

calendar

Generated by Doxygen 1.8.13



# Contents

<b>1</b>	<b>Module Index</b>	<b>1</b>
1.1	Modules . . . . .	1
<b>2</b>	<b>Data Structure Index</b>	<b>3</b>
2.1	Data Structures . . . . .	3
<b>3</b>	<b>File Index</b>	<b>5</b>
3.1	File List . . . . .	5
<b>4</b>	<b>Module Documentation</b>	<b>7</b>
4.1	Események részletei . . . . .	7
4.1.1	Detailed Description . . . . .	7
4.1.2	Typedef Documentation . . . . .	7
4.1.2.1	ModBy . . . . .	7
4.1.3	Enumeration Type Documentation . . . . .	8
4.1.3.1	ModBy . . . . .	8
4.1.4	Function Documentation . . . . .	8
4.1.4.1	deleteevent() . . . . .	8
4.1.4.2	moveevent() . . . . .	8
4.1.4.3	printeventrecord() . . . . .	9
4.1.4.4	scanrecordcommand() . . . . .	9
4.2	File kezelő . . . . .	10
4.2.1	Detailed Description . . . . .	10
4.2.2	Function Documentation . . . . .	10
4.2.2.1	calendarload() . . . . .	10

4.2.2.2	<code>calendarsave()</code>	10
4.2.2.3	<code>dstrcpy()</code>	11
4.3	Lista kezelő	12
4.3.1	Detailed Description	12
4.3.2	Typedef Documentation	12
4.3.2.1	<code>Tm</code>	12
4.3.3	Function Documentation	12
4.3.3.1	<code>createevent()</code>	12
4.3.3.2	<code>free_event()</code>	13
4.3.3.3	<code>freeEventList()</code>	13
4.3.3.4	<code>initeventlist()</code>	13
4.3.3.5	<code>insertevent()</code>	14
4.3.3.6	<code>starttime()</code>	14
4.4	Menü kezelő	15
4.4.1	Detailed Description	15
4.4.2	Function Documentation	15
4.4.2.1	<code>exitmenu()</code>	15
4.4.2.2	<code>filesave()</code>	15
4.4.2.3	<code>hosszu_sort_olvas()</code>	16
4.4.2.4	<code>mainmenu()</code>	16
4.4.2.5	<code>newevent()</code>	16
4.4.2.6	<code>printmenu()</code>	16
4.4.2.7	<code>savemenu()</code>	17
4.4.2.8	<code>scanexitmenu()</code>	17
4.4.2.9	<code>scanmainmenu()</code>	17
4.4.2.10	<code>scansavemenu()</code>	17
4.4.2.11	<code>searchmenu()</code>	18
4.5	Keresés kezelő	19
4.5.1	Detailed Description	19
4.5.2	Typedef Documentation	19

4.5.2.1	SearchBy	19
4.5.3	Enumeration Type Documentation	19
4.5.3.1	SearchBy	19
4.5.4	Function Documentation	20
4.5.4.1	eventtotm()	20
4.5.4.2	inserttofindlist()	20
4.5.4.3	printevent_short()	20
4.5.4.4	printfindlist()	21
4.5.4.5	scanfindlist()	21
4.5.4.6	scaninput()	21
4.5.4.7	searchactual()	22
4.5.4.8	searchbyday()	22
4.5.4.9	searchbymonth()	22
4.5.4.10	searchbyname()	23
4.5.4.11	searchbyweek()	23
4.5.4.12	searchiter()	23
4.5.4.13	tmtoweeek()	24
4.6	Szükséges struktúrák	25
4.6.1	Detailed Description	25
4.6.2	Typedef Documentation	25
4.6.2.1	Event	25
4.6.2.2	EventList	25
4.6.2.3	FindList	25
4.6.2.4	FoundEvent	26
4.6.2.5	MenuPont	26
4.6.2.6	SearchConditions	26
4.6.2.7	Tm	26

<b>5</b>	<b>Data Structure Documentation</b>	<b>27</b>
5.1	Event Struct Reference	27
5.1.1	Detailed Description	27
5.1.2	Field Documentation	27
5.1.2.1	comment	27
5.1.2.2	day	28
5.1.2.3	endhour	28
5.1.2.4	endmin	28
5.1.2.5	location	28
5.1.2.6	month	28
5.1.2.7	name	28
5.1.2.8	next	28
5.1.2.9	prev	28
5.1.2.10	starthour	29
5.1.2.11	startmin	29
5.1.2.12	year	29
5.2	EventList Struct Reference	29
5.2.1	Detailed Description	29
5.2.2	Field Documentation	29
5.2.2.1	first	29
5.2.2.2	last	30
5.3	FindList Struct Reference	30
5.3.1	Detailed Description	30
5.3.2	Field Documentation	30
5.3.2.1	first	30
5.3.2.2	last	30
5.4	FoundEvent Struct Reference	31
5.4.1	Detailed Description	31
5.4.2	Field Documentation	31
5.4.2.1	foundevent	31

5.4.2.2	nextfound	31
5.4.2.3	prevfound	31
5.5	Lefoglalt Struct Reference	32
5.5.1	Field Documentation	32
5.5.1.1	cel	32
5.5.1.2	file	32
5.5.1.3	fv	32
5.5.1.4	line	32
5.5.1.5	meret	32
5.5.1.6	next	33
5.5.1.7	prev	33
5.5.1.8	usernek	33
5.5.1.9	valodi	33
5.6	MenuPont Struct Reference	33
5.6.1	Detailed Description	33
5.6.2	Field Documentation	33
5.6.2.1	nev	34
5.7	SearchConditions Struct Reference	34
5.7.1	Detailed Description	34
5.7.2	Field Documentation	34
5.7.2.1	day	34
5.7.2.2	month	34
5.7.2.3	name	35
5.7.2.4	week	35
5.7.2.5	year	35

<b>6 File Documentation</b>	<b>37</b>
6.1 /home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/debugmalloc.c File Reference	37
6.1.1 Macro Definition Documentation	38
6.1.1.1 TABLA_OSZLOPOK	38
6.1.2 Typedef Documentation	38
6.1.2.1 Lefoglalt	38
6.1.3 Function Documentation	38
6.1.3.1 debugmalloc_calloc()	38
6.1.3.2 debugmalloc_dump()	38
6.1.3.3 debugmalloc_free()	38
6.1.3.4 debugmalloc_free_full()	39
6.1.3.5 debugmalloc_hash()	39
6.1.3.6 debugmalloc_malloc()	39
6.1.3.7 debugmalloc_malloc_full()	39
6.1.3.8 debugmalloc_naplofajl()	39
6.1.3.9 debugmalloc_realloc()	39
6.1.3.10 debugmalloc_realloc_full()	40
6.2 /home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/debugmalloc.h File Reference	40
6.2.1 Macro Definition Documentation	40
6.2.1.1 calloc	40
6.2.1.2 free	41
6.2.1.3 malloc	41
6.2.1.4 realloc	41
6.2.2 Function Documentation	41
6.2.2.1 debugmalloc_calloc()	41
6.2.2.2 debugmalloc_dump()	41
6.2.2.3 debugmalloc_free()	41
6.2.2.4 debugmalloc_free_full()	42
6.2.2.5 debugmalloc_malloc()	42



6.2.2.6	<a href="#">debugmalloc_malloc_full()</a>	42
6.2.2.7	<a href="#">debugmalloc_naplofajl()</a>	42
6.2.2.8	<a href="#">debugmalloc_realloc()</a>	42
6.2.2.9	<a href="#">debugmalloc_realloc_full()</a>	42
6.3	<a href="#">/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/eventrecord.c File Reference</a>	43
6.4	<a href="#">/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/eventrecord.h File Reference</a>	43
6.5	<a href="#">/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/file.c File Reference</a>	44
6.6	<a href="#">/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/file.h File Reference</a>	44
6.7	<a href="#">/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/list.c File Reference</a>	44
6.8	<a href="#">/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/list.h File Reference</a>	45
6.9	<a href="#">/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/main.c File Reference</a>	45
6.9.1	Function Documentation	45
6.9.1.1	<a href="#">main()</a>	45
6.10	<a href="#">/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/menu.c File Reference</a>	46
6.11	<a href="#">/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/menu.h File Reference</a>	46
6.12	<a href="#">/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/search.c File Reference</a>	47
6.13	<a href="#">/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/search.h File Reference</a>	47
6.14	<a href="#">/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/searchui.c File Reference</a>	48
6.15	<a href="#">/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/searchui.h File Reference</a>	48
6.16	<a href="#">/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/structures.h File Reference</a>	48



# Chapter 1

## Module Index

### 1.1 Modules

Here is a list of all modules:

Események részletei . . . . .	7
File kezelő . . . . .	10
Lista kezelő . . . . .	12
Menü kezelő . . . . .	15
Keresés kezelő . . . . .	19
Szükséges struktúrák . . . . .	25



## Chapter 2

# Data Structure Index

### 2.1 Data Structures

Here are the data structures with brief descriptions:

Event	27
EventList	29
FindList	30
FoundEvent	31
Lefoglalt	32
MenuPont	33
SearchConditions	34



## Chapter 3

# File Index

### 3.1 File List

Here is a list of all files with brief descriptions:

/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/ <a href="#">debugmalloc.c</a> . . . . .	37
/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/ <a href="#">debugmalloc.h</a> . . . . .	40
/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/ <a href="#">eventrecord.c</a> . . . . .	43
/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/ <a href="#">eventrecord.h</a> . . . . .	43
/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/ <a href="#">file.c</a> . . . . .	44
/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/ <a href="#">file.h</a> . . . . .	44
/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/ <a href="#">list.c</a> . . . . .	44
/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/ <a href="#">list.h</a> . . . . .	45
/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/ <a href="#">main.c</a> . . . . .	45
/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/ <a href="#">menu.c</a> . . . . .	46
/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/ <a href="#">menu.h</a> . . . . .	46
/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/ <a href="#">search.c</a> . . . . .	47
/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/ <a href="#">search.h</a> . . . . .	47
/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/ <a href="#">searchui.c</a> . . . . .	48
/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/ <a href="#">searchui.h</a> . . . . .	48
/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/ <a href="#">structures.h</a> . . . . .	48





## Chapter 4

# Module Documentation

### 4.1 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)

#### 4.1.1 Detailed Description

Események részleteinek kiírásával és módosításával kapcsolatos függvények

#### 4.1.2 Typedef Documentation

##### 4.1.2.1 [ModBy](#)

```
typedef enum ModBy ModBy
```

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

### 4.1.3 Enumeration Type Documentation

#### 4.1.3.1 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	

### 4.1.4 Function Documentation

#### 4.1.4.1 deleteevent()

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

Esemény törlése Megkérdezi, hogy töröljük-e

##### Parameters

<i>event</i>	törölendő <a href="#">Event</a> típusú eseményre mutató pointer
--------------	---

#### 4.1.4.2 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>	<a href="#">Event</a> típusra mutató pointer: a módosítandó esemény
<i>eventlist</i>	<a href="#">EventList</a> -re mutató pointer: a lista amelyikben a módosítandó esemény található

## 4.1.4.3 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>	<a href="#">Event</a> 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

## 4.1.4.4 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>	<a href="#">Event</a> 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

## 4.2 File kezelő

### Functions

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

### 4.2.1 Detailed Description

adatok fájlba mentése és ebből betöltése

### 4.2.2 Function Documentation

#### 4.2.2.1 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

#### 4.2.2.2 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

**4.2.2.3 dstncpy()**

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

String másolása dinamikusan foglalt tömbbe

**Parameters**

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

**Returns**

dinamikusan foglalt karaktertömb lezáró nullával

## 4.3 Lista kezelő

### 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)

#### 4.3.1 Detailed Description

listák, események létrehozás, törlése

#### 4.3.2 Typedef Documentation

##### 4.3.2.1 Tm

```
typedef struct tm Tm
```

időponthoz tartozó struktúra

#### 4.3.3 Function Documentation

##### 4.3.3.1 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

## 4.3.3.2 free\_event()

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

Felszabadítja az eseményt és a tartalmazott karaktertömböket

## Parameters

--	--

## 4.3.3.3 freeEventList()

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

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

## 4.3.3.4 initeventlist()

```
EventList* initeventlist ( )
```

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

## Returns

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

#### 4.3.3.5 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 szűrjük be. A keresést visszafelé (hátról előre) végzi.

##### Parameters

<i>eventlist</i>	Ebbe az <a href="#">EventList</a> típusú listába szűrjük az eseményt. címmel adjuk meg
<i>event</i>	Ezt az <a href="#">Event</a> típusú eseményt szűrjük be. Címmel adjuk meg

#### 4.3.3.6 starttime()

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

Létrehozza az események kezdőidejét jelző számot másodperc nagyságrendben a kezdőidő összehasonlításához.

##### Parameters

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

##### Returns

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



## 4.4 Menü kezelő

### 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`)

#### 4.4.1 Detailed Description

menük kiírása, parancsok, adatok bekérése

#### 4.4.2 Function Documentation

##### 4.4.2.1 `exitmenu()`

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

Kiírja a kilépéssel kapcsolatos opciókat és meghívja a parancsbeolvasó függvényt

##### Parameters

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

##### 4.4.2.2 `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 <code>EventList</code> típusú listának az eseményeit menti. címmel megadva.
------------------	--

#### 4.4.2.3 `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

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

##### Returns

a létrehozott dinamikusan foglalt karaktertömbre mutató pointer

#### 4.4.2.4 `mainmenu()`

```
void mainmenu ( )
```

Kiírja a főmenüt és meghívja a kapcsolódó parancsokat bekérő függvényt

#### 4.4.2.5 `newevent()`

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

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

##### Parameters

<code>eventlist</code>	Ebbe az <a href="#">EventList</a> típusú listába szúrjuk be az eseményt. címmel megadva
------------------------	---

##### Returns

továbbítja a `printeventrecord`-ban előállított kimenetet

#### 4.4.2.6 `printmenu()`

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

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

## Parameters

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

## 4.4.2.7 savemenu()

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

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

## Parameters

<i>eventlist</i>	Ennek az <a href="#">EventList</a> típusú listának az eseményeit menti. címmel megadva.
------------------	---

## 4.4.2.8 scanexitmenu()

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

beolvassa az exitmenu-ben felkínált lehetőségekhez kapcsolódó parancsot

## Parameters

<i>eventlist</i>	Ezt az <a href="#">EventList</a> típusú eseménylistát menthetjük, ha úgy választunk. címmel megadva
------------------	---

## Returns

mindig 0.

## 4.4.2.9 scanmainmenu()

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

Beolvassa a főmenünél megadott parancsot, és végrehajtja az utasítást

## 4.4.2.10 scansavemenu()

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

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

**Parameters**

<i>eventlist</i>	Ennek az <a href="#">EventList</a> típusú listának az eseményeit menti. címmel megadva.
------------------	---

**Returns**

mindig 0;

**4.4.2.11 searchmenu()**

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

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

## 4.5 Keresés kezelő

### 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 [tmtoweeek](#) ([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)

#### 4.5.1 Detailed Description

a keresést támogató háttérfüggvények

a kereséshez szükséges adatok bekérése, találatok kiírása

#### 4.5.2 Typedef Documentation

##### 4.5.2.1 SearchBy

```
typedef enum SearchBy SearchBy
```

Keresés módjának megadására szolgáló enum

#### 4.5.3 Enumeration Type Documentation

##### 4.5.3.1 SearchBy

```
enum SearchBy
```

Keresés módjának megadására szolgáló enum

## Enumerator

byweek	hét szerint
byday	nap szerint
bymonth	hónap szerint

## 4.5.4 Function Documentation

## 4.5.4.1 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)

## 4.5.4.2 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.

## 4.5.4.3 printevent\_short()

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

Kiírja az esemény nevét, dátumát, kezdőidejét

#### 4.5.4.4 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

#### 4.5.4.5 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

#### 4.5.4.6 scaninput()

```
int scaninput ( )
```

Bekér egy számot, és ezt adja vissza

#### 4.5.4.7 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

#### 4.5.4.8 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

#### 4.5.4.9 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

**4.5.4.10 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

**4.5.4.11 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

**4.5.4.12 searchiter()**

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

A megadott keresési feltételeknek megfelelően végig megy a listán hátulról előre. 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 <a href="#">EventList</a> típusú eseménylistában keresünk. cím szerint megadva
<i>condition</i>	ebben a <a href="#">SearchConditions</a> 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

## 4.5.4.13 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)

## 4.6 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](#)

#### 4.6.1 Detailed Description

#### 4.6.2 Typedef Documentation

##### 4.6.2.1 Event

```
typedef struct Event Event
```

Az eseményt leíró adatok

##### 4.6.2.2 EventList

```
typedef struct EventList EventList
```

Az események listájának eleje és vége

##### 4.6.2.3 FindList

```
typedef struct FindList FindList
```

A keresés által talált események listája

#### 4.6.2.4 FoundEvent

```
typedef struct FoundEvent FoundEvent
```

Keresés által megtalált esemény

#### 4.6.2.5 MenuPont

```
typedef struct MenuPont MenuPont
```

Menupontok neveinek listája

#### 4.6.2.6 SearchConditions

```
typedef struct SearchConditions SearchConditions
```

A keresést leíró paraméterek

#### 4.6.2.7 Tm

```
typedef struct tm Tm
```

struktúra idő tárolására

## Chapter 5

# Data Structure Documentation

### 5.1 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](#)

#### 5.1.1 Detailed Description

Az eseményt leíró adatok

#### 5.1.2 Field Documentation

##### 5.1.2.1 comment

```
char* comment
```

**5.1.2.2 day**

```
int day
```

**5.1.2.3 endhour**

```
int endhour
```

**5.1.2.4 endmin**

```
int endmin
```

**5.1.2.5 location**

```
char* location
```

**5.1.2.6 month**

```
int month
```

**5.1.2.7 name**

```
char* name
```

**5.1.2.8 next**

```
struct Event* next
```

a duplán láncolt listában az időben következő esemény

**5.1.2.9 prev**

```
struct Event* prev
```

a duplán láncolt listában az időben megelőző esemény

#### 5.1.2.10 starthour

```
int starthour
```

#### 5.1.2.11 startmin

```
int startmin
```

#### 5.1.2.12 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>

## 5.2 EventList Struct Reference

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

### Data Fields

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

### 5.2.1 Detailed Description

Az események listájának eleje és vége

### 5.2.2 Field Documentation

#### 5.2.2.1 first

```
Event* first
```

első órszem, üres adatokkal

#### 5.2.2.2 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>

### 5.3 FindList Struct Reference

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

#### Data Fields

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

#### 5.3.1 Detailed Description

A keresés által talált események listája

#### 5.3.2 Field Documentation

##### 5.3.2.1 first

`FoundEvent* first`

a találati lista első őrszeme, üres

##### 5.3.2.2 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>



## 5.4 FoundEvent Struct Reference

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

### Data Fields

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

### 5.4.1 Detailed Description

Keresés által megtalált esemény

### 5.4.2 Field Documentation

#### 5.4.2.1 foundevent

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

a talált eseményre mutató pointer

#### 5.4.2.2 nextfound

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

a találati listában a találatot követő elem

#### 5.4.2.3 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](#)

## 5.5 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](#)

### 5.5.1 Field Documentation

#### 5.5.1.1 cel

```
char cel[128]
```

#### 5.5.1.2 file

```
char file[64]
```

#### 5.5.1.3 fv

```
char fv[16]
```

#### 5.5.1.4 line

```
unsigned line
```

#### 5.5.1.5 meret

```
size_t meret
```

#### 5.5.1.6 next

```
struct Lefoglalt * next
```

#### 5.5.1.7 prev

```
struct Lefoglalt* prev
```

#### 5.5.1.8 usernek

```
void* usernek
```

#### 5.5.1.9 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](#)

## 5.6 MenuPont Struct Reference

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

### Data Fields

- char const \* [nev](#)

### 5.6.1 Detailed Description

Menupontok neveinek listája

### 5.6.2 Field Documentation

#### 5.6.2.1 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>

## 5.7 SearchConditions Struct Reference

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

### Data Fields

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

#### 5.7.1 Detailed Description

A keresést leíró paraméterek

#### 5.7.2 Field Documentation

##### 5.7.2.1 day

```
int day
```

amelyik napon keresünk vagy 0

##### 5.7.2.2 month

```
int month
```

melyik hónapban keresünk vagy 0

### 5.7.2.3 name

`char* name`

a keresendő nevet tartalmazó karaktertömb vagy NULL

### 5.7.2.4 week

`int week`

melyik hétben keresünk vagy 0

### 5.7.2.5 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 6

# File Documentation

### 6.1 /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)

## 6.1.1 Macro Definition Documentation

### 6.1.1.1 TABLA\_OSZLOPOK

```
#define TABLA_OSZLOPOK 256
```

## 6.1.2 Typedef Documentation

### 6.1.2.1 Lefoglalt

```
typedef struct Lefoglalt Lefoglalt
```

## 6.1.3 Function Documentation

### 6.1.3.1 debugmalloc\_calloc()

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

### 6.1.3.2 debugmalloc\_dump()

```
void debugmalloc_dump ( )
```

### 6.1.3.3 debugmalloc\_free()

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



#### 6.1.3.4 debugmalloc\_free\_full()

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

#### 6.1.3.5 debugmalloc\_hash()

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

#### 6.1.3.6 debugmalloc\_malloc()

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

#### 6.1.3.7 debugmalloc\_malloc\_full()

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

#### 6.1.3.8 debugmalloc\_naplofajl()

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

#### 6.1.3.9 debugmalloc\_realloc()

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

### 6.1.3.10 debugmalloc\_realloc\_full()

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

## 6.2 /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)

### 6.2.1 Macro Definition Documentation

#### 6.2.1.1 calloc

```
#define calloc(
    X,
    Y ) debugmalloc_malloc_full(X*Y, "calloc", #X " ", " #Y, __FILE__, __LINE__, true)
```

### 6.2.1.2 free

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

### 6.2.1.3 malloc

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

### 6.2.1.4 realloc

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

## 6.2.2 Function Documentation

### 6.2.2.1 debugmalloc\_calloc()

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

### 6.2.2.2 debugmalloc\_dump()

```
void debugmalloc_dump ( )
```

### 6.2.2.3 debugmalloc\_free()

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

#### 6.2.2.4 debugmalloc\_free\_full()

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

#### 6.2.2.5 debugmalloc\_malloc()

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

#### 6.2.2.6 debugmalloc\_malloc\_full()

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

#### 6.2.2.7 debugmalloc\_naplofajl()

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

#### 6.2.2.8 debugmalloc\_realloc()

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

#### 6.2.2.9 debugmalloc\_realloc\_full()

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

## 6.3 /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)

## 6.4 /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)

## 6.5 /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>
```

### Functions

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

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

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

### Functions

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

## 6.7 /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)

## 6.8 /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)

## 6.9 /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](#) ()

### 6.9.1 Function Documentation

#### 6.9.1.1 main()

```
int main ( )
```

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

## 6.10 /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)

## 6.11 /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)



## 6.12 /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 [tmtoweeek](#) ([Tm](#) \*tm)
- [Tm](#) \* [eventtotm](#) ([Event](#) \*event)

## 6.13 /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 [tmtoweeek](#) ([Tm](#) \*tm)
- [Tm](#) \* [eventtotm](#) ([Event](#) \*event)

## 6.14 /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)

## 6.15 /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)

## 6.16 /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, 37  
debugmalloc.h, 40

/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/debugmalloc.↔  
h, 40  
Lefoglalt, 32

/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/eventrecord.↔  
c, 43  
Event, 27

/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/eventrecord.↔  
h, 43  
Lista kezelő, 12

/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/file.↔  
c, 44  
day  
Event, 27

/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/file.↔  
h, 44  
SearchConditions, 34

/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/list.↔  
c, 44  
debugmalloc\_c  
debugmalloc\_calloc, 38

/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/list.↔  
h, 45  
debugmalloc\_dump, 38  
debugmalloc\_free, 38

/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/main.↔  
c, 45  
debugmalloc\_free\_full, 38  
debugmalloc\_hash, 39

/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/menu.↔  
c, 46  
debugmalloc\_malloc, 39  
debugmalloc\_malloc\_full, 39

/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/menu.↔  
h, 46  
debugmalloc\_naplofajl, 39  
debugmalloc\_realloc, 39

/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/search.↔  
c, 47  
debugmalloc\_realloc\_full, 39  
Lefoglalt, 38

/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/search.↔  
h, 47  
TABLA ÖSZLÖKÖK, 38  
debugmalloc.h

/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/searchui.↔  
c, 48  
calloc, 40  
debugmalloc\_calloc, 41

/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/searchui.↔  
h, 48  
debugmalloc\_dump, 41  
debugmalloc\_free, 41

/home/dani/Documents/egyetem/prog1/nagyhazi/hazi2/calendar2/calendar/structures.↔  
h, 48  
debugmalloc\_free\_full, 41  
debugmalloc\_malloc, 42  
debugmalloc\_malloc\_full, 42  
debugmalloc\_naplofajl, 42  
debugmalloc\_realloc, 42  
debugmalloc\_realloc\_full, 42  
free, 40  
malloc, 41  
realloc, 41

bydate  
Események részletei, 8

byday  
Keresés kezelő, 20

byend  
Események részletei, 8

bymonth  
Keresés kezelő, 20

bystart  
Események részletei, 8

byweek  
Keresés kezelő, 20

calendarload  
File kezelő, 10

calendarsave  
File kezelő, 10

- debugmalloc.h, 41
- debugmalloc\_hash
  - debugmalloc.c, 39
- debugmalloc\_malloc
  - debugmalloc.c, 39
  - debugmalloc.h, 42
- debugmalloc\_malloc\_full
  - debugmalloc.c, 39
  - debugmalloc.h, 42
- debugmalloc\_naplofajl
  - debugmalloc.c, 39
  - debugmalloc.h, 42
- debugmalloc\_realloc
  - debugmalloc.c, 39
  - debugmalloc.h, 42
- debugmalloc\_realloc\_full
  - debugmalloc.c, 39
  - debugmalloc.h, 42
- deleteevent
  - Események részletei, 8
- dstrcpy
  - File kezelő, 11
- endhour
  - Event, 28
- endmin
  - Event, 28
- Események részletei, 7
  - bydate, 8
  - byend, 8
  - bystart, 8
  - deleteevent, 8
  - ModBy, 7, 8
  - moveevent, 8
  - printeventrecord, 9
  - scanrecordcommand, 9
- Event, 27
  - comment, 27
  - day, 27
  - endhour, 28
  - endmin, 28
  - location, 28
  - month, 28
  - name, 28
  - next, 28
  - prev, 28
  - starthour, 28
  - startmin, 29
  - Szükséges struktúrák, 25
  - year, 29
- EventList, 29
  - first, 29
  - last, 29
  - Szükséges struktúrák, 25
- eventtotm
  - Keresés kezelő, 20
- exitmenu
  - Menü kezelő, 15
- file
  - Lefoglalt, 32
- File kezelő, 10
  - calendarload, 10
  - calendarsave, 10
  - dstrcpy, 11
- filesave
  - Menü kezelő, 15
- FindList, 30
  - first, 30
  - last, 30
  - Szükséges struktúrák, 25
- first
  - EventList, 29
  - FindList, 30
- FoundEvent, 31
  - foundevent, 31
  - nextfound, 31
  - prevfound, 31
  - Szükséges struktúrák, 25
- foundevent
  - FoundEvent, 31
- free
  - debugmalloc.h, 40
- free\_event
  - Lista kezelő, 13
- freeEventList
  - Lista kezelő, 13
- fv
  - Lefoglalt, 32
- hosszu\_sort\_olvas
  - Menü kezelő, 16
- initeventlist
  - Lista kezelő, 13
- insertevent
  - Lista kezelő, 13
- inserttofindlist
  - Keresés kezelő, 20
- Keresés kezelő, 19
  - byday, 20
  - bymonth, 20
  - byweek, 20
  - eventtotm, 20
  - inserttofindlist, 20
  - printevent\_short, 20
  - printfindlist, 21
  - scanfindlist, 21
  - scaninput, 21
  - SearchBy, 19
  - searchactual, 21
  - searchbyday, 22
  - searchbymonth, 22
  - searchbyname, 23
  - searchbyweek, 23
  - searchiter, 23
  - tmtoweeek, 24

- last
  - EventList, [29](#)
  - FindList, [30](#)
- Lefoglalt, [32](#)
  - cel, [32](#)
  - debugmalloc.c, [38](#)
  - file, [32](#)
  - fv, [32](#)
  - line, [32](#)
  - meret, [32](#)
  - next, [32](#)
  - prev, [33](#)
  - usernek, [33](#)
  - valodi, [33](#)
- line
  - Lefoglalt, [32](#)
- Lista kezelő, [12](#)
  - createevent, [12](#)
  - free\_event, [13](#)
  - freeEventList, [13](#)
  - initedventlist, [13](#)
  - insertevent, [13](#)
  - starttime, [14](#)
  - Tm, [12](#)
- location
  - Event, [28](#)
- main
  - main.c, [45](#)
- main.c
  - main, [45](#)
- mainmenu
  - Menü kezelő, [16](#)
- malloc
  - debugmalloc.h, [41](#)
- Menü kezelő, [15](#)
  - exitmenu, [15](#)
  - filesave, [15](#)
  - hosszu\_sort\_olvas, [16](#)
  - mainmenu, [16](#)
  - newevent, [16](#)
  - printmenu, [16](#)
  - savemenu, [17](#)
  - scanexitmenu, [17](#)
  - scanmainmenu, [17](#)
  - scansavemenu, [17](#)
  - searchmenu, [18](#)
- MenuPont, [33](#)
  - nev, [33](#)
  - Szükséges struktúrák, [26](#)
- meret
  - Lefoglalt, [32](#)
- ModBy
  - Események részletei, [7](#), [8](#)
- month
  - Event, [28](#)
  - SearchConditions, [34](#)
- moveevent
  - Események részletei, [8](#)
- name
  - Event, [28](#)
  - SearchConditions, [34](#)
- nev
  - MenuPont, [33](#)
- newevent
  - Menü kezelő, [16](#)
- next
  - Event, [28](#)
  - Lefoglalt, [32](#)
- nextfound
  - FoundEvent, [31](#)
- prev
  - Event, [28](#)
  - Lefoglalt, [33](#)
- prevfound
  - FoundEvent, [31](#)
- printevent\_short
  - Keresés kezelő, [20](#)
- printeventrecord
  - Események részletei, [9](#)
- printfindlist
  - Keresés kezelő, [21](#)
- printmenu
  - Menü kezelő, [16](#)
- realloc
  - debugmalloc.h, [41](#)
- savemenu
  - Menü kezelő, [17](#)
- scanexitmenu
  - Menü kezelő, [17](#)
- scanfindlist
  - Keresés kezelő, [21](#)
- scaninput
  - Keresés kezelő, [21](#)
- scanmainmenu
  - Menü kezelő, [17](#)
- scanrecordcommand
  - Események részletei, [9](#)
- scansavemenu
  - Menü kezelő, [17](#)
- SearchBy
  - Keresés kezelő, [19](#)
- SearchConditions, [34](#)
  - day, [34](#)
  - month, [34](#)
  - name, [34](#)
  - Szükséges struktúrák, [26](#)
  - week, [35](#)
  - year, [35](#)
- searchactual
  - Keresés kezelő, [21](#)
- searchbyday
  - Keresés kezelő, [22](#)
- searchbymonth
  - Keresés kezelő, [22](#)

- searchbyname
  - Keresés kezelő, [23](#)
- searchbyweek
  - Keresés kezelő, [23](#)
- searchiter
  - Keresés kezelő, [23](#)
- searchmenu
  - Menü kezelő, [18](#)
- starthour
  - Event, [28](#)
- startmin
  - Event, [29](#)
- starttime
  - Lista kezelő, [14](#)
- Szükséges struktúrák, [25](#)
  - Event, [25](#)
  - EventList, [25](#)
  - FindList, [25](#)
  - FoundEvent, [25](#)
  - MenuPont, [26](#)
  - SearchConditions, [26](#)
  - Tm, [26](#)
- TABLA\_OSZLOPOK
  - debugmalloc.c, [38](#)
- Tm
  - Lista kezelő, [12](#)
  - Szükséges struktúrák, [26](#)
- tmtowweek
  - Keresés kezelő, [24](#)
- usernek
  - Lefoglalt, [33](#)
- valodi
  - Lefoglalt, [33](#)
- week
  - SearchConditions, [35](#)
- year
  - Event, [29](#)
  - SearchConditions, [35](#)