, in the same directory `Tutorial_compilation` and make some modifications to get the following program:

   ```
   program test_run_time_error
     implicit none
     real a, b, c
     print *, "hello, I am a Fortran program."
     a = -1.
     a = sqrt(a)
     b = 3
     c = a + b
     print *, " c = ", c
   end program test_run_time_error
   ```

2. This program has a bug. Compile it and run it. What happens?

3. Remove the executable file. Replace your GNUmakefile with the GNUmakefile with debugging options. Take a look at the content of this GNUmakefile (with a text editor or with the command cat). Compile your program and run it. Notice the difference with the previous run.

4 Dépendances entre objets, bibliothèques liées

Installez la bibliothèque Numer_Rec_95. Téléchargez le programme Coriolis. Écrivez un makefile pour le programme Coriolis sachant que :

- l'unité de programme principale est dans le fichier `coriolis.f` ;
- le programme utilise les bibliothèques `Numer_Rec_95` et Jumble ;
- `Numer_Rec_95` utilise Jumble.

Compilez. Pour simuler l'effet d'une modification d'un fichier source, tapez la commande :

```
touch coriolis.f
```

Recompilez.

5 Conclusion

Erreur courante 1, ne faites pas :

```
make hello.f
```

pour compiler. Vous confondez le fichier source et l'exécutable.

Erreur courante 2, ne faites pas :

```
./hello.f
```

pour exécuter. Vous confondez le fichier source et l'exécutable.

Make sure you understand and remember:

- that you must have a GNUmakefile to compile your program;
- that the GNUmakefile must be in the current directory;
- the error you get when there is no GNUmakefile;
- that the source file must have the suffix `.f`;
- what should be the content of a GNUmakefile.