

calendar

Generated by Doxygen 1.8.13

Contents

1	Határidőnapló	1
2	Module Index	3
2.1	Modules	3
3	Data Structure Index	5
3.1	Data Structures	5
4	File Index	7
4.1	File List	7
5	Module Documentation	9
5.1	Események részletei	9
5.1.1	Detailed Description	9
5.1.2	Typedef Documentation	9
5.1.2.1	ModBy	9
5.1.3	Enumeration Type Documentation	10
5.1.3.1	ModBy	10
5.1.4	Function Documentation	10
5.1.4.1	deleteevent()	10
5.1.4.2	moveevent()	10
5.1.4.3	printeventrecord()	11
5.1.4.4	scanrecordcommand()	11
5.2	File kezelő	12
5.2.1	Detailed Description	12

5.2.2	Function Documentation	12
5.2.2.1	calendarload()	12
5.2.2.2	calendarsave()	12
5.2.2.3	dstrcpy()	13
5.3	Lista kezelő	14
5.3.1	Detailed Description	14
5.3.2	Typedef Documentation	14
5.3.2.1	Tm	14
5.3.3	Function Documentation	14
5.3.3.1	createevent()	15
5.3.3.2	free_event()	15
5.3.3.3	freeEventList()	16
5.3.3.4	initeventlist()	16
5.3.3.5	insertevent()	16
5.3.3.6	starttime()	16
5.4	Menü kezelő	18
5.4.1	Detailed Description	18
5.4.2	Function Documentation	18
5.4.2.1	exitmenu()	18
5.4.2.2	filesave()	18
5.4.2.3	hosszu_sort_olvas()	19
5.4.2.4	mainmenu()	19
5.4.2.5	newevent()	19
5.4.2.6	printmenu()	20
5.4.2.7	savemenu()	20
5.4.2.8	scanexitmenu()	20
5.4.2.9	scanmainmenu()	20
5.4.2.10	scansavemenu()	21
5.4.2.11	searchmenu()	21
5.5	Keresés kezelő	22

5.5.1	Detailed Description	22
5.5.2	Typedef Documentation	22
5.5.2.1	SearchBy	23
5.5.3	Enumeration Type Documentation	23
5.5.3.1	SearchBy	23
5.5.4	Function Documentation	23
5.5.4.1	eventtotm()	23
5.5.4.2	inserttofindlist()	24
5.5.4.3	printevent_short()	24
5.5.4.4	printfindlist()	24
5.5.4.5	scanfindlist()	24
5.5.4.6	scaninput()	25
5.5.4.7	searchactual()	25
5.5.4.8	searchbyday()	25
5.5.4.9	searchbymonth()	26
5.5.4.10	searchbyname()	26
5.5.4.11	searchbyweek()	26
5.5.4.12	searchiter()	27
5.5.4.13	tmtoweeek()	27
5.6	Szükséges struktúrák	28
5.6.1	Detailed Description	28
5.6.2	Typedef Documentation	28
5.6.2.1	Event	28
5.6.2.2	EventList	28
5.6.2.3	FindList	28
5.6.2.4	FoundEvent	29
5.6.2.5	MenuPont	29
5.6.2.6	SearchConditions	29
5.6.2.7	Tm	29

6	Data Structure Documentation	31
6.1	Event Struct Reference	31
6.1.1	Detailed Description	31
6.1.2	Field Documentation	31
6.1.2.1	comment	31
6.1.2.2	day	32
6.1.2.3	endhour	32
6.1.2.4	endmin	32
6.1.2.5	location	32
6.1.2.6	month	32
6.1.2.7	name	32
6.1.2.8	next	32
6.1.2.9	prev	32
6.1.2.10	starthour	33
6.1.2.11	startmin	33
6.1.2.12	year	33
6.2	EventList Struct Reference	33
6.2.1	Detailed Description	33
6.2.2	Field Documentation	33
6.2.2.1	first	33
6.2.2.2	last	34
6.3	FindList Struct Reference	34
6.3.1	Detailed Description	34
6.3.2	Field Documentation	34
6.3.2.1	first	34
6.3.2.2	last	34
6.4	FoundEvent Struct Reference	35
6.4.1	Detailed Description	35
6.4.2	Field Documentation	35
6.4.2.1	foundevent	35

6.4.2.2	nextfound	35
6.4.2.3	prevfound	35
6.5	Lefoglalt Struct Reference	36
6.5.1	Field Documentation	36
6.5.1.1	cel	36
6.5.1.2	file	36
6.5.1.3	fv	36
6.5.1.4	line	36
6.5.1.5	meret	36
6.5.1.6	next	37
6.5.1.7	prev	37
6.5.1.8	usernek	37
6.5.1.9	valodi	37
6.6	MenuPont Struct Reference	37
6.6.1	Detailed Description	37
6.6.2	Field Documentation	37
6.6.2.1	nev	38
6.7	SearchConditions Struct Reference	38
6.7.1	Detailed Description	38
6.7.2	Field Documentation	38
6.7.2.1	day	38
6.7.2.2	month	38
6.7.2.3	name	39
6.7.2.4	week	39
6.7.2.5	year	39

7 File Documentation	41
7.1 /home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/debugmalloc.c File Reference	41
7.1.1 Macro Definition Documentation	42
7.1.1.1 TABLA_OSZLOPOK	42
7.1.2 Typedef Documentation	42
7.1.2.1 Lefoglalt	42
7.1.3 Function Documentation	42
7.1.3.1 debugmalloc_calloc()	42
7.1.3.2 debugmalloc_dump()	42
7.1.3.3 debugmalloc_free()	42
7.1.3.4 debugmalloc_free_full()	43
7.1.3.5 debugmalloc_hash()	43
7.1.3.6 debugmalloc_malloc()	43
7.1.3.7 debugmalloc_malloc_full()	43
7.1.3.8 debugmalloc_naplofajl()	43
7.1.3.9 debugmalloc_realloc()	43
7.1.3.10 debugmalloc_realloc_full()	44
7.2 /home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/debugmalloc.h File Reference	44
7.2.1 Macro Definition Documentation	44
7.2.1.1 calloc	44
7.2.1.2 free	45
7.2.1.3 malloc	45
7.2.1.4 realloc	45
7.2.2 Function Documentation	45
7.2.2.1 debugmalloc_calloc()	45
7.2.2.2 debugmalloc_dump()	45
7.2.2.3 debugmalloc_free()	45
7.2.2.4 debugmalloc_free_full()	46
7.2.2.5 debugmalloc_malloc()	46

7.2.2.6	debugmalloc_malloc_full()	46
7.2.2.7	debugmalloc_naplofajl()	46
7.2.2.8	debugmalloc_realloc()	46
7.2.2.9	debugmalloc_realloc_full()	46
7.3	/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/eventrecord.c File Reference	47
7.4	/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/eventrecord.h File Reference	47
7.5	/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/file.c File Reference	48
7.6	/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/file.h File Reference	48
7.7	/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/list.c File Reference	48
7.8	/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/list.h File Reference	49
7.9	/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/main.c File Reference	49
7.9.1	Function Documentation	49
7.9.1.1	main()	49
7.10	/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/menu.c File Reference	50
7.11	/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/menu.h File Reference	50
7.12	/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/search.c File Reference	51
7.13	/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/search.h File Reference	51
7.14	/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/searchui.c File Reference	52
7.15	/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/searchui.h File Reference	52
7.16	/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/structures.h File Reference	52

Chapter 1

Határidőnapló

A határidőnapló program képes események létrehozására, keresésére, módosítására és törlésére. Az események duplán láncolt listában vannak tárolva, két üres strázsával, időben növekvő sorrendben rendezve (lásd: struktúrák). A keresésnél időben visszafelé haladunk, így kevesebb iterációt kell végezni feltételezve, hogy a felhasználó "közele eseményekre keres". Módosításkor ha az esemény kezdő idejét módosítjuk, a program áthelyezi a láncolt listában az eseményt. Szöveges infó módosításánál csak kicseréljük a stringet A háttérműködést vezérlő modulok: file-, lista-, keresés-kezelő, és a struktúrák A felhasználói felülethez kapcsolódó modulok: események részletei (itt lehet az eseményt módosítani is), menü kezelő, keresés-menü kezelő.

Chapter 2

Module Index

2.1 Modules

Here is a list of all modules:

Események részletei	9
File kezelő	12
Lista kezelő	14
Menü kezelő	18
Keresés kezelő	22
Szükséges struktúrák	28

Chapter 3

Data Structure Index

3.1 Data Structures

Here are the data structures with brief descriptions:

Event	31
EventList	33
FindList	34
FoundEvent	35
Lefoglalt	36
MenuPont	37
SearchConditions	38

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/ debugmalloc.c	41
/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/ debugmalloc.h	44
/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/ eventrecord.c	47
/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/ eventrecord.h	47
/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/ file.c	48
/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/ file.h	48
/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/ list.c	48
/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/ list.h	49
/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/ main.c	49
/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/ menu.c	50
/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/ menu.h	50
/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/ search.c	51
/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/ search.h	51
/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/ searchui.c	52
/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/ searchui.h	52
/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/ structures.h	52

Chapter 5

Module Documentation

5.1 Események részletei

Typedefs

- typedef enum [ModBy](#) [ModBy](#)

Enumerations

- enum [ModBy](#) { [bydate](#), [bystart](#), [byend](#) }

Functions

- void [moveevent](#) ([Event](#) *event, [EventList](#) const *eventlist, [ModBy](#) modby)
- void [deleteevent](#) ([Event](#) *event)
- int [scanrecordcommand](#) (bool isnewevent, int i, [Event](#) *event, [EventList](#) const *eventlist)
- int [printeventrecord](#) ([Event](#) *event, [SearchConditions](#) condition, [EventList](#) *eventlist)

5.1.1 Detailed Description

Események részleteinek kiírásával és módosításával kapcsolatos függvények Az eseményrekord adott eleméhez tartozó szám lenyomásával módosítani tudjuk az adott elemet, illetve meghívhatunk más menüpontot. Ha az esemény kezdő idejét szeretnénk módosítani, át kell helyezni az eseményt, egyébként elég az adott adat cseréje

5.1.2 Typedef Documentation

5.1.2.1 ModBy

```
typedef enum ModBy ModBy
```

Hogyan szeretnénk módosítani az eseményt: bydate: dátum szerint bystart: kezdőidő (óra, perc) szerint byend: befejezőidő (óra, perc) szerint

5.1.3 Enumeration Type Documentation

5.1.3.1 ModBy

enum [ModBy](#)

Hogyan szeretnénk módosítani az eseményt: bydate: dátum szerint bystart: kezdőidő (óra, perc) szerint byend: befejezőidő (óra, perc) szerint

Enumerator

bydate	
bystart	
byend	

5.1.4 Function Documentation

5.1.4.1 deleteevent()

```
void deleteevent (
    Event * event )
```

Esemény törlése Megkérdezi, hogy töröljük-e

Parameters

<i>event</i>	törölendő Event típusú eseményre mutató pointer
--------------	---

5.1.4.2 moveevent()

```
void moveevent (
    Event * event,
    EventList const * eventlist,
    ModBy modby )
```

Az esemény időhöz kapcsolódó változóinak módosítása Bekéri az új értéket, és erre módosítja Áthelyezi az eseményt a láncolt listában, ha dátum vagy kezdőidő szerint módosítjuk

Parameters

<i>event</i>	Event típusra mutató pointer: a módosítandó esemény
<i>eventlist</i>	EventList -re mutató pointer: a lista amelyikben a módosítandó esemény található

5.1.4.3 printeventrecord()

```
int printeventrecord (
    Event * event,
    SearchConditions condition,
    EventList * eventlist )
```

Kiírja az esemény részleteit és a szerkesztéshez, navigációhoz kapcsolatos menüpontokat majd meghívja a parancsbeolvasó fv-t.

Parameters

<i>event</i>	Event típusra mutató pointer: a módosítandó esemény
<i>condition</i>	a keresés feltételeit tartalmazó változó
<i>eventlist</i>	EventList-re mutató pointer: a lista amelyikben a módosítandó esemény található

Returns

5, ha a főmenübe akarunk menni; 0: kereses mashogy 1: kereses ugyanigy, vagy új esemény -1:talalati lista

5.1.4.4 scanrecordcommand()

```
int scanrecordcommand (
    bool isnewevent,
    int i,
    Event * event,
    EventList const * eventlist )
```

Bekéri az eseményrekord menüből az utasítást és meghívja a kívánt parancshoz tartozó függvényt

Parameters

<i>isnewevent</i>	true, ha újonnan létrehozott esemény rekordját nézzük
<i>i</i>	hány db menüpont van+1
<i>event</i>	Event típusú vizsgált eseményre mutató pointer
<i>eventlist</i>	EventList-re mutató pointer: a lista amelyikben a módosítandó esemény található

Returns

5, ha a főmenübe akarunk menni; 2, ha maradni akarunk az eseménynél 0: kereses mashogy 1: kereses ugyanigy vagy új esemény -1:talalati lista

5.2 File kezelő

adatok fájlba mentése és ebből betöltése mivel az adatok betöltésénél először fix méretű tömbbe tesszük az adatokat, fontos, hogy amikor a felhasználó új adatot ad meg, korlátozzuk annak hosszát, hogy megfeleljen a betöltéskor beolvasható max hosszal.

Functions

- char * [dstcopy](#) (char const *str)
- bool [calendarload](#) ([EventList](#) const *eventlist)
- bool [calendarsave](#) ([EventList](#) const *eventlist)

5.2.1 Detailed Description

adatok fájlba mentése és ebből betöltése mivel az adatok betöltésénél először fix méretű tömbbe tesszük az adatokat, fontos, hogy amikor a felhasználó új adatot ad meg, korlátozzuk annak hosszát, hogy megfeleljen a betöltéskor beolvasható max hosszal.

5.2.2 Function Documentation

5.2.2.1 calendarload()

```
bool calendarload (  
    EventList const * eventlist )
```

Betölti az adatokat a naptar.txt fajlbol

Parameters

<i>eventlist</i>	EventList-re mutató pointer: ebbe a már inicializált listába töltjük be az adatokat
------------------	---

Returns

false, ha nem sikerült végrehajtani, egyébként true

5.2.2.2 calendarsave()

```
bool calendarsave (  
    EventList const * eventlist )
```

Elmenti az adatokat a naptar.txt fajlba

Parameters

<i>eventlist</i>	EventList-re mutató pointer: ebből a listából mentjük ki az adatokat
------------------	--

Returns

false, ha nem sikerült végrehajtani, egyébként true

5.2.2.3 dstncpy()

```
char* dstncpy (  
    char const * str )
```

String másolása dinamikusan foglalt tömbbe

Parameters

<i>karakterstringre</i>	mutató pointer
-------------------------	----------------

Returns

dinamikusan foglalt karaktertömb lezáró nullával

5.3 Lista kezelő

listák, események létrehozás, törlése az eseményeket duplán láncolt listában tároljuk két üres strázsával, időben növekvő sorrendben. a sorrend miatt kell az eseménynek találnunk egy "egyesített" időmértékegységet, amivel össze tudjuk hasonlítani a többi eseményét. Ezt a starttime függvény oldja meg. A beszúráshoz visszafelé keresünk, mivel várhatóan mindig időben későbbi eseményeket szeretnénk bevinni

Typedefs

- typedef struct tm [Tm](#)

Functions

- [EventList](#) * [initeventlist](#) ()
- void [freeEventList](#) ([EventList](#) *eventlist)
- [Event](#) * [createevent](#) (int ev, int honap, int nap, int ora, int perc, int bora, int bperc, char *nev, char *hely, char *comment)
- void [insertevent](#) ([EventList](#) const *eventlist, [Event](#) *event)
- int [starttime](#) ([Event](#) *event)
- void [free_event](#) ([Event](#) *event)

5.3.1 Detailed Description

listák, események létrehozás, törlése az eseményeket duplán láncolt listában tároljuk két üres strázsával, időben növekvő sorrendben. a sorrend miatt kell az eseménynek találnunk egy "egyesített" időmértékegységet, amivel össze tudjuk hasonlítani a többi eseményét. Ezt a starttime függvény oldja meg. A beszúráshoz visszafelé keresünk, mivel várhatóan mindig időben későbbi eseményeket szeretnénk bevinni

5.3.2 Typedef Documentation

5.3.2.1 Tm

```
typedef struct tm Tm
```

időponthoz tartozó struktúra

5.3.3 Function Documentation

5.3.3.1 createevent()

```
Event* createevent (
    int ev,
    int honap,
    int nap,
    int ora,
    int perc,
    int bora,
    int bperc,
    char * nev,
    char * hely,
    char * comment )
```

Létrehoz egy eseményt dianmisan, és feltölti a megadott adatokkal

Parameters

<i>ev</i>	Esemény éve
<i>honap</i>	Esemény hava
<i>nap</i>	Esemény napja
<i>ora</i>	Esemény kezdő órája
<i>perc</i>	Esemény kezdő perce
<i>bora</i>	Esemény befejező órája
<i>bperc</i>	Esemény befejező perce
<i>nev</i>	Esemény nevét tartalmazó char típusú stringre mutató pointer. A megadott string már legyen lefoglalva. Max 128 karakter lezáró nullával együtt.
<i>nev</i>	Esemény helyét tartalmazó char típusú stringre mutató pointer. A megadott string már legyen lefoglalva. Max 128 karakter lezáró nullával együtt.
<i>nev</i>	Eseményre vonatkozó megjegyzést tartalmazó char típusú stringre mutató pointer. A megadott string már legyen lefoglalva. Max 256 karakter lezáró nullával együtt.

Returns

event: [Event](#) típusú létrehozott eseményre mutató pointer

5.3.3.2 free_event()

```
void free_event (
    Event * event )
```

Felszabadítja az eseményt és a tartalmazott karaktertömböket

Parameters

--	--

5.3.3.3 freeEventList()

```
void freeEventList (
    EventList * eventlist )
```

Felszabadítja az eseménylistát eseményekkel együtt A felszabadítandó [EventList](#) típusú listára mutató pointer

5.3.3.4 initeventlist()

```
EventList* initeventlist ( )
```

Létrehoz egy új eseménylistát duplán láncolt listaként, 2 db üres, nem használt órszemmel

Returns

a létrehozott [EventList](#) típusú listára mutató pointer

5.3.3.5 insertevent()

```
void insertevent (
    EventList const * eventlist,
    Event * event )
```

Beilleszt egy már létrehozott eseményt az eseménylistába Az eseményeket kezdő dátum és kezdő idő szerint növekvően rendezve szúrja be A keresést visszafelé (hátról előre) végzi

Parameters

<i>eventlist</i>	Ebbe az EventList típusú listába szúrjuk az eseményt. címmel adjuk meg
<i>event</i>	Ezt az Event típusú eseményt szúrjuk be. Címmel adjuk meg

5.3.3.6 starttime()

```
int starttime (
    Event * event )
```

Létrehozza az események kezdőidejét jelző számot másodperc nagyságrendben a kezdőidő összehasonlításához

Parameters

<i>event</i>	ennek az Event típusú eseménynek keressük a kezdőidejét. Az esemény kezdőmásodpercét nullának vettük
--------------	--

Returns

másodperc nagyságrendben adja vissza, hogy hol helyezkedik el az esemény az időben.

5.4 Menü kezelő

menük kiírása, parancsok, adatok bekérése A menüknél a függvények egyik része kiírja a menüben végrehajtható parancsokat a hozzájuk tartozó, azt meghívó számmal A függvények másik része beolvassa a felhasználó által megadott parancsot, és végrehajtja azt. A menük integereket adnak vissza, ezzel vezérelhetők a visszalépések.

Functions

- void `filesave` (`EventList` const *eventlist)
- int `newevent` (`EventList` *eventlist)
- char * `hosszu_sort_olvas` (int bufferhossz)
- void `mainmenu` ()
- void `scanmainmenu` (`EventList` *eventlist)
- void `searchmenu` (`EventList` *eventlist)
- void `savemenu` (`EventList` const *eventlist)
- int `scansavemenu` (`EventList` const *eventlist)
- void `exitmenu` (`EventList` *eventlist)
- int `scanexitmenu` (`EventList` *eventlist)
- void `printmenu` (`MenuPont` const *menupontok)

5.4.1 Detailed Description

menük kiírása, parancsok, adatok bekérése A menüknél a függvények egyik része kiírja a menüben végrehajtható parancsokat a hozzájuk tartozó, azt meghívó számmal A függvények másik része beolvassa a felhasználó által megadott parancsot, és végrehajtja azt. A menük integereket adnak vissza, ezzel vezérelhetők a visszalépések.

5.4.2 Function Documentation

5.4.2.1 `exitmenu()`

```
void exitmenu (  
    EventList * eventlist )
```

Kiírja a kilépéssel kapcsolatos opciókat és meghívja a parancsbeolvasó függvényt

Parameters

<code>eventlist</code>	Ezt az <code>EventList</code> típusú eseménylistát menthetjük, ha úgy választunk. címmel megadva
------------------------	--

5.4.2.2 `filesave()`

```
void filesave (  
    EventList const * eventlist )
```

Meghívja a mentéssel kapcsolatos menüt, és kiírja, hogy sikeres volt-e a mentés

Parameters

<i>eventlist</i>	Ennek az EventList típusú listának az eseményeit menti. címmel megadva.
------------------	---

5.4.2.3 `hosszu_sort_olvas()`

```
char* hosszu_sort_olvas (
    int bufferhossz )
```

Bekér egy max hosszúságú karaktersort. Szól, ha a megadott sor túl, hosszú, és újra bekéri a szöveget

Parameters

<i>bufferhossz</i>	max ekkora lehet a hossza a megadott stringnek, lezáró 0-val együtt
--------------------	---

Returns

a létrehozott dinamikusan foglalt karaktertömbre mutató pointer

5.4.2.4 `mainmenu()`

```
void mainmenu ( )
```

Kiírja a főmenüt és meghívja a kapcsolódó parancsokat bekérő függvényt

5.4.2.5 `newevent()`

```
int newevent (
    EventList * eventlist )
```

Bekéri a létrehozandó esemény adatait, meghívja a létrehozó függvényeket, meghívja az esemény részleteit kiíró és módosításra szolgáló menüt

Parameters

<i>eventlist</i>	Ebbe az EventList típusú listába szúrjuk be az eseményt. címmel megadva
------------------	---

Returns

továbbítja a `printeventrecord`-ban előállított kimenetet

5.4.2.6 printmenu()

```
void printmenu (
    MenuPont const * menupontok )
```

[MenuPont](#) típusú tömbben megadott menüpontok nevének kiírása

Parameters

<i>menupontok</i>	MenuPont típusú tömb, mely tartalmazza a menüpontok nevét és a kapcsolódó parancs függvénypointerét
-------------------	---

5.4.2.7 savemenu()

```
void savemenu (
    EventList const * eventlist )
```

Megkérdezi, hogy tényleg mentse-e a fájlt. (Le lehet mégészni) Meghívja a választ beolvasó függvényt

Parameters

<i>eventlist</i>	Ennek az EventList típusú listának az eseményeit menti. címmel megadva.
------------------	---

5.4.2.8 scanexitmenu()

```
int scanexitmenu (
    EventList * eventlist )
```

beolvassa az exitmenu-ben felkínált lehetőségekhez kapcsolódó parancsot

Parameters

<i>eventlist</i>	Ezt az EventList típusú eseménylistát menthetjük, ha úgy választunk. címmel megadva
------------------	---

Returns

mindig 0.

5.4.2.9 scanmainmenu()

```
void scanmainmenu (
    EventList * eventlist )
```

Beolvassa a főmenünél megadott parancsot, és végrehajtja az utasítást

5.4.2.10 scansavemenu()

```
int scansavemenu (
    EventList const * eventlist )
```

Beolvassa a savemenu-ben felkínált opciókhoz kapcsolódó parancsot és végrehajta az utasítást

Parameters

<i>eventlist</i>	Ennek az EventList típusú listának az eseményeit menti. címmel megadva.
------------------	---

Returns

mindig 0;

5.4.2.11 searchmenu()

```
void searchmenu (
    EventList * eventlist )
```

Kiírja a keresési módot választó menüt, beolvassa a felhasználó által megadott parancsot, és meghívja a vonatkozó függvényt

5.5 Keresés kezelő

a keresést támogató háttérfüggvények A keresés fő részét a searchiter fv végzi, ami a megadott feltételek alapján a megfelelő eseményeket egy találati listába teszi, és erre meghívja a printfindlist fv-t A keresés időben visszafelé történik, és megáll, ha átléptük a keresett évet vagy elértük a lista elejét.

Typedefs

- typedef enum [SearchBy](#) SearchBy

Enumerations

- enum [SearchBy](#) { byweek, byday, bymonth }

Functions

- int [searchiter](#) (EventList *eventlist, [SearchConditions](#) condition)
- int [searchactual](#) (EventList *eventlist, [SearchBy](#) searchmode)
- void [inserttofindlist](#) (FindList *findlist, [Event](#) *event)
- int [tmtoweek](#) (Tm *tm)
- Tm * [eventtotm](#) (Event *event)
- int [scaninput](#) ()
- int [searchbyname](#) (EventList *eventlist)
- int [searchbyweek](#) (EventList *eventlist)
- int [searchbymonth](#) (EventList *eventlist)
- int [searchbyday](#) (EventList *eventlist)
- int [printfindlist](#) (FindList *findlist, [SearchConditions](#) condition, EventList *eventlist)
- int [scanfindlist](#) (int i, FindList *findlist, [SearchConditions](#) condition, EventList *eventlist)
- void [printevent_short](#) (Event *event)

5.5.1 Detailed Description

a keresést támogató háttérfüggvények A keresés fő részét a searchiter fv végzi, ami a megadott feltételek alapján a megfelelő eseményeket egy találati listába teszi, és erre meghívja a printfindlist fv-t A keresés időben visszafelé történik, és megáll, ha átléptük a keresett évet vagy elértük a lista elejét.

a kereséshez szükséges adatok bekérése, találatok kiírása a keresés típusától függően bekérjük a felhasználótól a szükséges adatokat, ezt típusonként egy-egy fv kezeli.7 Ezek a függvények átadják az infót a külön modulban található searchiter fv-nek, ami meghívja a printfindlist fv-t a találati lista kiírásához a tlálati listában az eseményekhez tartozó szám bevitelével (amit a scanfindlist olvas be) az eseményrekordra jutunk, ahol módosítani és törölni tudjuk az eseményt.

5.5.2 Typedef Documentation

5.5.2.1 SearchBy

```
typedef enum SearchBy SearchBy
```

Keresés módjának megadására szolgáló enum

5.5.3 Enumeration Type Documentation

5.5.3.1 SearchBy

```
enum SearchBy
```

Keresés módjának megadására szolgáló enum

Enumerator

byweek	hét szerint
byday	nap szerint
bymonth	hónap szerint

5.5.4 Function Documentation

5.5.4.1 eventtotm()

```
Tm* eventtotm (  
    Event * event )
```

Esemény kezdőidejét Tm típusú alakítja

Parameters

<i>event</i>	ennek az eseménynek a kezdőidejét alakítjuk át
--------------	--

Returns

Tm típusú pointer (struct tm)

5.5.4.2 inserttofindlist()

```
void inserttofindlist (
    FindList * findlist,
    Event * event )
```

A talált elemek beillesztése egy találati listába

Parameters

<i>findlist</i>	ebbe a találati listába illesztünk be (a searchiter létrehozza). cím szerint adjuk meg.
<i>event</i>	ezt az eseményt illesztjük be. cím szerint.

5.5.4.3 printevent_short()

```
void printevent_short (
    Event * event )
```

Kiírja az esemény nevét, dátumát, kezdőidejét

5.5.4.4 printfindlist()

```
int printfindlist (
    FindList * findlist,
    SearchConditions condition,
    EventList * eventlist )
```

kiírja a keresés által talált események listáját és az elérhető menüpontokat. Meghívja a parancsbeolvasó függvényt a menüpontokhoz

Parameters

<i>findlist</i>	ebben vannak tárolva a keresés által talált elemek
<i>condition</i>	itt tároljuk a keresési feltételeket. lásd: searchiter
<i>eventlist</i>	ebben a listában keresünk

5.5.4.5 scanfindlist()

```
int scanfindlist (
    int i,
    FindList * findlist,
    SearchConditions condition,
    EventList * eventlist )
```

Beolvassa a találati listán érvényes parancsokat a felhasználótól és meghívja a vonatkozó függvényt

Parameters

<i>i</i>	a printfindlist-ben kiírt menüpontok száma+1
<i>findlist</i>	a keresés által visszaadott események listája
<i>condition</i>	keresési feltételek
<i>eventlist</i>	az események listája, amiben keresünk

Returns

1: keresés ugyanígy, 0: keresés máshogy, 5: főmenü

5.5.4.6 scaninput()

```
int scaninput ( )
```

Bekér egy számot, és ezt adja vissza

5.5.4.7 searchactual()

```
int searchactual (
    EventList * eventlist,
    SearchBy searchmode )
```

Keresést futtat a jelenlegi hétre, hónapra vagy napra

Parameters

<i>eventlist</i>	ebben a listában keresünk
<i>searchmode</i>	mi szerint keresünk (byweek, byday, bymonth)

Returns

továbbítja a searchiter-ből kapott visszatérési értéket

5.5.4.8 searchbyday()

```
int searchbyday (
    EventList * eventlist )
```

Nap szerinti kereséshez bekéri az évet, hónapot és napot, majd meghívja a searchiter-t kereséshez

Parameters

<i>eventlist</i>	ebben a listában keresünk
------------------	---------------------------

Returns

a searchiter visszatérési értékét továbbítja

5.5.4.9 searchbymonth()

```
int searchbymonth (
    EventList * eventlist )
```

Hó szerinti kereséshez bekéri az évet és a hónapot, majd meghívja a searchiter-t kereséshez

Parameters

<i>eventlist</i>	ebben a listában keresünk
------------------	---------------------------

Returns

a searchiter visszatérési értékét továbbítja

5.5.4.10 searchbyname()

```
int searchbyname (
    EventList * eventlist )
```

Név szerinti kereséshez bekéri a nevet, majd meghívja a searchiter-t kereséshez

Parameters

<i>eventlist</i>	ebben a listában keresünk
------------------	---------------------------

Returns

a searchiter visszatérési értékét továbbítja

5.5.4.11 searchbyweek()

```
int searchbyweek (
    EventList * eventlist )
```

Hét szerinti kereséshez bekéri az évet és a hetet, majd meghívja a searchiter-t kereséshez

Parameters

<i>eventlist</i>	ebben a listában keresünk
------------------	---------------------------

Returns

a searchiter visszatérési értékét továbbítja

5.5.4.12 searchiter()

```
int searchiter (
    EventList * eventlist,
    SearchConditions condition )
```

A megadott keresési feltételeknek megfelelően végig megy a listán hátulról előre. A keresés addig fut, amíg el nem érjük a lista elejét, vagy ha nem név szerint keresünk addig amíg az iterált esemény éve \geq keresett év. Ha talál a feltételnek megfelelő eseményt, beteszi a találati listába az insertofindlist függvénnyel kiírja a találati listát. Visszatérés előtt felszabadítja a találati listát.

Parameters

<i>eventlist</i>	ebben az EventList típusú eseménylistában keresünk. cím szerint megadva
<i>condition</i>	ebben a SearchConditions típusú változóban adjuk meg az adott kereséshez szükséges adatokat. ha nem név szerint keresünk akkor condition.name=NULL. ha nem hét szerint keresünk, akkor condition.week=0. ha nem nap szerint keresünk, akkor condition.day=0;

Returns

továbbítja a printfindlist visszatérési értékét

5.5.4.13 tmtoweeek()

```
int tmtoweeek (
    Tm * tm )
```

Kiszámítja, hogy adott dátum az év hányadik hetére esik (ISO 8601 szabvány szerint)

Parameters

<i>tm</i>	Tm típusú pointer cím szerint. Itt adjuk meg a vizsgált dátumot
-----------	---

Returns

a hét száma (1-53)

5.6 Szükséges struktúrák

Data Structures

- struct [Event](#)
- struct [EventList](#)
- struct [FoundEvent](#)
- struct [FindList](#)
- struct [SearchConditions](#)
- struct [MenuPont](#)

Typedefs

- typedef struct [Event](#) [Event](#)
- typedef struct [EventList](#) [EventList](#)
- typedef struct [FoundEvent](#) [FoundEvent](#)
- typedef struct [FindList](#) [FindList](#)
- typedef struct [SearchConditions](#) [SearchConditions](#)
- typedef struct [MenuPont](#) [MenuPont](#)
- typedef struct tm [Tm](#)

5.6.1 Detailed Description

5.6.2 Typedef Documentation

5.6.2.1 Event

```
typedef struct Event Event
```

Az eseményt leíró adatok

5.6.2.2 EventList

```
typedef struct EventList EventList
```

Az események listájának eleje és vége

5.6.2.3 FindList

```
typedef struct FindList FindList
```

A keresés által talált események listája

5.6.2.4 FoundEvent

```
typedef struct FoundEvent FoundEvent
```

Keresés által megtalált esemény

5.6.2.5 MenuPont

```
typedef struct MenuPont MenuPont
```

Menupontok neveinek listája

5.6.2.6 SearchConditions

```
typedef struct SearchConditions SearchConditions
```

A keresést leíró paraméterek

5.6.2.7 Tm

```
typedef struct tm Tm
```

struktúra idő tárolására

Chapter 6

Data Structure Documentation

6.1 Event Struct Reference

```
#include <structures.h>
```

Data Fields

- int [year](#)
- int [month](#)
- int [day](#)
- int [starthour](#)
- int [startmin](#)
- int [endhour](#)
- int [endmin](#)
- char * [name](#)
- char * [location](#)
- char * [comment](#)
- struct [Event](#) * [next](#)
- struct [Event](#) * [prev](#)

6.1.1 Detailed Description

Az eseményt leíró adatok

6.1.2 Field Documentation

6.1.2.1 comment

```
char* comment
```

6.1.2.2 day

```
int day
```

6.1.2.3 endhour

```
int endhour
```

6.1.2.4 endmin

```
int endmin
```

6.1.2.5 location

```
char* location
```

6.1.2.6 month

```
int month
```

6.1.2.7 name

```
char* name
```

6.1.2.8 next

```
struct Event* next
```

a duplán láncolt listában az időben következő esemény

6.1.2.9 prev

```
struct Event* prev
```

a duplán láncolt listában az időben megelőző esemény

6.1.2.10 starthour

```
int starthour
```

6.1.2.11 startmin

```
int startmin
```

6.1.2.12 year

```
int year
```

The documentation for this struct was generated from the following file:

- </home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/structures.h>

6.2 EventList Struct Reference

```
#include <structures.h>
```

Data Fields

- [Event](#) * [first](#)
- [Event](#) * [last](#)

6.2.1 Detailed Description

Az események listájának eleje és vége

6.2.2 Field Documentation

6.2.2.1 first

```
Event* first
```

első órszem, üres adatokkal

6.2.2.2 last

`Event* last`

utolsó őrszem, üres adatokkal

The documentation for this struct was generated from the following file:

- </home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/structures.h>

6.3 FindList Struct Reference

```
#include <structures.h>
```

Data Fields

- `FoundEvent* first`
- `FoundEvent* last`

6.3.1 Detailed Description

A keresés által talált események listája

6.3.2 Field Documentation

6.3.2.1 first

`FoundEvent* first`

a találati lista első őrszeme, üres

6.3.2.2 last

`FoundEvent* last`

a találati lista utolsó őrszeme, üres

The documentation for this struct was generated from the following file:

- </home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/structures.h>

6.4 FoundEvent Struct Reference

```
#include <structures.h>
```

Data Fields

- [Event](#) * [foundevent](#)
- struct [FoundEvent](#) * [prevfound](#)
- struct [FoundEvent](#) * [nextfound](#)

6.4.1 Detailed Description

Keresés által megtalált esemény

6.4.2 Field Documentation

6.4.2.1 foundevent

[Event](#)* [foundevent](#)

a talált eseményre mutató pointer

6.4.2.2 nextfound

struct [FoundEvent](#)* [nextfound](#)

a találati listában a találatot követő elem

6.4.2.3 prevfound

struct [FoundEvent](#)* [prevfound](#)

a találati listában a találatot megelőző elem

The documentation for this struct was generated from the following file:

- /home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/[structures.h](#)

6.5 Lefoglalt Struct Reference

Data Fields

- void * [valodi](#)
- void * [usernek](#)
- size_t [meret](#)
- char [fv](#) [16]
- char [cel](#) [128]
- char [file](#) [64]
- unsigned [line](#)
- struct [Lefoglalt](#) * [prev](#)
- struct [Lefoglalt](#) * [next](#)

6.5.1 Field Documentation

6.5.1.1 cel

```
char cel[128]
```

6.5.1.2 file

```
char file[64]
```

6.5.1.3 fv

```
char fv[16]
```

6.5.1.4 line

```
unsigned line
```

6.5.1.5 meret

```
size_t meret
```

6.5.1.6 next

```
struct Lefoglalt * next
```

6.5.1.7 prev

```
struct Lefoglalt* prev
```

6.5.1.8 usernek

```
void* usernek
```

6.5.1.9 valodi

```
void* valodi
```

The documentation for this struct was generated from the following file:

- /home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/[debugmalloc.c](#)

6.6 MenuPont Struct Reference

```
#include <structures.h>
```

Data Fields

- char const * [nev](#)

6.6.1 Detailed Description

Menupontok neveinek listája

6.6.2 Field Documentation

6.6.2.1 nev

```
char const* nev
```

menüpont nevére mutató karaktertömb

The documentation for this struct was generated from the following file:

- </home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/structures.h>

6.7 SearchConditions Struct Reference

```
#include <structures.h>
```

Data Fields

- char * [name](#)
- int [year](#)
- int [week](#)
- int [month](#)
- int [day](#)

6.7.1 Detailed Description

A keresést leíró paraméterek

6.7.2 Field Documentation

6.7.2.1 day

```
int day
```

amelyik napon keresünk vagy 0

6.7.2.2 month

```
int month
```

melyik hónapban keresünk vagy 0

6.7.2.3 name

`char* name`

a keresendő nevet tartalmazó karaktertömb vagy NULL

6.7.2.4 week

`int week`

melyik hétben keresünk vagy 0

6.7.2.5 year

`int year`

melyik évben keresünk

The documentation for this struct was generated from the following file:

- </home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/structures.h>

Chapter 7

File Documentation

7.1 /home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/debugmalloc.c File Reference

```
#include <stdbool.h>
#include <stdlib.h>
#include <stdio.h>
#include <ctype.h>
#include <string.h>
#include <time.h>
#include <stdarg.h>
#include "debugmalloc.h"
```

Data Structures

- struct [Lefoglalt](#)

Macros

- #define [TABLA_OSZLOPOK](#) 256

Typedefs

- typedef struct [Lefoglalt](#) [Lefoglalt](#)

Functions

- size_t [debugmalloc_hash](#) (void *mem)
- void [debugmalloc_naplofajl](#) (char const *nev)
- void [debugmalloc_dump](#) ()
- void * [debugmalloc_malloc_full](#) (size_t meret, char const *fv, char const *cel, char const *file, unsigned line, bool zero)
- void [debugmalloc_free_full](#) (void *mem, char const *fv, char const *file, unsigned line)
- void * [debugmalloc_realloc_full](#) (void *oldmem, size_t newsize, char const *fv, char const *cel, char const *file, unsigned line)
- void * [debugmalloc_malloc](#) (size_t meret)
- void * [debugmalloc_calloc](#) (size_t nmemb, size_t meret)
- void [debugmalloc_free](#) (void *mem)
- void * [debugmalloc_realloc](#) (void *oldmem, size_t meret)

7.1.1 Macro Definition Documentation

7.1.1.1 TABLA_OSZLOPOK

```
#define TABLA_OSZLOPOK 256
```

7.1.2 Typedef Documentation

7.1.2.1 Lefoglalt

```
typedef struct Lefoglalt Lefoglalt
```

7.1.3 Function Documentation

7.1.3.1 debugmalloc_calloc()

```
void* debugmalloc_calloc (
    size_t nmemb,
    size_t meret )
```

7.1.3.2 debugmalloc_dump()

```
void debugmalloc_dump ( )
```

7.1.3.3 debugmalloc_free()

```
void debugmalloc_free (
    void * mem )
```

7.1.3.4 debugmalloc_free_full()

```
void debugmalloc_free_full (
    void * mem,
    char const * fv,
    char const * file,
    unsigned line )
```

7.1.3.5 debugmalloc_hash()

```
size_t debugmalloc_hash (
    void * mem )
```

7.1.3.6 debugmalloc_malloc()

```
void* debugmalloc_malloc (
    size_t meret )
```

7.1.3.7 debugmalloc_malloc_full()

```
void* debugmalloc_malloc_full (
    size_t meret,
    char const * fv,
    char const * cel,
    char const * file,
    unsigned line,
    bool zero )
```

7.1.3.8 debugmalloc_naplofajl()

```
void debugmalloc_naplofajl (
    char const * nev )
```

7.1.3.9 debugmalloc_realloc()

```
void* debugmalloc_realloc (
    void * oldmem,
    size_t meret )
```

7.1.3.10 debugmalloc_realloc_full()

```
void* debugmalloc_realloc_full (
    void * oldmem,
    size_t newsize,
    char const * fv,
    char const * cel,
    char const * file,
    unsigned line )
```

7.2 /home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/debugmalloc.h File Reference

```
#include <stdlib.h>
#include <stdbool.h>
```

Macros

- #define [malloc](#)(X) [debugmalloc_malloc_full](#)(X, "malloc", #X, __FILE__, __LINE__, false)
- #define [calloc](#)(X, Y) [debugmalloc_malloc_full](#)(X*Y, "calloc", #X " ", " #Y, __FILE__, __LINE__, true)
- #define [realloc](#)(P, X) [debugmalloc_realloc_full](#)(P, X, "realloc", #X, __FILE__, __LINE__)
- #define [free](#)(P) [debugmalloc_free_full](#)(P, "free", __FILE__, __LINE__)

Functions

- void [debugmalloc_naplofajl](#) (char const *nev)
- void [debugmalloc_dump](#) ()
- void * [debugmalloc_malloc_full](#) (size_t meret, char const *fv, char const *cel, char const *file, unsigned line, bool zero)
- void * [debugmalloc_realloc_full](#) (void *regimem, size_t newsize, char const *fv, char const *cel, char const *file, unsigned line)
- void [debugmalloc_free_full](#) (void *mem, char const *fv, char const *file, unsigned line)
- void * [debugmalloc_malloc](#) (size_t meret)
- void * [debugmalloc_calloc](#) (size_t nmemb, size_t meret)
- void * [debugmalloc_realloc](#) (void *regimem, size_t meret)
- void [debugmalloc_free](#) (void *mem)

7.2.1 Macro Definition Documentation

7.2.1.1 calloc

```
#define calloc(
    X,
    Y ) debugmalloc\_malloc\_full(X*Y, "calloc", #X " ", " #Y, __FILE__, __LINE__, true)
```

7.2.1.2 free

```
#define free(  
    P ) debugmalloc_free_full(P, "free", __FILE__, __LINE__)
```

7.2.1.3 malloc

```
#define malloc(  
    X ) debugmalloc_malloc_full(X, "malloc", #X, __FILE__, __LINE__, false)
```

7.2.1.4 realloc

```
#define realloc(  
    P,  
    X ) debugmalloc_realloc_full(P, X, "realloc", #X, __FILE__, __LINE__)
```

7.2.2 Function Documentation

7.2.2.1 debugmalloc_calloc()

```
void* debugmalloc_calloc (  
    size_t nmemb,  
    size_t meret )
```

7.2.2.2 debugmalloc_dump()

```
void debugmalloc_dump ( )
```

7.2.2.3 debugmalloc_free()

```
void debugmalloc_free (  
    void * mem )
```

7.2.2.4 debugmalloc_free_full()

```
void debugmalloc_free_full (
    void * mem,
    char const * fv,
    char const * file,
    unsigned line )
```

7.2.2.5 debugmalloc_malloc()

```
void* debugmalloc_malloc (
    size_t meret )
```

7.2.2.6 debugmalloc_malloc_full()

```
void* debugmalloc_malloc_full (
    size_t meret,
    char const * fv,
    char const * cel,
    char const * file,
    unsigned line,
    bool zero )
```

7.2.2.7 debugmalloc_naplofajl()

```
void debugmalloc_naplofajl (
    char const * nev )
```

7.2.2.8 debugmalloc_realloc()

```
void* debugmalloc_realloc (
    void * regimem,
    size_t meret )
```

7.2.2.9 debugmalloc_realloc_full()

```
void* debugmalloc_realloc_full (
    void * regimem,
    size_t newsize,
    char const * fv,
    char const * cel,
    char const * file,
    unsigned line )
```


7.3 /home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/eventrecord.c File Reference

```
#include "eventrecord.h"
#include "structures.h"
#include <stdio.h>
#include <stdbool.h>
#include "menu.h"
#include "list.h"
#include <stdlib.h>
```

Functions

- void [moveevent](#) ([Event](#) *event, [EventList](#) const *eventlist, [ModBy](#) modby)
- void [deleteevent](#) ([Event](#) *event)
- int [scanrecordcommand](#) (bool isnewevent, int i, [Event](#) *event, [EventList](#) const *eventlist)
- int [printeventrecord](#) ([Event](#) *event, [SearchConditions](#) condition, [EventList](#) *eventlist)

7.4 /home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/eventrecord.h File Reference

```
#include "structures.h"
#include <stdbool.h>
```

Typedefs

- typedef enum [ModBy](#) [ModBy](#)

Enumerations

- enum [ModBy](#) { [bydate](#), [bystart](#), [byend](#) }

Functions

- void [moveevent](#) ([Event](#) *event, [EventList](#) const *eventlist, [ModBy](#) modby)
- void [deleteevent](#) ([Event](#) *event)
- int [scanrecordcommand](#) (bool isnewevent, int i, [Event](#) *event, [EventList](#) const *eventlist)
- int [printeventrecord](#) ([Event](#) *event, [SearchConditions](#) condition, [EventList](#) *eventlist)

7.5 /home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/file.c File Reference

```
#include "file.h"
#include <stdio.h>
#include <stdbool.h>
#include "structures.h"
#include "list.h"
#include <string.h>
#include <stdlib.h>
```

Functions

- char * [dstcopy](#) (char const *str)
- bool [calendarload](#) ([EventList](#) const *eventlist)
- bool [calendarsave](#) ([EventList](#) const *eventlist)

7.6 /home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/file.h File Reference

```
#include "structures.h"
#include <stdbool.h>
```

Functions

- char * [dstcopy](#) (char const *str)
- bool [calendarload](#) ([EventList](#) const *eventlist)
- bool [calendarsave](#) ([EventList](#) const *eventlist)

7.7 /home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/list.c File Reference

```
#include "list.h"
#include <stdlib.h>
#include "structures.h"
#include <time.h>
```

Functions

- [EventList](#) * [initeventlist](#) ()
- void [freeEventList](#) ([EventList](#) *eventlist)
- [Event](#) * [createevent](#) (int ev, int honap, int nap, int ora, int perc, int bora, int bperc, char *nev, char *hely, char *comment)
- void [insertevent](#) ([EventList](#) const *eventlist, [Event](#) *event)
- int [starttime](#) ([Event](#) *event)
- void [free_event](#) ([Event](#) *event)

7.8 /home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/list.h File Reference

```
#include "structures.h"
#include <time.h>
```

Typedefs

- typedef struct tm [Tm](#)

Functions

- [EventList](#) * [initeventlist](#) ()
- void [freeEventList](#) ([EventList](#) *eventlist)
- [Event](#) * [createevent](#) (int ev, int honap, int nap, int ora, int perc, int bora, int bperc, char *nev, char *hely, char *comment)
- void [insertevent](#) ([EventList](#) const *eventlist, [Event](#) *event)
- int [starttime](#) ([Event](#) *event)
- void [free_event](#) ([Event](#) *event)

7.9 /home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/main.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include "structures.h"
#include "file.h"
#include "list.h"
#include "menu.h"
#include <stdbool.h>
```

Functions

- int [main](#) ()

7.9.1 Function Documentation

7.9.1.1 main()

```
int main ( )
```

Inicializál egy üres eseménylistát és feltölti adatokkal a fájlból, ha létezik a naptar.txt

7.10 /home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/menu.c

File Reference

```
#include "menu.h"
#include "structures.h"
#include <stdlib.h>
#include "file.h"
#include <stdio.h>
#include <string.h>
#include <stdbool.h>
#include "list.h"
#include "search.h"
#include "eventrecord.h"
#include "searchui.h"
```

Functions

- void [filesave](#) ([EventList](#) const *eventlist)
- int [newevent](#) ([EventList](#) *eventlist)
- char * [hosszu_sort_olvas](#) (int bufferhossz)
- void [mainmenu](#) ()
- void [scanmainmenu](#) ([EventList](#) *eventlist)
- void [searchmenu](#) ([EventList](#) *eventlist)
- void [savemenu](#) ([EventList](#) const *eventlist)
- int [scansavemenu](#) ([EventList](#) const *eventlist)
- void [exitmenu](#) ([EventList](#) *eventlist)
- int [scanexitmenu](#) ([EventList](#) *eventlist)
- void [printmenu](#) ([MenuPont](#) const *menupontok)

7.11 /home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/menu.h

File Reference

```
#include "structures.h"
```

Functions

- void [filesave](#) ([EventList](#) const *eventlist)
- int [newevent](#) ([EventList](#) *eventlist)
- char * [hosszu_sort_olvas](#) (int bufferhossz)
- void [mainmenu](#) ()
- void [scanmainmenu](#) ([EventList](#) *eventlist)
- void [searchmenu](#) ([EventList](#) *eventlist)
- void [savemenu](#) ([EventList](#) const *eventlist)
- int [scansavemenu](#) ([EventList](#) const *eventlist)
- void [exitmenu](#) ([EventList](#) *eventlist)
- int [scanexitmenu](#) ([EventList](#) *eventlist)
- void [printmenu](#) ([MenuPont](#) const *menupontok)

7.12 /home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/search.c File Reference

```
#include "search.h"
#include "structures.h"
#include <stdlib.h>
#include <string.h>
#include <stdbool.h>
#include <time.h>
#include <stdio.h>
#include "searchui.h"
```

Functions

- int [searchiter](#) ([EventList](#) *eventlist, [SearchConditions](#) condition)
- int [searchactual](#) ([EventList](#) *eventlist, [SearchBy](#) searchmode)
- void [inserttofindlist](#) ([FindList](#) *findlist, [Event](#) *event)
- int [tmtoweek](#) ([Tm](#) *tm)
- [Tm](#) * [eventtotm](#) ([Event](#) *event)

7.13 /home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/search.h File Reference

```
#include <stdbool.h>
#include "structures.h"
```

Typedefs

- typedef enum [SearchBy](#) [SearchBy](#)

Enumerations

- enum [SearchBy](#) { [byweek](#), [byday](#), [bymonth](#) }

Functions

- int [searchiter](#) ([EventList](#) *eventlist, [SearchConditions](#) condition)
- int [searchactual](#) ([EventList](#) *eventlist, [SearchBy](#) searchmode)
- void [inserttofindlist](#) ([FindList](#) *findlist, [Event](#) *event)
- int [tmtoweek](#) ([Tm](#) *tm)
- [Tm](#) * [eventtotm](#) ([Event](#) *event)

7.14 /home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/searchui.c

File Reference

```
#include "structures.h"
#include <stdbool.h>
#include <stdio.h>
```

Functions

- int [scaninput](#) ()
- int [searchbyname](#) ([EventList](#) *eventlist)
- int [searchbyweek](#) ([EventList](#) *eventlist)
- int [searchbymonth](#) ([EventList](#) *eventlist)
- int [searchbyday](#) ([EventList](#) *eventlist)
- int [printfindlist](#) ([FindList](#) *findlist, [SearchConditions](#) condition, [EventList](#) *eventlist)
- int [scanfindlist](#) (int i, [FindList](#) *findlist, [SearchConditions](#) condition, [EventList](#) *eventlist)
- void [printevent_short](#) ([Event](#) *event)

7.15 /home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/searchui.h

File Reference

Functions

- int [scaninput](#) ()
- int [searchbyname](#) ([EventList](#) *eventlist)
- int [searchbyweek](#) ([EventList](#) *eventlist)
- int [searchbymonth](#) ([EventList](#) *eventlist)
- int [searchbyday](#) ([EventList](#) *eventlist)
- int [printfindlist](#) ([FindList](#) *findlist, [SearchConditions](#) condition, [EventList](#) *eventlist)
- int [scanfindlist](#) (int i, [FindList](#) *findlist, [SearchConditions](#) condition, [EventList](#) *eventlist)
- void [printevent_short](#) ([Event](#) *event)

7.16 /home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/structures.h

File Reference

```
#include <time.h>
```

Data Structures

- struct [Event](#)
- struct [EventList](#)
- struct [FoundEvent](#)
- struct [FindList](#)
- struct [SearchConditions](#)
- struct [MenuPont](#)

Typedefs

- typedef struct [Event](#) [Event](#)
- typedef struct [EventList](#) [EventList](#)
- typedef struct [FoundEvent](#) [FoundEvent](#)
- typedef struct [FindList](#) [FindList](#)
- typedef struct [SearchConditions](#) [SearchConditions](#)
- typedef struct [MenuPont](#) [MenuPont](#)
- typedef struct tm [Tm](#)

Index

/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/debugmalloc.↔
c, 41
debugmalloc.h, 44

/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/debugmalloc.↔
h, 44
Lefoglalt, 36

/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/eventrecord.↔
c, 47
Event, 31

/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/eventrecord.↔
h, 47
Lista kezelő, 14

/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/file.↔
c, 48
day
Event, 31

/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/file.↔
h, 48
SearchConditions, 38

/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/list.↔
c, 48
debugmalloc_c
debugmalloc_calloc, 42

/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/list.↔
h, 49
debugmalloc_dump, 42
debugmalloc_free, 42

/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/main.↔
c, 49
debugmalloc_free_full, 42
debugmalloc_hash, 43

/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/menu.↔
c, 50
debugmalloc_malloc, 43
debugmalloc_malloc_full, 43

/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/menu.↔
h, 50
debugmalloc_naplofajl, 43
debugmalloc_realloc, 43

/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/search.↔
c, 51
debugmalloc_realloc_full, 43
Lefoglalt, 42

/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/search.↔
h, 51
TABLEA OSZLOPOK, 42
debugmalloc.h

/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/searchui.↔
c, 52
calloc, 44
debugmalloc_calloc, 45

/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/searchui.↔
h, 52
debugmalloc_dump, 45
debugmalloc_free, 45

/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/structures.↔
h, 52
debugmalloc_free_full, 45
debugmalloc_malloc, 46
debugmalloc_malloc_full, 46
debugmalloc_naplofajl, 46
debugmalloc_realloc, 46
debugmalloc_realloc_full, 46
free, 44
malloc, 45
realloc, 45

bydate
Események részletei, 10

byday
Keresés kezelő, 23

byend
Események részletei, 10

bymonth
Keresés kezelő, 23

bystart
Események részletei, 10

byweek
Keresés kezelő, 23

calendarload
File kezelő, 12

calendarsave
File kezelő, 12

- debugmalloc.h, [45](#)
- debugmalloc_hash
 - debugmalloc.c, [43](#)
- debugmalloc_malloc
 - debugmalloc.c, [43](#)
 - debugmalloc.h, [46](#)
- debugmalloc_malloc_full
 - debugmalloc.c, [43](#)
 - debugmalloc.h, [46](#)
- debugmalloc_naplofajl
 - debugmalloc.c, [43](#)
 - debugmalloc.h, [46](#)
- debugmalloc_realloc
 - debugmalloc.c, [43](#)
 - debugmalloc.h, [46](#)
- debugmalloc_realloc_full
 - debugmalloc.c, [43](#)
 - debugmalloc.h, [46](#)
- deleteevent
 - Események részletei, [10](#)
- dstrcpy
 - File kezelő, [13](#)
- endhour
 - Event, [32](#)
- endmin
 - Event, [32](#)
- Események részletei, [9](#)
 - bydate, [10](#)
 - byend, [10](#)
 - bystart, [10](#)
 - deleteevent, [10](#)
 - ModBy, [9](#), [10](#)
 - moveevent, [10](#)
 - printeventrecord, [11](#)
 - scanrecordcommand, [11](#)
- Event, [31](#)
 - comment, [31](#)
 - day, [31](#)
 - endhour, [32](#)
 - endmin, [32](#)
 - location, [32](#)
 - month, [32](#)
 - name, [32](#)
 - next, [32](#)
 - prev, [32](#)
 - starthour, [32](#)
 - startmin, [33](#)
 - Szükséges struktúrák, [28](#)
 - year, [33](#)
- EventList, [33](#)
 - first, [33](#)
 - last, [33](#)
 - Szükséges struktúrák, [28](#)
- eventtotm
 - Keresés kezelő, [23](#)
- exitmenu
 - Menü kezelő, [18](#)
- file
 - Lefoglalt, [36](#)
- File kezelő, [12](#)
 - calendarload, [12](#)
 - calendarsave, [12](#)
 - dstrcpy, [13](#)
- filesave
 - Menü kezelő, [18](#)
- FindList, [34](#)
 - first, [34](#)
 - last, [34](#)
 - Szükséges struktúrák, [28](#)
- first
 - EventList, [33](#)
 - FindList, [34](#)
- FoundEvent, [35](#)
 - foundevent, [35](#)
 - nextfound, [35](#)
 - prevfound, [35](#)
 - Szükséges struktúrák, [28](#)
- foundevent
 - FoundEvent, [35](#)
- free
 - debugmalloc.h, [44](#)
- free_event
 - Lista kezelő, [15](#)
- freeEventList
 - Lista kezelő, [15](#)
- fv
 - Lefoglalt, [36](#)
- hosszu_sort_olvas
 - Menü kezelő, [19](#)
- initeventlist
 - Lista kezelő, [16](#)
- insertevent
 - Lista kezelő, [16](#)
- inserttofindlist
 - Keresés kezelő, [23](#)
- Keresés kezelő, [22](#)
 - byday, [23](#)
 - bymonth, [23](#)
 - byweek, [23](#)
 - eventtotm, [23](#)
 - inserttofindlist, [23](#)
 - printevent_short, [24](#)
 - printfindlist, [24](#)
 - scanfindlist, [24](#)
 - scaninput, [25](#)
 - SearchBy, [22](#), [23](#)
 - searchactual, [25](#)
 - searchbyday, [25](#)
 - searchbymonth, [26](#)
 - searchbyname, [26](#)
 - searchbyweek, [26](#)
 - searchiter, [27](#)
 - tmtoweeek, [27](#)

- last
 - EventList, [33](#)
 - FindList, [34](#)
- Lefoglalt, [36](#)
 - cel, [36](#)
 - debugmalloc.c, [42](#)
 - file, [36](#)
 - fv, [36](#)
 - line, [36](#)
 - meret, [36](#)
 - next, [36](#)
 - prev, [37](#)
 - usernek, [37](#)
 - valodi, [37](#)
- line
 - Lefoglalt, [36](#)
- Lista kezelő, [14](#)
 - createevent, [14](#)
 - free_event, [15](#)
 - freeEventList, [15](#)
 - initedeventlist, [16](#)
 - insertevent, [16](#)
 - starttime, [16](#)
 - Tm, [14](#)
- location
 - Event, [32](#)
- main
 - main.c, [49](#)
- main.c
 - main, [49](#)
- mainmenu
 - Menü kezelő, [19](#)
- malloc
 - debugmalloc.h, [45](#)
- Menü kezelő, [18](#)
 - exitmenu, [18](#)
 - filesave, [18](#)
 - hosszu_sort_olvas, [19](#)
 - mainmenu, [19](#)
 - newevent, [19](#)
 - printmenu, [19](#)
 - savemenu, [20](#)
 - scanexitmenu, [20](#)
 - scanmainmenu, [20](#)
 - scansavemenu, [20](#)
 - searchmenu, [21](#)
- MenuPont, [37](#)
 - nev, [37](#)
 - Szükséges struktúrák, [29](#)
- meret
 - Lefoglalt, [36](#)
- ModBy
 - Események részletei, [9](#), [10](#)
- month
 - Event, [32](#)
 - SearchConditions, [38](#)
- moveevent
 - Események részletei, [10](#)
- name
 - Event, [32](#)
 - SearchConditions, [38](#)
- nev
 - MenuPont, [37](#)
- newevent
 - Menü kezelő, [19](#)
- next
 - Event, [32](#)
 - Lefoglalt, [36](#)
- nextfound
 - FoundEvent, [35](#)
- prev
 - Event, [32](#)
 - Lefoglalt, [37](#)
- prevfound
 - FoundEvent, [35](#)
- printevent_short
 - Keresés kezelő, [24](#)
- printeventrecord
 - Események részletei, [11](#)
- printfindlist
 - Keresés kezelő, [24](#)
- printmenu
 - Menü kezelő, [19](#)
- realloc
 - debugmalloc.h, [45](#)
- savemenu
 - Menü kezelő, [20](#)
- scanexitmenu
 - Menü kezelő, [20](#)
- scanfindlist
 - Keresés kezelő, [24](#)
- scaninput
 - Keresés kezelő, [25](#)
- scanmainmenu
 - Menü kezelő, [20](#)
- scanrecordcommand
 - Események részletei, [11](#)
- scansavemenu
 - Menü kezelő, [20](#)
- SearchBy
 - Keresés kezelő, [22](#), [23](#)
- SearchConditions, [38](#)
 - day, [38](#)
 - month, [38](#)
 - name, [38](#)
 - Szükséges struktúrák, [29](#)
 - week, [39](#)
 - year, [39](#)
- searchactual
 - Keresés kezelő, [25](#)
- searchbyday
 - Keresés kezelő, [25](#)
- searchbymonth
 - Keresés kezelő, [26](#)

- searchbyname
 - Keresés kezelő, [26](#)
- searchbyweek
 - Keresés kezelő, [26](#)
- searchiter
 - Keresés kezelő, [27](#)
- searchmenu
 - Menü kezelő, [21](#)
- starthour
 - Event, [32](#)
- startmin
 - Event, [33](#)
- starttime
 - Lista kezelő, [16](#)
- Szükséges struktúrák, [28](#)
 - Event, [28](#)
 - EventList, [28](#)
 - FindList, [28](#)
 - FoundEvent, [28](#)
 - MenuPont, [29](#)
 - SearchConditions, [29](#)
 - Tm, [29](#)
- TABLA_OSZLOPOK
 - debugmalloc.c, [42](#)
- Tm
 - Lista kezelő, [14](#)
 - Szükséges struktúrák, [29](#)
- tmtowweek
 - Keresés kezelő, [27](#)
- usernek
 - Lefoglalt, [37](#)
- valodi
 - Lefoglalt, [37](#)
- week
 - SearchConditions, [39](#)
- year
 - Event, [33](#)
 - SearchConditions, [39](#)