

CS470

Lab 4

1.)

Terminal Screenshots performing a hardlink and a softlinking:

```
cs470@CS470-virtualbox:~$ touch filetobesoftlinked.txt
cs470@CS470-virtualbox:~$ ln -s filetobesoftlinked.txt softlink.txt
cs470@CS470-virtualbox:~$ ls -l
total 72
drwxrwxr-x 3 cs470 cs470 4096 Feb  1 17:10 CLionProjects
drwxr-xr-x 2 cs470 cs470 4096 Jan 10 08:24 Desktop
drwxr-xr-x 2 cs470 cs470 4096 Jan 10 08:24 Documents
drwxr-xr-x 2 cs470 cs470 4096 Jan 10 08:24 Downloads
-rw-rw-r-- 2 cs470 cs470   48 Feb 22 05:55 filetobehardlinked.txt
-rw-rw-r-- 1 cs470 cs470    0 Feb 22 07:40 filetobesoftlinked.txt
-rwxrwxr-x 1 cs470 cs470 17240 Jan 30 22:28 fork
-rw-rw-r-- 1 cs470 cs470   260 Jan 30 22:28 fork.c
-rw-rw-r-- 2 cs470 cs470   48 Feb 22 05:55 hardlink.txt
drwxr-xr-x 2 cs470 cs470 4096 Jan 10 08:24 Music
drwxr-xr-x 2 cs470 cs470 4096 Jan 10 08:24 Pictures
drwxr-xr-x 2 cs470 cs470 4096 Jan 10 08:24 Public
drwx----- 6 cs470 cs470 4096 Jan 12 08:26 snap
lrwxrwxrwx 1 cs470 cs470   22 Feb 22 07:40 softlink.txt -> filetobesoftlinked.t
xt
drwxr-xr-x 2 cs470 cs470 4096 Jan 10 08:24 Templates
drwxr-xr-x 2 cs470 cs470 4096 Jan 10 08:24 Videos

cs470@CS470-virtualbox:~$ touch filetobehardlinked.txt
cs470@CS470-virtualbox:~$ ln filetobehardlinked.txt hardlink.txt
cs470@CS470-virtualbox:~$ ls
CLionProjects  Downloads      fork.c         Pictures      Templates
Desktop        filetobehardlinked.txt  hardlink.txt  Public        Videos
Documents      fork          Music          snap

cs470@CS470-virtualbox:~$ ls -;
ls: cannot access '-': No such file or directory
cs470@CS470-virtualbox:~$ ls -l
total 64
drwxrwxr-x 3 cs470 cs470 4096 Feb  1 17:10 CLionProjects
drwxr-xr-x 2 cs470 cs470 4096 Jan 10 08:24 Desktop
drwxr-xr-x 2 cs470 cs470 4096 Jan 10 08:24 Documents
drwxr-xr-x 2 cs470 cs470 4096 Jan 10 08:24 Downloads
-rw-rw-r-- 2 cs470 cs470    0 Feb 22 05:54 filetobehardlinked.txt
-rwxrwxr-x 1 cs470 cs470 17240 Jan 30 22:28 fork
-rw-rw-r-- 1 cs470 cs470   260 Jan 30 22:28 fork.c
-rw-rw-r-- 2 cs470 cs470    0 Feb 22 05:54 hardlink.txt
```

2.) Multithreaded program that computes max and min of an array using separate threads.

The code written takes guidance from the example given in Tutorial 8, with the implementation of two threads, arguments are collected specifically through using `strtol()`, these args are converted into

numbers to be stored in an array. Here is a screenshot of the program being used:

```
/home/cs470/CLionProjects/lab3/cmake-build-debug/lab3 1 2 3 4 5942392 48842534 4 -5
The minimum value is -5
The maximum value is 48842534

Process finished with exit code 0
```

Here is the sourcecode:

```
#include <pthread.h>

#include <stdio.h>

#include <stdlib.h>

#include <iostream>

#include <string>

#define NUM_THREADS 2

int* numbers;

int num_count;

int max, min;

void *calc_max(void *arg) {

    max = numbers[0];

    for (int i = 1; i < num_count; i++) {

        if (numbers[i] > max) {

            max = numbers[i];

        }

    }

    pthread_exit(NULL);

}

void *calc_min(void *arg) {

    min = numbers[0];

    for (int i = 1; i < num_count; i++) {

        if (numbers[i] < min) {

            min = numbers[i];

        }

    }

}
```

```

    }

    pthread_exit(NULL);
}

int main(int argc, char *argv[]) {
    pthread_t threads[NUM_THREADS];
    numbers = new int[argc-1];
    for (int i = 1; i < argc; i++){
        numbers[i-1] = (int)strtol(argv[i], NULL, 10);
    }
    num_count = argc-1;
    int rc;
    rc = pthread_create(&threads[0], NULL, calc_max, NULL);
    if (rc) {
        printf("Error: Unable to create thread.\n");
        exit(-1);
    }
    rc = pthread_create(&threads[1], NULL, calc_min, NULL);
    if (rc) {
        printf("Error: Unable to create thread.\n");
        exit(-1);
    }
    for (int i = 0; i < NUM_THREADS; i++) {
        rc = pthread_join(threads[i], NULL);
        if (rc) {
            printf("Error: Unable to join thread.\n");
            exit(-1);
        }
    }
    printf("The minimum value is %d\n", min);
}

```

```

printf("The maximum value is %d\n", max);

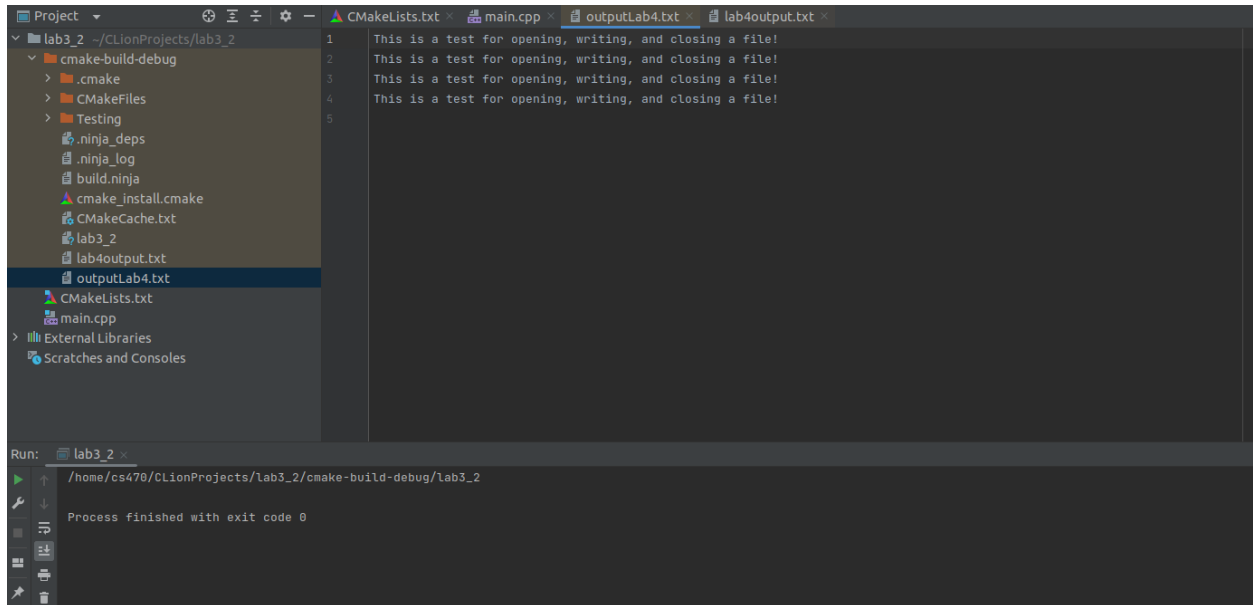
pthread_exit(NULL);
}

```

3.) create a program that appends a sentence to a specific file.

The implementation of this was to just use `open()` and check if the file is open, if it is the case that the file is open it simply appends to file the sentence desired using the `fstream` and the `<<` operator. Finally the `fstream` is closed.

here is a screenshot of the code run and the resulting file:



Sourcecode:

```

//
// Created by Austin B. on 2/22/23.
//
//

```

```

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include "iostream"

#include <string>

```

```
#include <fcntl.h>
```

```
#include <fstream>
```

```
using std::cout; using std::fstream;
```

```
using std::endl; using std::string;
```

```
int main() {
```

```
    string filename("outputLab4.txt");
```

```
    fstream file;
```

```
    file.open(filename, std::ios_base::app | std::ios_base::in);
```

```
    if (file.is_open())
```

```
        file << "This is a test for opening, writing, and closing a file!" << endl;
```

```
    file.close();
```

```
    return EXIT_SUCCESS;
```

```
}
```

4.) implement matrix math operations

```
// TODO [#C] write the code for this.
```