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.

Tervezési megfontolások

A program magja egy láncolt lista, ami tartalmazza az eseményeket. Ez egy dinamikus adatszerkezet, ami megadja a program működéséhez szükséges rugalmasságot. A menükezelés egyszerű egymásban meghívott függvényekkel van megoldva. Meg lehetett volna oldani függvénypointerekkel is, egy külön menükezelő rendszerrel. Utólag nem ez a megoldás lett választva, mert egyszerűbb megoldásnak tűnt az előbbi.

Forrásfájlok

Menük dokumentálásra nem kerül sor, triviális.

- debug.h
  Importálja a debugmallocot, további funkciókat biztosít a fejlesztés megkönnyítéséhez. Pl DEBUG érték.
- debugmalloc.h
  Memóriabiztonsági makró mágia.
- main.h, main.c
  A program fő része. Itt kezdődik minden.
- menu.h, menu.c
  Menük által használt közös funkcionalitás definiálására szolgáló forrásfájlok.
- state.h, state.c
  A program futása során használt állapot kezelésére felhasznált struktúrákat, függvényeket tartalmazó fájlok.
- util.h, util.c
  Apró "hasznosságok" tárháza. Ide kerülnek azok a függvények, amik nagyon általánosak és nem feltétlen van helyük máshol.

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, perceket 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.
- MonthInfo
  Hónap láncolt lista elem. Tartalmaz évet, hónapot, következő elemre mutató pointert.
- MonthMenuData
  Menüadatok a hónapválasztós menühöz. Tartalmaz évet és hónapot.
- WeekInfo
  Hét láncolt lista elem. Tartalmaz évet, hogy hanyadik hét, következő elemre mutató pointert.
- WeekMenuData
  Menüadatok a hétválasztós menühöz. Tartalmaz évet és a hét számát.

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.
- char* alloc_optional_str(char* str)
  Segédfüggvény esemény szövegkezeléséhez. Dinamikusan foglal memóriaterületet a bemeneti szöveg alapján. Ha NULL, üres (\0) sztringet foglal.
- 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 eseménylláncoltlistát szabadít fel.
- int restore_state(State *state, char *filename)
  A program állapotát állítja vissza egy CSV fájlból fájlnév 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.
- char *get_month_name(int month)
  Visszaadja az adott hónapszámhoz tartozó hónap nevét.
- free_month_list(MonthInfo *head)
  Felszabadít egy hónapokat tartalmazó láncolt listát.
- int month_exists(MonthInfo *head, int year, int month)
  Megmondja, hogy egy hónap láncolt listában létezik-e az adott hónap-év kombináció.
- void add_month_if_new(MonthInfo **head, int year, int month)
  Hozzáadja a hónap láncolt listához a megadott hónap-év kombinációt, ha még nincs benne.
- int get_week_number(Date d)
  Visszaadja, hogy hanyadik hétben van az adott dátum.
- free_week_list(WeekInfo *head)
  Felszabadít egy heteket tartalmazó láncolt listát.
- int week_exists(WeekInfo *head, int year, int week)
  Megmondja, hogy egy hét láncolt listában létezik-e az adott hét-év kombináció.
- void add_week_if_new(WeekInfo **head, int year, int week)
  Hozzáadja a hét láncolt listához a megadott hét-év kombinációt, ha még nincs benne.

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.