COMP1511 - Programming Fundamentals

Week 1 - Lecture 1

Total Assessment

- 10% Labs due Sunday
- 5% Weekly Tests 1 hour 3Q Week 3-5 7-10
- 15% Assignment 1 due week 6
- 25% Assignment 2 week 10
- 45% Exam

To pass the course you must:

- Score at least 50/100 overall
- Solve problems using arrays in the final exam
- Solve problems using linked lists in the final exam

What is a Computer?

A tool . . . a machine . . .

重新设定

The ultimate tool in its ability to be reconfigured for different purposes.

The key elements:

处理器 执行 命令

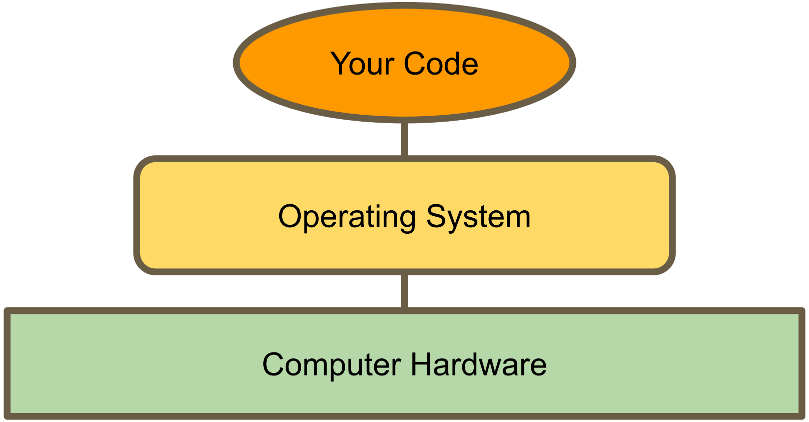
- A processor to execute commands
- Memory to store information

Working in Linux

The first thing is to get setup with a simple programming environment

Here at CSE we use the Linux Operating System

An Operating System sits between our code and the computer, providing essential services



Using a Terminal 终端

主界面

The main interface to Linux is a terminal

互动

This means all our interaction is in text

Some commands:

目录

- **`ls`**
  - Lists all the files in the current directory
- **`mkdir` *`directoryName`***
  - Makes a new directory called directoryName
- **`cd`**
  - Changes the current directory
- **`pwd`**
  - Tells you where you are in the directory structure at the moment

What the basics look like

gedit

- A basic text editor 编辑器
- Helps out a little by highlighting C in different colours

dcc/gcc

- A compiler 编译器 - A translator that takes our formal human readable C and turns it into the actual machine readable program
- The result of the compiler is a program we can "run"

You can use VLAB to access CSE's editor and compiler

Let's see some C

```c
// Demo Program showing output
// Marc Chee, June 2019

#include <stdio.h>

int main (void) {
    printf("Hello World.\n");
    return 0;
}
```

comment

/* ...... */

可在code中

加comment

is not code

Comments

之后都是 comments

```
// Demo Program showing output
// Marc Chee, June 2019
```

Words for humans

- Half our code is for the machine, the other half is for humans! (roughly)
- We put “**comments**” in to describe to our future selves or our colleagues what we intended for this code
- **//** in front of a line makes it a comment
- If we use **/*** and ***/** everything between them will be comments
- The compiler will ignore comments, so they don’t have to be proper code

#include

```
#include <stdio.h>
```

#include is a special tag for our compiler

夺取

It asks the compiler to grab another file of code and add it to ours

输入，出库

In this case, it's the Standard Input Output Library, allowing us to make text appear on the screen (as well as other things)

The "main" Function

```
int main (void) {
    printf("Hello World.\n");
    return 0;
}
```

A function is a block of code that is a set of instructions

指令

Our computer will run this code line by line, executing our instructions

The first line tells us (things we'll cover in detail later):

- int is the output - this stands for integer, which is a whole number
- main is the name of the function
- (void) means that this function doesn’t take any input

The Body of the Function

```
int main (void) {
    printf("Hello World.\n");
    return 0;
}
```

Between the { and } are a set of program instructions

printf() is actually another function from **stdio.h** which we included. It makes text appear on the screen

return is a C keyword that says we are now delivering the output of the function. A main that returns **0** is signifying a correct outcome of the program

Editing and Compilation

We can open a terminal now and try the code we've just looked at

In the linux terminal we will open the file to edit by typing:

```
    gedit helloWorld.c &
```

Once we're happy with the code we've written, we'll compile it by typing:

```
    dcc helloWorld.c -o helloWorld
```

The **-o** part tells our compiler to write out a file called "helloWorld" that we can then run by typing:

```
    ./helloWorld
```

The ./ lets us run the program "helloWorld" that is in our current directory