|  |  |  |
| --- | --- | --- |
| Assignment Code | : | C.L.P0025 |
| Assignment Name | : | Simulate ATM’s operation |
| Student Name | : | Le Thi Thanh Nhan |
| Time/Date | : | 2h00,22 /11/2019 |

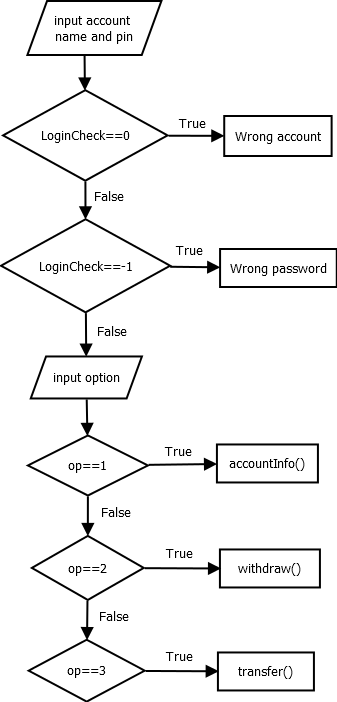
Approach

Use 4 files to store account information, login, withdraw and transfer.

Input account name and pin, read file login and check them correctly or not.

* Account info: read file information, find the account and print out information.
* Withdraw: input money to withdraw and update the money in file infotmation and write withdraw transactions in file withdraw.
* Transfer: input account and money to transfer and update the money in file infotmation and write transfer transactions in file transfer.

Flowchart



Source code

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAX 10

int LoginCheck(char \*account, char \*pin)

{

    char data[99];

    char \*temp = (char \*)malloc(99 \* sizeof(char));

    FILE \*f = fopen("loginInfo.txt", "r");

    do

    {

        fgets(data, 99, f);

        temp = strtok(data, " ");

        if (strcmp(temp, account) == 0)

        {

            temp = strtok(NULL, " ");

            if (temp[strlen(temp) - 1] == '\n')

                temp[strlen(temp) - 1] = '\0';

            if (strcmp(temp, pin) == 0)

                return 1;

            else

                return -1;

        }

    } while (data[0] != '\0' && data[0] != '\n');

    return 0;

}

int AccountIdExist(char \*accountId, char \*accountName, char \*accountBalance, char \*moneyType)

{

    char tempData[99];

    char x[2] = " ";

    char \*temp = (char \*)malloc(14 \* sizeof(char));

    fflush(stdin);

    FILE \*f = fopen("accountInfo.txt", "r");

    int i = 0;

    while (fgets(tempData, 99, f) && tempData[0] != '\n' && tempData[0] != '\0')

    {

        temp = strtok(tempData, x);

        if (strcmp(temp, accountId) == 0)

        {

            temp = strtok(NULL, x);

            strcpy(accountName, temp);

            temp = strtok(NULL, x);

            strcpy(accountBalance, temp);

            temp = strtok(NULL, x);

            strcpy(moneyType, temp);

            fclose(f);

            return i;

        }

        i += 1;

    }

    if (moneyType[strlen(moneyType) - 1] == '\n')

        moneyType[strlen(moneyType) - 1] = '\0';

    fclose(f);

    return -1;

}

void FileUpdate(char \*fileName, char \*type, char data[][99], int stt)

{

    FILE \*f = fopen(fileName, type);

    for (int i = 0; i < stt; i += 1)

        if (data[i][0] != '\0' && data[i][0] != '\n')

            fputs(data[i], f);

    printf("Updated!\n");

    fclose(f);

}

int FileGetData(char \*fileName, char data[][99])

{

    FILE \*f = fopen(fileName, "r");

    int count = 0;

    do

    {

        data[count][0] = '\0';

        fgets(data[count], 99, f);

        count += 1;

    } while (data[count - 1][0] != '\n' && data[count - 1][0] != '\0');

    return count - 1;

}

void accountInfo(char \*accountId)

{

    int locate;

    char accountName[21];

    char accountBalance[21];

    char moneyType[7];

    int stt = AccountIdExist(accountId, accountName, accountBalance, moneyType);

    if (stt == -1)

        printf("Not exist!\n");

    else

    {

        printf("Account ID: %s\n", accountId);

        printf("Account name: %s\n", accountName);

        printf("Account balance: %s\n", accountBalance);

        printf("Money type: %s\n", moneyType);

    }

}

void withdraw(char \*accountId)

{

    char data[MAX][99];

    accountId[strlen(accountId)] = '\0';

    int i = FileGetData("accountInfo.txt", data);

    char accountBalance[20];

    char accountName[20];

    char moneyType[8];

    int index = AccountIdExist(accountId, accountName, accountBalance, moneyType);

    printf("Enter money to withdraw: ");

    int withdraw;

    scanf("%d", &withdraw);

    if (withdraw > atoi(accountBalance))

        printf("Over your balance!\n");

    else

    {

        char tempWithdraw[2][99];

        sprintf(tempWithdraw[0], "%s %s %d %d\n", accountId, accountBalance, withdraw, atoi(accountBalance) - withdraw);

        sprintf(data[index], "%s %s %d %s\n", accountId, accountName, atoi(accountBalance) - withdraw, moneyType);

        FileUpdate("withdrawInfo.txt", "a", tempWithdraw, 1);

        FileUpdate("accountInfo.txt", "w+", data, i);

    }

}

void transfer(char \*accountId)

{

    char accountName[20], accountBalance[20], moneyType[8];

    char targetId[14], targetName[20], targetBalence[20], targetMoneyType[8];

    char data[MAX][99];

    int accountIndex = AccountIdExist(accountId, accountName, accountBalance, moneyType);

    if (moneyType[strlen(moneyType) - 1] == '\n')

        moneyType[strlen(moneyType) - 1] = '\0';

    printf("Enter other accountId to transfer: ");

    gets(targetId);

    FILE \*fr = fopen("accountInfo.txt", "r");

    int stt = 0;

    do

    {

        data[stt][0] = '\0';

        fgets(data[stt], 99, fr);

        stt += 1;

    } while (data[stt - 1][0] != '\0' && data[stt - 1][0] != '\n');

    fclose(fr);

    stt -= 1;

    char tempId[14];

    int targetIndex = AccountIdExist(targetId, targetName, targetBalence, targetMoneyType);

    if (targetIndex == -1)

        printf("Not found id %s\n", targetId);

    else

    {

        if (targetMoneyType[strlen(targetMoneyType) - 1] == '\n')

            targetMoneyType[strlen(targetMoneyType) - 1] = '\0';

        int balance = atoi(accountBalance);

        int money;

        printf("Your balance is: %d\n", balance);

        printf("Enter value to transfer money: ");

        scanf("%d", &money);

        if (money > balance)

            printf("Over your balance!\n");

        else

        {

            balance -= money;

            sprintf(data[accountIndex], "%s %s %d %s\n", accountId, accountName, balance, moneyType);

            sprintf(data[targetIndex], "%s %s %d %s\n", targetId, targetName, atoi(targetBalence) + money, targetMoneyType);

            FileUpdate("accountInfo.txt", "w+", data, stt);

            sprintf(data[0], "%s %s %d\n", accountId, targetId, money);

            FileUpdate("transferInfo.txt", "a", data, 1);

        }

    }

}

int main()

{

    char accountId[15];

    char pin[7];

    printf("Login:\n");

    printf("Enter account: ");

    gets(accountId);

    fflush(stdin);

    printf("Enter pin: ");

    gets(pin);

    fflush(stdin);

    int stt = LoginCheck(accountId, pin);

    if (stt == 0)

        printf("Wrong account\n");

    else if (stt == -1)

        printf("Wrong password\n");

    else

    {

        int option;

        char accountName[20], accountBalance[20], moneyType[8];

            printf("\n1. Account info");

            printf("\n2. Withdraw");

            printf("\n3. Transfer");

            printf("\n4. Exit");

            do

        {

            printf("\nEnter your option:");

            scanf("%d", &option);

            switch (option)

            {

            case 1:

                accountInfo(accountId);

                break;

            case 2:

                withdraw(accountId);

                break;

            case 3:

                transfer(accountId);

            }

        } while (option != 4);

    }

}

Result

