COMP 2718: Introduction

By: Dr. Andrew Vardy

Adapted from the notes of Dr. Rod Byrne

Outline

- What this course is about
- The command line interface (CLI)
- CLI vs. GUI
- What uses are there for the CLI?

What this course is about

Its all in the title:
"Software Development Tools, Work Flows, and Concepts"

- Tools: Individual programs with a specific job
- Workflow: A pattern of activities, usually to process information
- Concept: An abstract idea

Most of the tools we will use are executed from the command line.
So what is the command line?

The Command Line

We interact with desktop computers through a command line interface (CLI) and/or a graphical user interface (GUI).

A command line is a line of text where the first word specifies a command (i.e. a program) to execute, the rest of the line specifies the arguments (a.k.a. options) that are passed to the program.

For example, the following command lists files in the current *directory* with the `-S` option meaning that files should be listed in sorted order, by size.

```
ls -S
```

[Demo]

CLI vs. GUI

Use of the CLI is based on reading and outputing text, yet the underlying commands may be any sort of data (e.g. images).

The following are advantages of CLI over GUI:

- Commands can be connected together in sophisticated ways
- Interacting with remote computers requires much less bandwidth
- Requires less computing power for the interface itself (e.g. fancy graphics card not required)
- More universal

This final point may seem strange. If all you have seen are GUIs, how can the CLI be more universal?

CLI vs. GUI (Universality)

Consider the following types of computing systems:

- Servers
- Desktop computers
- Smartphones
- Embedded devices (with an operating system)

With some exceptions there are CLIs for all of the above, but some computers may not be powerful enough for GUIs (embedded devices) and some may be physically inaccessible (servers).

CLI vs. GUI (Disadvantages)

CLIs have a number of disadvantages when compared with GUIs:

- Effective use of the CLI requires more memorization (e.g. there are no menus)
- Discovering the capabilities of the CLI is harder
- No graphical feedback!

It is quite typical to utilize the best features of both CLI and GUI simultaneously by running a *terminal program* that provides a CLI within your GUI.

What uses are there for the CLI?

- Software development
- System adminstration
- Processing data from various applications
- Numerous others. . .

We will emphasize software development, but there are command line users in every domain—particularly in technical fields such as Computer Science, Physics, Mathematics, Engineering, etc. . .

Workflows

A workflow is a set of tasks performed by a person to accomplish an effect. The tasks are usually described in a general way, but are performed with all necessary details provided.

Lets see some example workflows. Note that the commands used below will be introduced later on. This is just to give a flavour for command line workflows.

WF: Copy files to remote machine

```
% scp ZZZ.txt av@garfield.cs.mun.ca:.
```

This copies local file ZZZ.txt to my home directory on the server, garfield. To adapt this workflow to your use, change the appropriate parts (e.g. ZZZ.txt, av).

WF: Compile simple C program

[Place the following contents in t.c]

```
#include <stdio.h>
int main() {
    printf("hi!\n");
    return 0;
}
```

```
% gcc t.c
% ./a.out
Hi
```

WF: Compile another C program, specifying the name of the executable produced

[Place the following contents in square.c]

```c
/* Takes in number as first arg and outputs its square. */
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char* argv[]) {
    int i = atoi(argv[1]);
    printf("%d\n", i*i);
    return 0;
}
```

Lets compile and execute it:

```
% gcc square.c -o square
% square 3
9
```