

Nagy házfeladat 2021

- Határidőnapló-

Programozói dokumentáció

A határidőnapló program lényege az, hogy eseményeket lehet létrehozni fájlok formájában és azokkal különböző műveleteket végezni. Egy esemény fő adatai a következők: név, helyszín, megjegyzés, dátum, óra és perc pontos idő.

A program építése:

A programot egyaránt lehet Windows illetve UNIX alapú operációs rendszereken építeni **gcc**-vel a következőképpen: **gcc main.c fuggvenyek.c folder.c keres.c modosit.c listaz.c -o hataridonaplo**. Ezután meg futtatni a **./hataridonaplo** paranccsal.

Adatszerkezetek választása:

Egy esemény fő adatait az alábbi módon strukturában tárolom, ez megkönnyíti a felesleges változóknak a felvételét, illetve egyes függvényeknél a visszatérítés is könnyebb.

```
typedef struct Rekord {  
    char *nev;  
    char *helyszin;  
    char *megjegyzes;  
    char *datum;  
    int ora, perc;  
} Rekord;
```

Az esemény pontos idején kívül minden dinamikusan van foglalva, tehát akármilyen hosszú lehet egy fájlnev, megjegyzés vagy helyszín. A struktúrához tartozik még egy függvény: **void RekordFree(Rekord r)** amely paraméterként kap egy Rekord típusú változót és felszabadítja annak minden dinamikusan foglalt elemét.

Az események közti keresésre van létrehozva még egy struktúra, amely láncolt listáknak a létrehozásához illetve kereséséhez szükséges.

```
typedef struct SzoLista {  
    char *szo;  
    struct SzoLista *kov;  
} SzoLista;
```

Az ehhez tartozó függvények:

```
SzoLista *ujegyszavas(char const *szo);
SzoLista *keres(SzoLista *l, char *mit);
void lista_free(SzoLista *lista);
void EsemenyKeres(char *s);
```

A függvények sorba a következőket csinálják:

- Paraméterként kap egy stringet és egy láncolt listát hoz létre azzal
- Paraméterként kapott láncolt listában megkeres egy paraméterként kapott stringet
- Felszabadítja a paraméterként kapott listát
- Az előző három függvényt rakja össze egy függvénybe

Függvények:

A **void menuopciok()** illetve **void menu()** függvények nem kapnak paraméterül semmit, csupán a megjelenítésre és a többi függvényhívására vannak használva.

char *dintext() nevű függvény egy dinamikusan foglalt stringet hoz létre, amelyet később, a függvény hívása után kell felszabadítani.

void esemenyFeltolt() bekéri a felhasználótól az esemény adatait és fájlba írja azokat.

void torol(char *s) egy esemény nevét kapja meg paraméterként és törli azt a rekordok mappából.

void fajlkilistaz() nem kap semmit paraméterként, kilistázza az összes fájlnak ezzel egybe az események nevét ami a rekordok mappában megtalálható.

void modmenuop() hasonlóan a **menuopciok()** – hoz ez is megjeleníti az opciókat a módosításhoz.

void modmenu(char *s) paraméterként kap egy eseménynevet és kezeli annak a módosításait.

Rekord fileertek(char *s) egy eseményt nyit meg és olvassa be egy struktúrába az adatokat amelyet aztán visszatérít.

void honap(), void nap(), void het() mindhárom függvény arra van, hogy összehasonlítsa, hogy ugyanabban a hónapban, napon vagy héten van e esemény.

char *reszstring(char const *eredeti, int mettol, int meddig) egy stringet állít elő, kivágva belőle a karaktereket a megadott indextől a másik indexig.

int hetnap(int y, int m, int d) dátumból előállítja, hogy milyen napra esik az, visszatér egy számmal 1 -> Hétfő..

void kilistaz(char *s) paraméterként megkap egy dátumot és az alapján keresi meg, hogy arra a dátumra és közelébe milyen események esnek.

Források:

- <https://infoc.eet.bme.hu/>
- https://www.clipartmax.com/download/m2i8b1A0Z5d3A0N4_512-x-512-timetable-icon/
- <https://cs.uwaterloo.ca/~alopez-o/math-faq/node73.html>