

Compiling a Fortran Program

In this exercise you will create a file containing a very simple Fortran program. You will compile this program with the `f77` compiler and then run the program.

1. First, use the `cat` command of Unix (“`cat`” is short for “concatenate”) to create a file called `hello.f` containing a very simple Fortran program. Type the following:

```
% cat > hello.f
<tab>    print *,'hello, world!'
<tab>    stop
<tab>    end
<^D>
%
```

Here `<tab>` means you press the “tab” key, while `<^D>` means you hold down the “control” key while you press the “D” key. The `%` is the Unix prompt, so don’t type that. The `cat` command reads in the characters you type on the keyboard and the `> hello.f` *redirects* the output from `cat` into the file `hello.f`. The `<^D>` tells `cat` that you are done typing; it is the “end-of-file” character in Unix.

2. To see that this really created the file `hello.f` you can issue the Unix command `ls` to list the names of all of the files in the current directory:

```
% ls
bin      Mail      hello.f
```

Note that some of the files listed here are really directories. If you’d like to see which one, give the command `ls -F` instead, and this will put a `/` at the end of the name of each directory. (It will also put a `*` at the end of the name of each file which is executable, which we will come to in a moment.)

3. To verify what is in the file you can use the `more` command. It will display what’s in a file on your screen, one page at a time if it is longer than one screenful. Type `more` followed by the name of the file you wish to see, like so:

```
% more hello.f
print *,'hello, world!'
stop
end
```

4. Now we would like to *compile* the program in the *source* file (`hello.f`). The command to run the Fortran compiler is `'f77'`, so type:

```
% f77 hello.f
```

After the compiler finishes you can list the files in the current directory again,

```
% ls
a.out      bin      hello.f   Mail
```

There is a new file, called `a.out`, which contains the *executable* version of the program. (Type `'ls -F'` and you'll see a `"*"` after `"a.out"`.)

5. To run any executable file in Unix you simply type the name of that file at the prompt. So to run this simple program type:

```
% a.out
hello, world!
```

That's all there is to compiling and running a Fortran program. Now all you need to do is learn more Fortran commands.

Actually, that's not quite true, because we also need to have a better way to enter the program into the "source" file. What would you do if you made a mistake while typing in your program line-by-line, and your program happened to have 100 lines? Fortunately there are editing programs which make it easy both to enter your program from the start and to correct errors later on. One of the best editors is called "emacs," and that is the one we will use. Emacs has many special features which make it especially useful for working on Fortran programs.]