

CSC 412 - Programming Assignment 04, Fall 2019

Jordan Coats

Abstract—The objective of this assignment is to experiment with `fork()`, image processing application, use bash scripting through a directory, and Doxygen for documentation.

I. INTRODUCTION

This project was built around TGA image manipulation. The goals were to make a program that takes n amount of images and tiles them together in the "most square" orientation as possible. The second part was to create a dispatcher app that uses `fork()` to do various image manipulation.

II. TILE.C

The first obstacle to tackle for this assignment was creating the `tile.c` program. The first part of this assignment is to take the first of n images and use it to create a tile to hold all of the entries. As we were guaranteed same dimensions I took the first image's dimensions and by taking the square root of the n , then the ceiling of that number, I was able to multiply the first image's width and height by that number to create, in my opinion, the most square tile image as possible. Next it was a matter of looping through the rest of the images, while moving to each quadrant, and writing the images into the tile.

III. DISPATCHER.CPP

The following program I unfortunately was not able to finish. My ideas for it are as followed:

```
std::vector<std::string> rotate;
std::vector<std::string> crop;
std::vector<std::string> split;
```

I would create three vectors that would store the flattened arguments with a counter to know how many children I would need to run.

```
std::vector<std::string>
args(argv, argv + argc);
for (int i = 3; i < args.size(); ++i)
{
    std::string str;
    if (args[i] == "rotate")
    {
        for (int r = 0; r < 2; r++)
        {
            str.append(args[i]);
            str.append(" ");
            i++;
            child_counter++;
        }
        rotate.push_back(str);
    }
}
```

```
str.clear();
}
...
```

The next step was to store all of the arguments into one vector, `args`, to be able to string compare for when it hit a keyword of "rotate", "crop", or "split" then it would know to store those in the respective vector and increment the line to where the next potential input should be. From there my plan was to create children depending on "children_counter" which would send each command, one at a time, to each children for rotate, crop, split along with the output and correct input.

IV. BASH SCRIPT

```
#!/bin/bash

gcc tile.c /ImageIO/imageIO_TGA.c
-o -lm "tile.o"

LIST=" "
for entry in $(ls *.tga | sort -V);
do
    echo $entry
    LIST+="${entry} "
done

./tile.o $LIST $2
echo LIST
```

My bash script did not work as intended as I could not get the "-lm" function for `math.h` to work correctly. Otherwise the goal was to flatten all of the images into a sorted list then use that list as the command for `tile` along with the output directory which should have been `$2`.

V. CONCLUSIONS

Another challenging project that I spent too much time trying to figure out by myself instead of going to office hours or asking for help from Dr. Herve. Hopefully the next assignment will prove to be better.