

Lecture 0 Notes

Introduction

CSE 201 is one of the first CSE courses that students at CSUSB enroll in. Many new students are unaware of what Computer Science and Engineering demands from it's potential scholars. It is first important to know that we are not studying the computer; the computer is simply a tool for us to apply our theories. What we are studying is: not the computer, but computation itself, or (in other words) algorithms. An algorithm is a set of instructions for performing a particular task.

Creating algorithms involves a significant application of logic and math. Inside a computer, an algorithm can physically exist in the form of a computer program. the process of creating a computer program is called computer programming.

CSE 201 serves as an introduction to computer programming. We will use a programming language known as C++.

Command-Line Interface

Before we start learning about C++, it is important for any CSE student to become familiar with using the Command-Line Interface (CLI).

A long time ago, computers lacked the capability to display images; these computers also lacked mouse input. Computers back then were only capable of reading and displaying information in the form of text to it's users. Because of this, interfacing with a computer required the user to know a collection of commands that they would have to type line by line. After each command was input, the computer would respond with some text that would fill the screen.

This type of interface between the user and the computer came to be known as the command-line interface.

Despite it's primitiveness and steep learning curve, CLI is still very popular among programmers and computer enthusiasts today. Unlike modern Graphical User Interfaces (GUI), CLI does not need to clutter the screen with additional buttons or menus for each new functionality that it supports. CLI offers more power and flexibility to the user by offering automation of certain tasks and the ability to connect the processing of data through the execution of multiple programs.

The official operating system for the CSE department is Linux, it is also the most popular operating system for programmers and computer enthusiasts. You can access the CLI on this operating system by running an application known as Terminal. While the terminal is used for displaying the command-line interface, the real program responsible for interpreting the commands that you type is known as the shell. There are several shells available, the most popular one used on Linux is known as Bash.

This will be the only lecture where CLI is covered. As a student, you will be expected to become skilled at the command-line through practice and self-study. If you would like, you may take an optional course: CSE 360. Script Programming. That course covers command-line and text processing.

Very brief introduction to CLI

For a very brief introduction to Bash and CLI in general, Let us start looking at some simple commands supported by Bash. After you run Terminal, you will be greeted with a black or white window and some text that might look like:

```
mark@mainpc ~/
```

This text is known as the shell prompt. It is what gets displayed before and after each command that you type. You can customize what gets displayed in the prompt through further investigation. By default, the prompt displays your username followed by an '@' symbol, followed by the name of the computer, followed by the folder that you are navigated in (known as the current directory).

One command that you should know is "ls". ls stands for list (probably) and it is used for listing files. Type "ls" without double-quotes and press enter. The window will fill with a list of files and folders inside of your current directory. So you might see a list that looks like this:

```
Desktop      Documents    Downloads    Music
Pictures     Videos
```

These are all folders. If I wanted, I can choose to create a new folder named "Projects" by typing in the following command:

```
mkdir Projects
```

mkdir stands for Make Directory, it is used for creating a folder (In Linux we call folders directories).

If I wanted to move the projects directory into my Documents directory, I would use this command:

```
mv Projects Documents
```

If I had a file named "hello.cpp" in my home directory, I can use the following command to display it's contents in the terminal:

```
cat hello.cpp
```

I can create a backup file of hello.cpp with the name hello.cpp.bak:

```
cp hello.cpp hello.cpp.bak
```

The pattern here is that everything you type in the CLI follows this format:

```
[command] [argument 1] [argument 2] [...] [argument n]
```

Depending on the command, the arguments might be optional, otherwise the arguments are additional information needed by the command in order to function. For example, the mkdir command needs the argument specifying the name of the folder to create.

Some commands do not need arguments, such as the "date" command which displays the current date and time, or the "whoami" command which displays your username.

The CLI is not only useful for file manipulation, but also: text processing, math, computer security configuration, online file transfers, email, music playback. Even web browsing is possible in the CLI (but without images or videos). If you learn enough about it, you can do anything on the command-line except web browsing, and anything else related to graphics.

There are too many useful Bash commands to cover. Even though your first assignment will be on using the command-line, you will be expected to learn them on your own, there are plenty of online resources to help you. Most of the instruction in this course will be on computer programming using C++.

Terminology

Before you start researching this on your own, remember these terminologies:

- “Linux” is an operating system.
- “Bash” is a shell.
- “Gnome Terminal” is a terminal emulator.

An operating system is the software that directly controls your computer’s hardware on behalf of all other software installed on your computer. Examples of operating systems include: Linux, Mac OS X, and Windows.

A shell is a program for performing actions available from your operating system. A shell be a command-line interface (CLI) or a graphical-user interface (GUI). Examples of graphical shells for Linux include: Gnome, LXDE, KDE. The official Mac OS X graphical shell is called “Aqua”. The official Windows graphical shell is called “Windows Shell”. Bash is a command-line interface shell available for Linux and Mac OS X. Command Prompt is the CLI shell offered by Windows.

A terminal emulator is a program that provides an alternative interface from the rest of your operating system’s interface. Typically this alternative interface is a text-only interface. Terminal emulators are often used for interfacing with a CLI shell.

If you want to learn about CLI commands for Linux, you should use the word “Bash” in your search engine queries.

Here is a list of Bash commands that you may find yourself frequently using: pwd, cd, ls, cp, mv, rm, mkdir, rmdir, zip, unzip, ssh, scp, sftp, cat, pr, expand, nl, lpstat, lpr, clear, grep, find, du.