

# Határidőnapló dokumentáció

A programot Makefile segítségével fordítjuk. A fordításhoz elengedhetetlen egy C fordító. GCC 13.3.1 használatával lefordul a program hiba nélkül. A "make" parancs egyszeri lefuttatása után megjelenik a lefordult program "main" néven. Parancssorban a "./main" karaktersorozat lefuttatásával el tudjuk indítani a frissen generált programot. Módosítások után érdemes újrafordítani a forráskódot, hogy frissüljön a futtatható program.

## Precompile hasznosságok

- **DEBUG**  
Értéke 0 vagy 1 (igaz-hamis). debug.h fájlban definiálva. Ha be van kapcsolva, debugmalloc.h keretrendszert használja memóriavesztés-ellenőrzéshez.
- **MENU\_DEFINITION**  
Makró. A menu/menu.h fájlban menü függvények deklarálásához.
- **MENU**  
Makró. C forrásfájlokban menük definiálásához. Automatikusan csinál while ciklust.
- **CLEAR\_SCREEN**  
Makró. ASCII escape karaktersorozattal törli a parancssor/terminál kimenetét.
- **WAIT\_FOR\_ENTER**  
Makró. Menüben használatos, Enter gombnyomásra vár a menüciklusban.
- **VERSION**  
Makró. main.h fájlban definiálva, a program verzióját tartalmazza

## Struktúrák, konstansok

- **Date**  
Évek, hónapok, napok számát tároló struktúra.
- **Time**  
Órákat, percekét tároló struktúra.
- **Event**  
Az adatbázisban tárolt/tárolandó eseményeket reprezentálja ez a struktúra. Tárol dátumot, időt, címet, helyet, leírást.
- **EventListNode**  
Esemény láncolt lista elem. ID-t, eseményt és a következő elemre mutató pointert tárolja.
- **State**  
A program állapotát tartalmazza. Tárolja az adatbázisfájl nevét, menü argumentumokat és a láncolt listát, ami az eseményeket tárolja.
- **EventMenuData**  
Menüadatok egy esemény szerkesztéséhez. Tartalmazza az esemény azonosítóját.
- **AddMenuData**  
Menüadatok esemény hozzáadásához. Tartalmaz egy Event struktúrát.
- **ListMenuData**  
Menüadatok a listázó menühöz. Keresési szöveget tárol.

## Függvények, eljárások

Menük nem kerülnek ide, ugyanúgy működik az összes.

- **int main(int argc, char \*\*argv)**  
A program kezdőpontja. Beolvassa a parancssorból a paramétereket, flageket és aszerint indítja a programot. Inicializálja az állapotot, betölti az adatbázist, megkezdi a menüvezérlést, majd kilépés előtt rendezi és menti az eseménylistát.
- **void print\_help()**  
Kiírja a help menüt. A "--help" vagy "-h" flaggel lehet program indításakor ezt előhozni.
- **int parse\_date(Date \*d, char \*str)**  
Dátumot állít elő egy szövegből (ÉÉÉÉ-HH-NN formátum).
- **int parse\_time(Time \*t, char \*str)**  
Időt állít elő egy szövegből (óó:pp formátum).
- **char\* date\_to\_str(Date d)**  
Dátumot alakít szöveggé.
- **char\* time\_to\_str(Time t)**  
Időt alakít szöveggé.
- **int validate\_date(Date \*d)**  
Dátumot ellenőriz helyességre.
- **int validate\_time(Time \*t)**  
Időt ellenőriz helyességre.
- **void free\_event\_list(EventListNode \*head)**  
Egy dinamikusan allokált esemenyláncoltlistát szabadít fel.
- **int restore\_state(State \*state, char \*filename)**  
A program állapotát állítja vissza egy CSV fájlból fájlnev alapján.
- **int save\_state(State \*state)**  
Elmenti az eseménylistát CSV fájlba.
- **Event\* find\_event\_by\_index(EventListNode \*head, int index)**  
Megkeres egy eseményt az indexe alapján a láncolt listában.
- **int remove\_node(EventListNode \*\*head, int index)**  
Töröl egy elemet a láncolt listából index alapján.
- **int add\_event(EventListNode \*head, Event e)**  
Hozzáad egy eseményt a láncolt listához.
- **void reindex(EventListNode \*head)**  
Újraindexeli a láncolt listát.
- **void sort\_event\_list(EventListNode \*\*head\_ptr)**  
Rendezi az eseménylistát időrendi sorrendbe dátum és idő szerint.
- **void remove\_newl(char \*str)**  
Eltávolítja az újsor karaktert egy szövegből.
- **void flush()**  
Kiüríti a standard input buffert.

- `int prompt_date_into_dest(Date *dest)`  
Bekér egy dátumot a felhasználótól és a megadott címre írja.
- `int prompt_time_into_dest(Time *dest)`  
Bekér egy időt a felhasználótól és a megadott címre írja.
- `int prompt_str_into_dest(char **dest)`  
Bekér egy szöveget a felhasználótól és a megadott címre írja.
- `void print_event(Event *event)`  
Kiír egy eseményt formázottan a képernyőre.

## Menük

---

- `menu_main`  
Főmenü. Innen lehet navigálni a többi menübe.
- `menu_list`  
Eseménylista megjelenítő menü keresési funkcióval.
- `menu_event`  
Egyedi esemény szerkesztő menü.
- `menu_add`  
Új esemény hozzáadó menü.
- `menu_weeks`  
Heti nézet menü.
- `menu_months`  
Havi nézet menü.