CPE422 Homework 05
Fall Semester 2024
By John Daddona

Link to repository: https://github.com/JohnDaddona/Daddona-HW6.git

Problem 01:

1. Explain how strcmp works. [5 points]

Strcmp compares two strings against one another character by character. In our case, it compares argv[1] to the commands outlined within the program. If any of the strings within the program match the string provided by the user, it will return 0, allowing the program to execute the action.

2. Explain the statement: FILE* fp; [5 points]

This declares the pointer fp of type FILE to handle files. It points to the directory, more specifically, the files within the directory /sys/class/leds/ and so on, for controlling the LEDs and reading their status.

3. Explain how sprintf works. [5 points]

Sprintf formats and then proceeds to store a string within a buffer. It combines LED3_PATH and the associated file name, /brightness or /trigger into fullFileName to access the correct control files to control the LED.

4. Explain how fprintf works. [5 points]

Fprintf writes formatted data to a file. In our code, it writes the values of the LED status (0 or 1) to the appropriate LED control file, fp, which will then modify the LED’s state.

5. Explain how fgets works. [5 points]

Fgets reads a line from a given file. In our code, it reads the status of the LED from /trigger. Based on what it reads, it prints it to the console to tell the user of the LED's current mode.

```
#include <stdio.h>
```

```
#include <stdlib.h>
#include <string.h>

#define LED3_PATH "/sys/class/leds/beaglebone:green:usr3"

void writeLED(char filename[], char value[]);
void removeTrigger();

int main(int argc, char* argv[]) {
   if (argc != 2) {
      printf("Usage is makeLEDC and one of:\n");
      printf("   on, off, flash, or status\n");
      printf(" e.g. makeLED flash\n");
      return 2;
   }

   printf("Starting the makeLED program\n");
   printf("The current LED Path is: " LED3_PATH "\n");

   // select whether the command is on, off, flash, or status
   if (strcmp(argv[1], "on") == 0) {
      printf("Turning the LED on\n");
      removeTrigger();
      writeLED("/brightness", "1");
   } else if (strcmp(argv[1], "off") == 0) {
      printf("Turning the LED off\n");
      removeTrigger();
      writeLED("/brightness", "0");
   } else if (strcmp(argv[1], "flash") == 0) {
      printf("Flashing the LED\n");
      writeLED("/trigger", "timer");
      writeLED("/delay_on", "50");
      writeLED("/delay_off", "50");
   } else if (strcmp(argv[1], "status") == 0) {
      FILE* fp;
      char fullFileName[100];
      char line[80];
      sprintf(fullFileName, LED3_PATH "/trigger");
      fp = fopen(fullFileName, "rt"); // reading text this time
      if (fp == NULL) {
         perror("Error opening file");
         return 1;
      }
      while (fgets(line, sizeof(line), fp) != NULL) {
         printf("%s", line);
      }
      fclose(fp);
   } else {
      printf("Invalid command!\n");
      while(2){
         writeLED("/brightness", "1");
         sleep(1);
         writeLED("/brightness", "0");
         sleep(1);
      }
   }

   }

   printf("Finished the makeLED Program\n");
   return 0;
}

void writeLED(char filename[], char value[]) {
   FILE* fp;
   char fullFileName[100];
   sprintf(fullFileName, LED3_PATH "%s", filename); // write path and filename
   fp = fopen(fullFileName, "w+"); // open file for writing
   if (fp == NULL) {
      perror("Error opening file");
      return;
   }
   fprintf(fp, "%s", value);  // send the value to the file
   fclose(fp);  // close the file using the file pointer
}

void removeTrigger() {
   writeLED("/trigger", "none");
}
```