

COMP2100 Workshop Week 1

Getting Started in Unix and C

Background

C programming is a major thread running throughout this unit. This practical exercise is designed to get you started in C and in using Linux. All your assignments (a.k.a *labs* in this unit) are also to be done using C in the Linux environment.

Logging In to Unix and Editing Files

Before you start C programming, you need a working environment. All the programs that you use in COMP2100 will be compiled on our Linux servers `ash.science.mq.edu.au` and `iceberg.science.mq.edu.au`. It is necessary for you to be able to edit, compile and run programs in this environment. Along the way you will learn how to use a Unix system if you have not experienced this before.

Your lab supervisor will show you how to connect from the Windows machines in the lab to `ash` or `iceberg`. Detailed information about working on COMP2100 from Windows machines, whether in the Computing lab or at home, can be found in the document *COMP2100 from Windows*, which is in the Lab Notes section on iLearn. For Mac users, you may try <https://www.ssh.com/ssh/putty/mac/>.

The best starting point for working in the Computing lab is to use Putty to login to `ash` or `iceberg`.

The UNIX/Linux (or Unix-like OS) `man` command

Linux is the Operating System (OS) used in both `ash` and `iceberg`. Try the following command to find out the actual OS name. Note: “\$ ” represents the system prompt – you don’t type “\$ “. The command that you type is underlined in this example, but in future we won’t bother with underlining.

```
$ uname
```

UNIX manuals are stored on each UNIX computer and can be accessed with the `man` command. Try the following command.

```
$ man ls
```

This command will display the first screen full of the manual page for the `ls` command. The ‘:’ prompt shows you that the `man` command is waiting for you to respond. You can: press up and down arrows to scroll through the manual page content, press space bar to get the next screen full, press Enter to get the next line, or press ‘q’ to quit.

Task 1: Now find out the Linux distribution and kernel version used in `ash` and `iceberg`.

Here are some other commands that you will find useful in the coming weeks. Have a quick look at the `man` pages now.

```
gcc, od, strings
```

The UNIX commands

Task 2: Read “Introduction to the UNIX Operating System” and try “Tutorial One” and “Tutorial Two” in the following page.

<http://www.ee.surrey.ac.uk/Teaching/Unix/>.

Everybody's First C Program

Try your first C program.

Task 3: type in the following C program and save it as `hello.c` using `vi`, `vim` and `pico/nano`, in your terminal, respectively. Use `man` command, e.g., `man vi` or search the web

(<https://www.cs.colostate.edu/helpdocs/vi.html>)

```
1.  # include <stdio.h>
2.
3.  int main(int argc, char **argv) {
4.      printf ("Hello world!\n");
5.      return 0;
6.  }
```

* You may edit the file using a text editor of your choice with GUI, e.g., Notepad++ on Windows; then make sure that your file has Unix line endings.

Compiling and Running the Program

The C compiler is called `gcc`. It has many command-line options. The simplest way to compile a program like this is as follows.

```
$ gcc hello.c
```

To run your program, type its name at the command prompt as follows.

```
$ ./a.out
Hello world!
$
```

Note that you should always use the “./” when running your own programs because otherwise the system will look for a system command with the same name and the results may surprise you.

Task 4: recompile `hello.c` with the executable name as `hello`.

Modifying the Program

Now it is time to personalise your hello program.

Task 5: Change the word “world” to your name and verify that the program now says hello to you personally.