

A program egy parancssor alapú közösségi terminál mely hálózat nélkül működik és egyszerre egy felhasználó tudja igénybe venni. Jelenleg támogatott funkciók a keret menürendszer és a lehetőség, hogy egy felhasználó regisztrálni tudjon a rendszerben és a fiókjába be tudjon lépni.

Fájl lista betűszerinti sorrendben:

Forrásfájlok:

- contact_functions.c – Ismerősökkel kapcsolatos funkciók
- econio.c – Szöveges megjelenítéshez tartozó függvények
- formatting_functions.c – Szövegformálással kapcsolatos funkciók
- list_functions.c – Láncolt listákkal kapcsolatos funkciók
- main.c – A program fő függvénye, jelenleg minden funkciót ez tartalmaz
- message_functions.c – Üzenetekkel kapcsolatos funkciók
- user_functions.c – A felhasználóval kapcsolatos funkciók

Fejlécfájlok

- contact_functions.h
- econio.h
- formatting_functions.h
- list_functions.h
- message_functions.h
- structures.h
- user_functions.h

Függvények funkciói, struktúrák:

main.c

```
typedef struct Address{  
    char country[51];  
    char city[51];  
    char street[51];  
    int house;  
} Address;
```

Felhasználó címét tároló struktúra.

```
typedef struct User{  
    char Username[21];  
    char Password[21];  
    char Name[51];  
    char Sex[7];
```

```
    char Birth[12];
    Address Useraddress;
    char Phonenum[11];
    char School[51];
    char Work[51];
} User;
```

A felhasználó összes adatát egybefoglaló struktúra.

contact_functions.c

void add_contact(char *file, char *filename, int state)

Más felhasználók ismerősnek jelöléséért felel. Fájlból olvassa be a felhasználókat és fájlba menti a kapcsolati listát. Paraméterként kapja a felhasználók listájának fájlnevét, a jelenlegi felhasználó adatait tartalmazó fájl nevét és egy állapot változót.

void check_contact(char *filename, int state)

Ezzel a függvénnyel lehet megtekinteni más felhasználók kapcsolati listáját. Fájlból olvas be. Paraméterként kapja a keresett fájlnevet és egy állapot változót.

void search(LinkNode *first, int usercount, User *current, LinkNode *result, char *filename)

Ez a függvény végzi a keresést a felhasználók listájában megadott paraméterek szerint. A felhasználókat először egy láncolt listába olvassa, majd az eredményeket egy fájlba menti. Paraméterként kapja a láncolt listát, a felhasználók számát, a jelenlegi felhasználót, az eredmények láncolt listáját és a keresett fájlnevet.

char* check_info(int idx, LinkNode *list, int state)

Ez a függvény egy láncolt listában megkeresi a megadott felhasználót és kilistázza a személyes adatait. Paraméterként kapja a keresett felhasználó indexét, a láncolt listát és egy állapot változót. Visszatérési értéke a keresett felhasználó felhasználó neve.

formatting_functions.c

void remove_n(char *str)

A paraméterként kapott szövegből eltávolítja a sor vége jelet.

list_functions.c

LinkNode *load_userbase(char *file, int count)

Egy láncolt listába beolvassa az összes felhasználót aki regisztrálva van a program adatbázisában. Paraméterként kapja az olvasandó fájl nevét és a benne lévő sorok számát. Visszatérése értéke a lista első eleme.

void free_list(LinkNode *first)

Felszabadítja a láncolt listát melynek első elemét paraméterként megkapja.

main.c

int main()

Tartalmazza a fő menü rendszert, ellenőrzi, hogy bejelentkezett-e a felhasználó és switch-ekkel éri el a program többi funkcióját.

message_functions.c

void messages(int messagecount, int usercount, User *current)

Az üzeneteket kezelő menü rendszer, innen érhető el az üzenetek olvasásának és írásának a lehetősége. Paraméterként kapja az üzenetek számát, a felhasználók számát és a jelenlegi felhasználó adatait.

void read_message(int usercount, int messagecount, User *current)

A jelenleg bejelentkezett felhasználónak címzett üzeneteket keresi meg és jeleníti meg a szabványos kimeneten. Paraméterként kapja az üzenetek számát, a felhasználók számát és a jelenlegi felhasználó adatait.

void send_message(int messagecount, int usercount)

Ezzel a függvénnyel lehet üzeneteket írni más felhasználóknak. Fájlba menti az üzeneteket melyek neve tartalmazza a címzett felhasználó nevét, az üzenet azonosítóját, megjelölést, hogy a fájl egy üzenet és a fájlban belül az üzeneten kívül a küldő neve is jelezve van. Paraméterként kapja az üzenetek számát és a felhasználók számát.

user_functions.c

char* login(bool *loggedon)

A függvényen keresztül történik egy profilba való bejelentkezés. Visszatér a felhasználó adatait tartalmazó fájl nevével. Paraméterként kap egy bool változót mely jelzi, hogy be van-e jelentkezve egy felhasználó. Visszatérési értéke a bejelentkezett felhasználó adatait tartalmazó fájlnak a neve.

void registration()

A regisztrációt kezelő függvény. A felhasználó által megadott adatokat felhasználva létrehoz egy profilt melyet egy szöveges fájlban tárol. Minden profilt külön fájlba ment a felhasználó nevet használva mint cím. Ellenőrzi, hogy foglalt-e már a felhasználónév.

void infoedit(char *filename)

Ezzel a függvénnyel tudja a felhasználó megváltoztatni a személyes adatait regisztráció után. Fájlból olvas és ugyanabban a fájlban változtatja meg az információt. Paraméterként kap a felhasználó adatait tartalmazó fájl nevét.

Fejlesztő környezet:

Nyelv: C

Program: Code::Blocks 17.12

Compiler: MinGW