SIM UOW CSCI251

ASSIGNMENT 1

Last updated: 8 January 2024

Timothy Mah

10258663

Lecture Group L01

Contents

[Introduction 2](#_Toc155625895)

[Limitations 2](#_Toc155625896)

# Introduction

Hello this is my documentation for SIM UOW CSCI251 ASSIGNMENT 1.

# Run

To install

sudo apt-get install libncurses5-dev

To compile

g++ testreg.cpp -lncurses -o testreg.app

# Limitations

1. Regex string may be system agnostic. However, in today’s context where many programs run over the web, server side processing would mitigate this issue.
2. The ncurser library is only available on Linux. Hence my app can only be compiled on Linux systems.

# Learnings

Some cool stuff I learnt

## How to update elements in vector with range type for loop

Personally I prefer to code in python style. Hence, the python style for loop is more intuitive to me.

        vector<string> strings;

        for (string s: strings)

        {

            s = "some string";

        }

In the above code snippet, the strings within the vector would be updated with the value, “some string”. However, the value “some strings” will only stay persistent within this for loop. Outside of the for loop, the values in strings is still empty.

To make the value persistent globally, reference the memory location of elements within the vector.

        vector<string> strings;

        for (string &s: strings)

        {

            s = "some string";

        }

Just one simple & is needed.

Of course I could also do the traditional for loop style with counter++ and all but I like this python style for loop. So this is like a mix of C++ style with python. Lol

## Some functions want C style string, not regular string

I used this function called mvwprintw() from the ncurser library. It’s fourth parameter takes in a string. I had an int type variable that I wanted to use. I looked up ways to convert int to string, however I still received an error message. Upon closer inspection, the function wanted C style string const char\*. So I searched up how to convert string to C style string. In the end I mashed two functions together to become to\_string(gridy).c\_str()

But in the end I realised I cannot use external libraries so I scrapped it lol.