Calendar Management System

Intro to C Programming

CSCI – 1110

KAVAD.C

University of New Haven
TAGLIATELA COLLEGE OF ENGINEERING, West Haven, CT

Submitted To:
Dr. Reza Sadeghi

Fall 2020

Final Project Report of Calendar Management System

Team Name

KAVAD.C

Team Members

Amaris Acevedo – Aacev3@unh.newhaven.edu

Ashely Estrella - Cestr1@unh.newhaven.edu

Destiny Ray – Dray2@unh.newhaven.edu

Kaylie Neal - Kneal5@unh.newhaven.edu

Victoria Soto - Vsoto1@unh.newhaven.edu

Roles of Team Members

1. Amaris Acevedo

Team leader, login information and welcome page

2. Destiny Ray

Login and event encryption and decryption, writing report, Menu

3. Kaylie Neal

 Adding and deleting tasks/events, writing report

 4. Victoria Soto

 Editing and searching tasks/events, writing report

 5. Ashley Estrella

 Physical calendar

Table of Contents

Table of Tables

Variables Name	Description
char fname[120], lname[120], username[120], password[120];	Used to create an account
char user[120], pass[120];	Used to help the user login
char fileReader[255];	Used to allow the file to be read
char admin;	Used to understand if it is an admin account or not
FILE * accountInfoPointer;	Pointer for the file

Table 1. The table above is important for signup, login and main menu on page 8.

Variables Name	Description
FILE *fp1, *fp2;	Creates the file pointer to read the file and create new one.
char ch, en;	ch represents the character the program is reading, and en represents its encryption equivalent.

Table 2. The table above includes important variables for encryption.

char durationTime[255];	Allows writing for duration
FILE*fp=NULL;	Creates file pointer
Char eventNote[255]	Writes in event description
char fName[225];	Scans user title; allows for creation of name
char fNameFile[255];	Creates user title with .txt; makes it a text file
char eventStart[30];	Writes in event start
char reminder[255];	Scans user input for reminder
char eventReminder[255];	Writes in event reminder

char eventEnd[30]	Writes in event end time
char newName[255]	Scans user input for title/text file
Char newFile[255]	Renames selected file name to user input
Char eventDescription[255]	Edits event description
Char newReminder[255]	Edits event reminder
Char string1[30]	Searches duration
Char string2[30]	Searches start time
Char string3[30]	Searches end time
Char *months[]	Array of the names of the months
Int year	Variable for the current year
Int month	Puts months into numbers for listing
Int d	Gets the first ever day.
Int day	Lists all the numbered days of the month
Int daysinmonth	Stores the total days of each month
Int monthday	Connects the number of days to their respective month
Int weekday	Splits days into weeks
Int startday	Has each month start on the correct weekday

Table 3. The table above shows the important variables for the calendar.

Functions	Description
goto what;	A goto statement is to jump anywhere that has a function. This jumps to the encryption header file.
void encrypted1(int Encrypt)	Allows the header file of the encryption code to be to calling the function in a different file.
Int firstweekday(int year)	Stores the first ever weekday
Int calendarsetup	Starts the calendar code
Int encrypt(void)	Creates text file with login information and encrypts each letter with a shift of 3

Int encrypt_view(void)	Reads encrypted file and prints it into command prompt
Int decrypt_view(void)	Reads decrypted file and prints it into command prompt
Int okay	Allows for user to go to the calendar

Table 4. The table above shows the list of functions for this program.
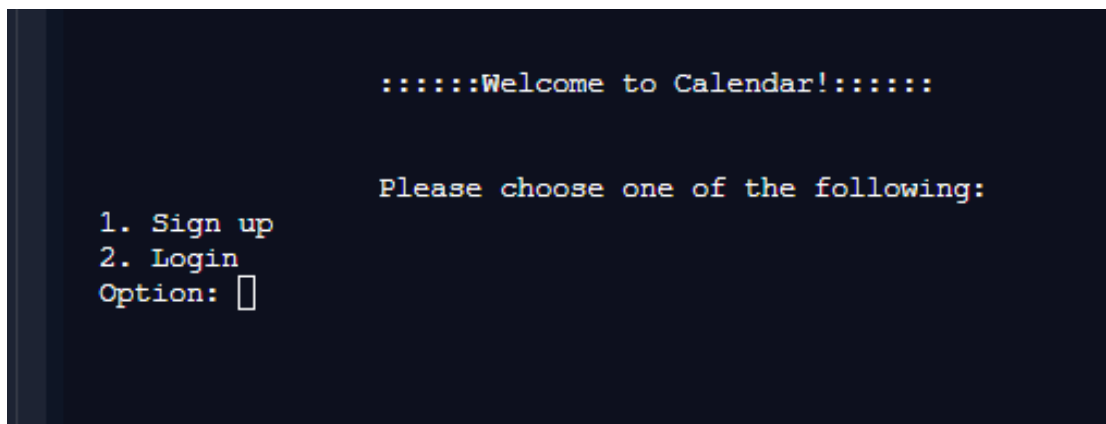
Table of Figures

Figure 1. Figure shows welcome page.

Introduction

The Calendar Management System (CMS) is useful for storing events and their details. This would include the title, duration of the event, start time, end time, reminders, and even a description to include any other notes you have. This system can add, delete, edit, or search for different parts of an event. It also includes a login system with a decryption code and can display the full calendar every month or year. Using the Calendar Management System is extremely beneficial, especially if events easily slip your mind. This system's modules include a login, signup, an illustration of the reports, add a task, remove a task, edit task details, search based on specific details, such as title, duration, time, and encryption.

Amaris Acevedo completed the ability to have an admin username and password, adding a normal user by creating a new username and password. Additionally, the capability to remove CMS users by removing the username, password, and corresponding recorded files. Destiny Ray completed encrypting and decrypting all calendar tasks and login information using Caesar cipher with a shift of 3. She also did the welcome page with the menu of the functions and an exit function. Kaylie Neal completed adding a task function and removing a task function for a title, duration, time, description, and reminders. Victoria Soto completed the ability to edit task details and set up a search option for title, duration, time, description, and reminders. Lastly, Ashley Estrella completed the print calendar.

The following shows the remainder of the report: the project description describes our system modules in detail about the system that's been constructed from login/sign up, storing events, and searching events—also, the relations of the modules to each other. The list of key variables represents the most important variables in our CMS program. Lastly, the references that each KAVAD.C member used to help complete the CMS program.

Project Descriptions

First, the user will be greeted with a welcome page and a menu with a list of functions. The user will be prompted to choose whether the user wants to log in as an admin, user, or create a new account. If the user creates an account, it will be put through the process, and at the end, the user will be asked to encrypt their login information. When this is done, the user will be able to either view the encrypted or decrypted file of their login information or view their calendar. This will then take the user to a list of calendar functions, which are "View Calendar," "Add an event," "Remove Event," "Edit or search event," and "Logout." The user will choose their option by entering the corresponding number associated with it. (put in info about calendar) If "Add event" is chosen, the user will be asked to enter the event title, becoming the text file name. The file will then be created.

The user will then be prompted to enter the date, event description, start and end time, duration, and reminder. Each piece of information will be written into the text file. When "Delete note" is chosen, the user will only be asked to enter the title the user made for the event. The program will scan the name the user inputs and use the sprintf function to add .txt to the end in order to identify the proper file. It will then use the remove function to remove that text file and tell the user whether the file was deleted.

"Edit or search event" is the most extensive option since it contains a menu within the choice. The option begins by asking the user to enter the event title. This will allow the user to edit only that file. The editing options include "Edit title," "Edit duration," "Edit start," "Edit end," "Edit description," and "Edit reminders." Each of the edit tasks works like other tasks by scanning the new input the user provides and writing that into the file to replace the previous data for that option. The only exception to this is "Edit title," which takes the user input for the new title, scans it, and uses the sprintf function to add the .txt to the end of the name. The program then renames the entire file, keeping the rest of the contents the same. The options for searching include "Search title," "Search duration," "Search start," and "Search end."

For "Search title," the user can input the event title, and the program will print the contents of that file into the command prompt. For the start time, end time, and duration, the user can put in the time that the user wish and the program will confirm that it is the correct time for that event. Lastly, option five will allow the user to logout of the calendar with all of their saved information.

List of Key Variables

Important variables use for signup, login, and main menu

Variables Name	Description
char fname[120], lname[120], username[120], password[120];	Used to create an account
char user[120], pass[120];	Used to help the user login
char fileReader[255];	Used to allow the file to be read
char admin;	Used to understand if it is an admin account or not
FILE * accountInfoPointer;	Pointer for the file

Variables Name	Description
FILE *fp1, *fp2;	Creates the file pointer to read the file and create new one.
char ch, en;	ch represents the character the program is reading, and en represents its encryption equivalent.

Important variables use for encryption

Important variables use for the calendar

char durationTime[255];	**Allows writing for duration**
FILE*fp=NULL;	Creates file pointer
Char eventNote[255]	Writes in event description
char fName[225];	Scans user title; allows for creation of name
char fNameFile[255];	Creates user title with .txt; makes it a text file
char eventStart[30];	Writes in event start
char reminder[255];	Scans user input for reminder
char eventReminder[255];	Writes in event reminder
char eventEnd[30]	Writes in event end time
char newName[255]	Scans user input for title/text file
Char newFile[255]	Renames selected file name to user input
Char eventDescription[255]	Edits event description
Char newReminder[255]	Edits event reminder
Char string1[30]	Searches duration
Char string2[30]	Searches start time
Char string3[30]	Searches end time
Char *months[]	Array of the names of the months
Int year	Variable for the current year
Int month	Puts months into numbers for listing
Int d	Gets the first ever day.
Int day	Lists all the numbered days of the month
Int daysinmonth	Stores the total days of each month
Int monthday	Connects the number of days to their respective month
Int weekday	Splits days into weeks
Int startday	Has each month start on the correct weekday

<u>List of functions</u>

Functions	Description
goto what;	A goto statement is to jump anywhere that has a function. This jumps to the encryption header file.
void encrypted1(int Encrypt)	Allows the header file of the encryption code to be to calling the function in a different file.
Int firstweekday(int year)	Stores the first ever weekday
Int calendarsetup	Starts the calendar code
Int encrypt(void)	Creates text file with login information and encrypts each letter with a shift of 3
Int encrypt_view(void)	Reads encrypted file and prints it into command prompt
Int decrypt_view(void)	Reads decrypted file and prints it into command prompt
Int okay	Allows for user to go to the calendar

References

https://github.com/debasree888/calendar-in-c/blob/master/Calendar-in-C-master/main.c

https://stackoverflow.com/questions/1782080/what-is-eof-in-the-c-programming-language

https://www.youtube.com/watch?v=8Aiy1q1SsqA&feature=emb_title

https://www.geeksforgeeks.org/c-program-to-append-content-of-one-text-file-to-another/

https://www.researchgate.net/publication/337195393_C-PROGRAMMING-REPORT_AIRLINE_RESERVATION_SYSTEM

https://www.academia.edu/34215145/C_Programming_Full_Project_Documentation

https://www.w3resource.com/c-programming-exercises/file-handling/c-file-handling-exercise-13.php

https://codescracker.com/c/program/c-program-encrypt-file.htm

https://www.dummies.com/programming/c/basics-of-global-variables-in-c-programming/

https://www.tutorialspoint.com/cprogramming/c_do_while_loop.htm

https://www.dummies.com/programming/c/basics-of-files-in-c-programming/

https://www.dummies.com/programming/c/how-to-create-a-custom-header-file-in-c/

https://dyclassroom.com/c/c-file-handling-read-and-write-multiple-data

https://www.tutorialspoint.com/c_standard_library/c_function_remove.htm

https://www.tutorialspoint.com/c_standard_library/c_function_rename.htm

https://github.com/Bibeknam/programming-techniques/blob/master/Calendar/calendar.c

https://www.codingalpha.com/file-handling-code-to-encrypt-and-decrypt-c-program-text-files/

TA: Ashish Shrestha

Professor: Reza Sadeghi