Topics Covered

- Simple Shell Script
- C++ and Fortran
  - installation, compiling source files
  - Makefile

Table of Contents

Installing a Fortran & C++ Compiler

Daniel Price Fortran Beginner's Tutorial (part 1)

OS: Ubuntu on Windows, i.e. Windows Subsystem for Linux

- to create a fortran file, type `touch file-name.f90` at the unix prompt.
- "f90" is the extension used for all modern fortran code.

Go ahead and open `file-name.f90` on your text editor.

hello world code:

```
program hello

print*, "hello world"

end program hello
```

Fortran is a compiled language. This means you'll need a compiler to run this script.

How to Install GCC Compiler on Ubuntu 18.04

Install manual pages about using GNU/Linux for development:

```
sudo apt-get install manpages-dev
```

To validdte that the GCC compiler is successfully installed, check which version you have:

```
gcc --version
```

- GCC can compile programs written in C, C++, Java, and Fortran.
- "GCC" is a common shorthand term for the GNU Compiler Collection.

GNU Fortran: Using the compiler

First use `apt install gfortran` to install gfortran.

Compile the source file `hello.f90`:

```
gfortran -c hello.f90
```

This makes an object file, `hello.o`, which needs to be "linked" into an executable.

Shell Scripting

Writing Our First (Shell) Script and Getting It to Work

To successfully write a shell script, we must do three things:

1. Write a script
2. Give the shell permission to execute it
3. Put it somewhere the shell can find it

1. Writing a shell script

- A shell script is a file that contains ASCII tet
- To create a shell script, we use a text editor.
- A text editor is a program that reads and writes ASCII text files.

Fire up your text editor and type (copy-paste) in this script:

```
#!/bin/bash

# My first script

echo "Hello World!"
```

The first line of the script has a special construct, `#!`, called a "shebang".

- This indicates which program will be used to interpret the script. In this case, `/bin/bash`.
- Other scripting languages such as Perl and Python also use this mechanism.

The second line is a comment.

- Everything that appears after a "#" symbol is ignored by BASH.
- As scripts become

The last line uses the `echo` command.

- `echo` prints its arguments on the display.

2. Setting permissions

3.

Write and compile a shell script

Makefile

Pipes, BASH commands

[Linux Leech video](#)

Pipes allow you to take the output of one command and use it as input for another.

Let's say you had two commands, c1 and c2. The syntax used to pipe would be `c1 | c2`. Whatever c1 would output to the display will now be used as input to c2.

Some BASH commands:

- `clear`: clear the screen
- `mv`: Move files
- `pwd`: print the working (current) directory
- `ls -R`: List everything in a directory and in all subdirectories recursively
- `cp [options] source dest`: copy a file from source to destination
- `cp [options] source0 source1... dest_dir`: copy multiple files from source to destination
- `rm`: "remove": deletes files
  - `rm [options] file0 file1...`:

Pagers:

- `head`

- `tail`

- `touch`: create blank file

- `mkdir`: make directory

- `rmdir`: remove directory

Filters:

- `grep`: Search input using regular expressions
- `sort`: Sorts input by lines (lexically, or numerically)
- `uniq`: Unique, removes identical, adjacent lines
- `wc`: Word count (line count, character count)
- `cut`: Select fields of a line

[continue linux tutorial](#)

Move files from WSL to Windows

1. Open the file explorer.
2. Enter `\\wsl$\` in the address bar and you should see your WSL distribution.
3. The files form WSL are listed in the `root` directory.