Projet C

1.0.0

Generated by Doxygen 1.8.17

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 2

File Index

2.1 File List

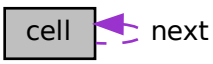
Here is a list of all documented files with brief descriptions:

Chapter 3

Data Structure Documentation

3.1 cell Struct Reference

Collaboration diagram for cell:



Data Fields

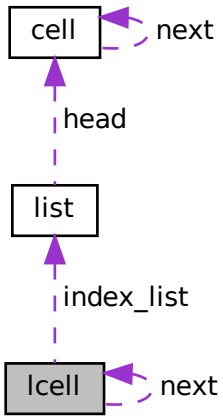
- char * **fname**
- char * **lname**
- char * **zip**
- struct cell * **next**

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

- src/list.h

3.2 lcell Struct Reference

Collaboration diagram for lcell:



Data Fields

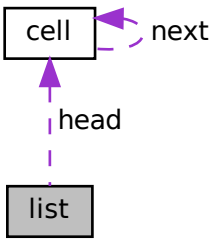
- char * **index**
- struct list * **index_list**
- struct lcell * **next**

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

- src/llist.h

3.3 list Struct Reference

Collaboration diagram for list:



Data Fields

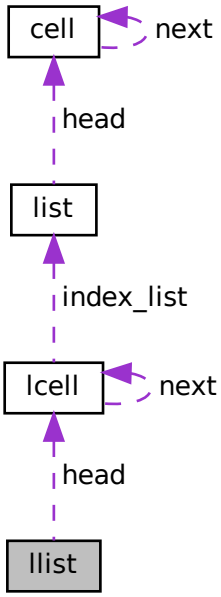
- struct cell * **head**

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

- src/list.h

3.4 llist Struct Reference

Collaboration diagram for llist:



Data Fields

- struct lcell * **head**

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

- src/llist.h

Chapter 4

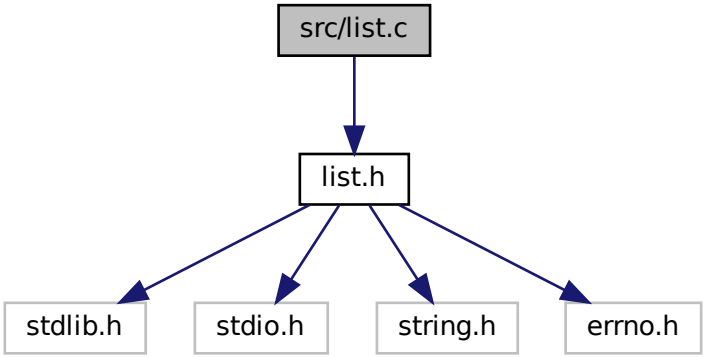
File Documentation

4.1 src/list.c File Reference

This is the list.c file.

```
#include "list.h"
```

Include dependency graph for list.c:



Functions

- struct list ∗ new_list ()
- void free_list (struct list ∗lst)
- void print_cell (struct cell ∗c)
- void print_list (struct list ∗lst)
- struct cell ∗ make_cell (char ∗fname, char ∗lname, char ∗zip)
- void push (struct list ∗lst, struct cell ∗c)
- void pop (struct list ∗lst, struct cell ∗out)
- struct cell ∗ make_cell_from_line (char ∗line)
- struct list ∗ load_file (char ∗file_name)
- int compare_cells (struct cell ∗a, struct cell ∗b)
- void insert (struct list ∗lst, struct cell ∗c)

4.1.1 Detailed Description

This is the list.c file.

Author

Mathis URIEN (LBF38)

Version

1.0.0

Date

2022-10-19

Copyright

Copyright (c) 2022

4.1.2 Function Documentation

4.1.2.1 compare_cells()

```
int compare_cells (
            struct cell * a,
            struct cell * b )
```

Parameters

a	
b	

Returns

int

4.1.2.2 free_list()

```
void free_list (
            struct list * lst )
```

Parameters

lst	

4.1.2.3 insert()

```
void insert (
            struct list * lst,
            struct cell * c )
```

Parameters

lst	
c	

4.1.2.4 load_file()

```
struct list* load_file (
            char * file_name )
```

Parameters

file_name	

Returns

struct list*

4.1.2.5 make_cell()

```
struct cell* make_cell (
            char * fname,
            char * lname,
            char * zip )
```

Parameters

fname	
lname	
zip	

Returns

struct cell*

4.1.2.6 make_cell_from_line()

```
struct cell* make_cell_from_line (
            char * line )
```

Parameters

line	

Returns

struct cell*

4.1.2.7 new_list()

```
struct list* new_list ( )
```

Returns

struct list*

4.1.2.8 pop()

```
void pop (
            struct list * lst,
            struct cell * out )
```

Parameters

lst	
out	

4.1.2.9 print_cell()

```
void print_cell (
            struct cell * c )
```

Parameters

c	

4.1.2.10 print_list()

```
void print_list (
            struct list * lst )
```

Parameters

lst	

4.1.2.11 push()

```
void push (
            struct list * lst,
            struct cell * c )
```

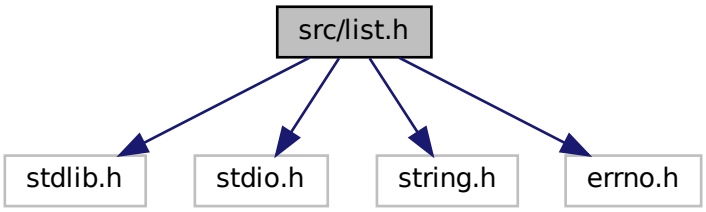
Parameters

lst	
c	

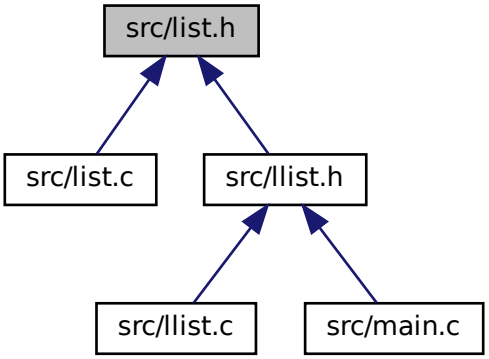
4.2 src/list.h File Reference

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <errno.h>
```

Include dependency graph for list.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct list
- struct cell

Macros

- #define **NAME_LENGTH** 40
- #define **ZIP_LENGTH** 10

Functions

- struct list ∗ new_list ()
- void free_list (struct list ∗lst)
- void print_cell (struct cell ∗c)
- void print_list (struct list ∗lst)
- struct cell ∗ make_cell (char ∗fname, char ∗lname, char ∗zip)
- struct cell ∗ make_cell_from_line (char ∗line)
- void push (struct list ∗lst, struct cell ∗c)
- void pop (struct list ∗lst, struct cell ∗out)
- int compare_cells (struct cell ∗a, struct cell ∗b)
- void insert (struct list ∗lst, struct cell ∗c)
- struct list ∗ load_file (char ∗file_name)

4.2.1 Detailed Description

Author

Mathis URIEN (LBF38)

Version

1.0.0

Date

2022-10-19

Copyright

Copyright (c) 2022

4.2.2 Function Documentation

4.2.2.1 compare_cells()

```
int compare_cells (
            struct cell * a,
            struct cell * b )
```

Parameters

a	
b	

Returns

int

4.2.2.2 free_list()

```
void free_list (
            struct list * lst )
```

Parameters

lst	

4.2.2.3 insert()

```
void insert (
            struct list * lst,
            struct cell * c )
```

Parameters

lst	
c	

4.2.2.4 load_file()

```
struct list* load_file (
            char * file_name )
```

Parameters

file_name	

Returns

struct list*

4.2.2.5 make_cell()

```
struct cell* make_cell (
            char * fname,
```

```
            char * lname,
            char * zip )
```

Parameters

fname	
lname	
zip	

Returns

struct cell*

4.2.2.6 make_cell_from_line()

```
struct cell* make_cell_from_line (
            char * line )
```

Parameters

line	

Returns

struct cell*

4.2.2.7 new_list()

```
struct list* new_list ( )
```

Returns

struct list*

4.2.2.8 pop()

```
void pop (
            struct list * lst,
            struct cell * out )
```

Parameters

lst	
out	

4.2.2.9 print_cell()

```
void print_cell (
            struct cell * c )
```

Parameters

c	

4.2.2.10 print_list()

```
void print_list (
            struct list * lst )
```

Parameters

lst	

4.2.2.11 push()

```
void push (
            struct list * lst,
            struct cell * c )
```

Parameters

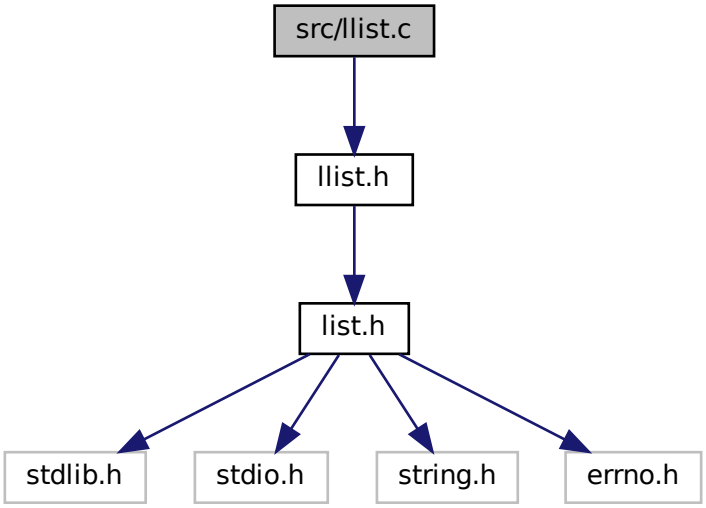
lst	
c	

4.3 src/llist.c File Reference

This is the llist.c file It implements the 2nd method of the project. In order to optimize the loading of data, we use an index to store more efficiently.

```
#include "llist.h"
```

Include dependency graph for llist.c:



Functions

- struct llist * new_llist ()

  new_llist allocates memory for a new llist and returns its pointer.
- void free_llist (struct llist *llst)

  free_llist frees the llist by calling free_lcells and then frees the llist pointer.
- void print_lcell (struct lcell *lcell)

  Prints the lcell content using a list format :
  Example of printing :
  [A,
  {
  [Firstname,Aname,1234],
  }
  ,B]
  Therefore, it shows the content of the current index letter, the index_list and the next index letter.
- void print_llist (struct llist *llst)

  Prints the llist using a format like :
  {{
  "prints the lcells"
  }}.
- struct lcell * make_lcell (struct list *index_list, struct cell *c)

  make_lcell creates a lcell by using the first character of the c->lname and inserts the cellule c into the index_list It also allocates the memory for the lcellule and its core components. (index, index_list)
- int compare_lcells (struct lcell *lcellule, struct cell *c)

  Compare cells to the index to insert it to the right place.
- void insert_optimized (struct llist *llst, struct cell *c)

  Add the cell to the llist respecting alphabetical order of names/lnames and inserting into the right index to optimize the llist.
- struct llist * load_file_optimized (char *file_name)

  Load the contents of a file in a sorted list.

4.3.1 Detailed Description

This is the llist.c file It implements the 2nd method of the project. In order to optimize the loading of data, we use an index to store more efficiently.

Author

Mathis URIEN (LBF38)

Version

1.0.0

Date

2022-10-19

Copyright

Copyright (c) 2022

4.3.2 Function Documentation

4.3.2.1 compare_lcells()

```
int compare_lcells (
            struct lcell * lcellule,
            struct cell * c )
```

Compare cells to the index to insert it to the right place.

Parameters

lcellule	
c	

Returns

int

4.3.2.2 free_llist()

```
void free_llist (
            struct llist * llst )
```

free_llist frees the llist by calling free_lcells and then frees the llist pointer.

Parameters

llst	

4.3.2.3 insert_optimized()

```
void insert_optimized (
            struct llist * llst,
            struct cell * c )
```

Add the cell to the llist respecting alphabetical order of names/lnames and inserting into the right index to optimize the llist.

Parameters

llst	
c	

4.3.2.4 load_file_optimized()

```
struct llist* load_file_optimized (
            char * file_name )
```

Load the contents of a file in a sorted list.

Parameters

file_name	

Returns

struct llist*

4.3.2.5 make_lcell()

```
struct lcell* make_lcell (
            struct list * index_list,
            struct cell * c )
```

make_lcell creates a lcell by using the first character of the c->lname and inserts the cellule c into the index_list It also allocates the memory for the lcellule and its core components. (index, index_list)

Parameters

index_list	
c	

Returns

struct lcell*

4.3.2.6 new_llist()

```
struct llist* new_llist ( )
```

new_llist allocates memory for a new llist and returns its pointer.

Allocate memory for a llist structure and return the pointer.

Returns

struct llist*

4.3.2.7 print_lcell()

```
void print_lcell (
            struct lcell * lcell )
```

Prints the lcell content using a list format :
Example of printing :
[A,
{
[Firstname,Aname,1234],
}
,B]
Therefore, it shows the content of the current index letter, the index_list and the next index letter.

Parameters

lcell	

4.3.2.8 print_llist()

```
void print_llist (
            struct llist * llst )
```

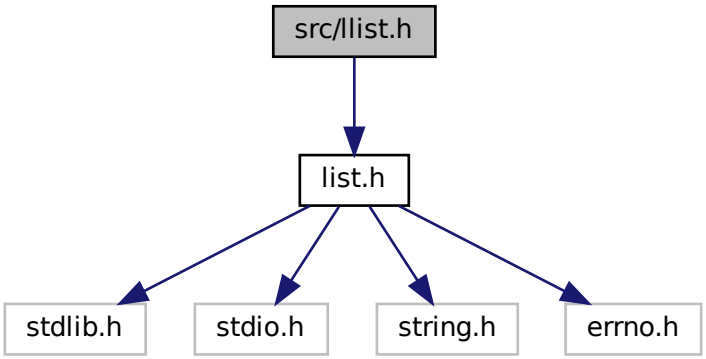
Prints the llist using a format like :
{{
"prints the lcells"
}}.

Parameters

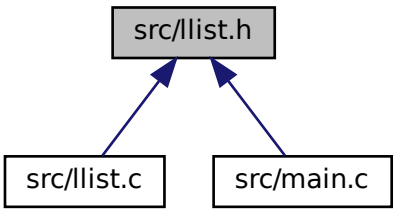
llst	

4.4 src/llist.h File Reference

```
#include "list.h"
```
Include dependency graph for llist.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct llist
- struct lcell

Functions

- struct llist * new_llist ()

  Allocate memory for a llist structure and return the pointer.
- void free_llist (struct llist *llst)

  free_llist frees the llist by calling free_lcells and then frees the llist pointer.
- void print_lcell (struct lcell *lcell)

  Prints the lcell content using a list format :
  Example of printing :
  [A,
  {
  [Firstname,Aname,1234],
  }
  ,B]
  Therefore, it shows the content of the current index letter, the index_list and the next index letter.
- void print_llist (struct llist *llst)

  Prints the llist using a format like :
  {{
  "prints the lcells"
  }}.
- struct lcell * make_lcell (struct list *index_list, struct cell *c)

  make_lcell creates a lcell by using the first character of the c->lname and inserts the cellule c into the index_list It also allocates the memory for the lcellule and its core components. (index, index_list)
- int compare_lcells (struct lcell *lcellule, struct cell *c)

  Compare cells to the index to insert it to the right place.
- void insert_optimized (struct llist *llst, struct cell *c)

  Add the cell to the llist respecting alphabetical order of names/lnames and inserting into the right index to optimize the llist.
- struct llist * load_file_optimized (char *file_name)

  Load the contents of a file in a sorted list.

4.4.1 Detailed Description

Author

Mathis URIEN (LBF38)

Version

1.0.0

Date

2022-10-19

Copyright

4.4.2 Function Documentation

4.4.2.1 compare_lcells()

```
int compare_lcells (
            struct lcell * lcellule,
            struct cell * c )
```

Compare cells to the index to insert it to the right place.

Parameters

lcellule	
c	

Returns

int

4.4.2.2 free_llist()

```
void free_llist (
            struct llist * llst )
```

free_llist frees the llist by calling free_lcells and then frees the llist pointer.

Parameters

llst	

4.4.2.3 insert_optimized()

```
void insert_optimized (
            struct llist * llst,
            struct cell * c )
```

Add the cell to the llist respecting alphabetical order of names/lnames and inserting into the right index to optimize the llist.

Parameters

llst	
c	

4.4.2.4 load_file_optimized()

```
struct llist* load_file_optimized (
            char * file_name )
```

Load the contents of a file in a sorted list.

Parameters

file_name	

Returns

struct llist*

4.4.2.5 make_lcell()

```
struct lcell* make_lcell (
            struct list * index_list,
            struct cell * c )
```

make_lcell creates a lcell by using the first character of the c->lname and inserts the cellule c into the index_list It also allocates the memory for the lcellule and its core components. (index, index_list)

Parameters

index_list	
c	

Returns

struct lcell*

4.4.2.6 new_llist()

```
struct llist* new_llist ( )
```

Allocate memory for a llist structure and return the pointer.

Returns

struct llist*

Allocate memory for a llist structure and return the pointer.

Returns

struct llist*

4.4.2.7 print_lcell()

```
void print_lcell (
            struct lcell * lcell )
```

Prints the lcell content using a list format :
Example of printing :
[A,
{
[Firstname,Aname,1234],
}
,B]
Therefore, it shows the content of the current index letter, the index_list and the next index letter.

Parameters

lcell	

4.4.2.8 print_llist()

```
void print_llist (
            struct llist * llst )
```

Prints the llist using a format like :
{{
"prints the lcells"
}}.

Parameters

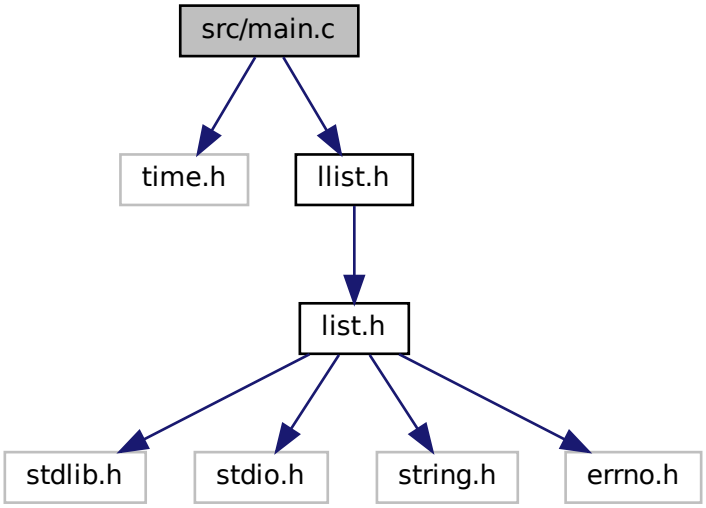
llst	

4.5 src/main.c File Reference

This is the main.c file used to launch the methods for loading files.

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

Include dependency graph for main.c:



Functions

- int main (int argc, char ∗argv[ ])

4.5.1 Detailed Description

This is the main.c file used to launch the methods for loading files.

Author

Mathis URIEN (LBF38)

Version

1.0.0

Date

2022-10-19

Copyright

Copyright (c) 2022

4.5.2 Function Documentation

4.5.2.1 main()

```
int main (
            int argc,
            char * argv[] )
```

Parameters

argc	
argv	

Returns

int

Index