CSC 412 - Programming Assignment 02, Fall 2019

Jordan Coats

Abstract— **The first objective of this assignment is to get some practice with C arrays, pointers, and strings. The second is to learn how to iterate through the catalog of a directory in bash.**

I. INTRODUCTION

The first part of this assignment is taking in an input string and eventually sorting it completely by ASCII values and type. The second part of this assignment deals with constructing a custom struct data type to deal with the sorting of the previous part. Finally the assignment deals with directory cataloging in bash.

II. STRUCTS

The first two parts of this assignment were relatively simple compared to the third part where we were tasked with creating a ”struct of structs” to hold our data type. The encompassing struct would need to be a container struct that held the different the structs of certain types. They would each need to include name, unique index for sorting, and an ASCII range at minimum. For this assignment I started off creating a struct named ”data” to use for my different categories:

```
struct data
{
    char name[8];
    unsigned int index;
    unsigned int rangeCount;
    unsigned int startRange[10];
    unsigned int endRange[10];
};
```

The struct takes a name of 8 length, with an integer index (which should match the index type chosen), and the potential for 10 different range pairs stored in two arrays. They are all unsigned ints because I didn’t see the need of having negatives as the ASCII values are only positive.

Next I created a struct that would contain this struct:

```
struct cat
{
    unsigned int count;
    struct data type[10];
};
```

This ”cat” or category struct has the number of categories as an integer and the potential for 10 different categories to be added to it. Each ”type” is identified by the unique index and char name that would be given to it.

For the forth version of the task I created the ”maths” struct to hold the 7 different characters each as a start and end range value.

III. BASH

For the bash part of the assignment the first step was to research how to list files using bash commands. While coming across several different ways I settled on find being the most appropriate for my use. Testing of the script resulted in:

```
path="$1"
filenames=`ls -p $path | grep -v /`
for entry in $filenames
do
   echo $entry
done
```

This code works well to list all of the files in a directory without listing the directories by forcing them with ”-p” to follow ”/*” and using grep to not include them. For the categories I used:

```
source=`ls -p $path | egrep '\.c$|\.cpp$'`
header=`ls -p $path | egrep '\.h$|\.hpp$'`
script=`ls -p $path | egrep '\.sh$'`
text=`ls -p $path | egrep '\.txt$'`
directory=`ls -p $path  | egrep  '/$'`
other=`ls -p $path --ignore="*"
    {.c,.cpp,.h,.hpp,.sh,.txt} | egrep -v /$`
```

As a way to sort everything into different category lists, then using several if/else statements like:

```
if [ -z "$source" ]; then
    :
else
    echo "Source:"
    for entry in $source; do
        echo -n "$entry "
    done
    echo ""
fi
```

So if the list is empty it won’t even echo the header, this keeps the script uncluttered.

IV. CONCLUSIONS

This change of data structure did involve a rewrite to the original sorting function as it no longer used hard coded values and instead used the values that populated the struct. Once that rewrite was complete the use of structs opens up the program to be modified to have more categories easily input without the need to change the sorting function every time. I’ve tested the BASH scripts on multiple areas and both seem to accomplish both goals.