LinkedList

Generated by Doxygen 1.8.11

Contents

1	Data Structure Index							
	1.1	Data Structures	1					
2	File	File Index						
	2.1	File List	2					
3	Data	a Structure Documentation	2					
	3.1	llist_t Struct Reference	2					
		3.1.1 Detailed Description	2					
		3.1.2 Field Documentation	3					
	3.2	Inode_t Struct Reference	3					
		3.2.1 Detailed Description	3					
		3.2.2 Field Documentation	4					
4	File	Documentation	4					
	4.1	llist.c File Reference	4					
		4.1.1 Function Documentation	5					
	4.2	llist.c	7					
	4.3	llist.h File Reference	7					
		4.3.1 Detailed Description	9					
		4.3.2 Typedef Documentation	9					
		4.3.3 Function Documentation	9					
	4.4	llist.h	11					
Index								
1	Da	ta Structure Index						
1.1 Data Structures								
He	re are	e the data structures with brief descriptions:						
llist_t								

Inode_t 3

2 File Index

2.1 File List

Here is a list of all files with brief descriptions:

llist.c

llist.h

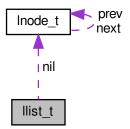
Circular doubly linked list definition and basic operations 7

3 Data Structure Documentation

3.1 Ilist_t Struct Reference

#include <llist.h>

Collaboration diagram for llist_t:



Data Fields

- size_t width
- Inode_t * nil
- int count

3.1.1 Detailed Description

Definition at line 22 of file Ilist.h.

3.1.2 Field Documentation

3.1.2.1 int count

count element amount

Definition at line 25 of file llist.h.

3.1.2.2 **Inode_t*** nil

sentinel (dummy node)

Definition at line 24 of file Ilist.h.

3.1.2.3 size_t width

element size (in bytes)

Definition at line 23 of file Ilist.h.

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

· Ilist.h

3.2 Inode_t Struct Reference

#include <llist.h>

Collaboration diagram for Inode_t:



Data Fields

- struct Inode_t * prev
- void * data
- struct Inode_t * next

3.2.1 Detailed Description

Definition at line 16 of file Ilist.h.

3.2.2 Field Documentation

3.2.2.1 void* data

Definition at line 18 of file Ilist.h.

3.2.2.2 struct Inode t* next

Definition at line 19 of file Ilist.h.

3.2.2.3 struct Inode_t* prev

Definition at line 17 of file Ilist.h.

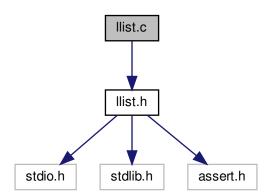
The documentation for this struct was generated from the following file:

• Ilist.h

4 File Documentation

4.1 Ilist.c File Reference

#include "llist.h"
Include dependency graph for llist.c:



Functions

- llist_t * llist_create (size_t width)
- void llist_destruct (llist_t *I)
- Inode_t * Ilist_Isearch (Ilist_t *I, int n)
- void llist_delete (llist_t *I, int n)
- Inode_t * Ilist_insert (Ilist_t *I, int n, void *e)
- void llist_int_print (llist_t *I)

4.1 Ilist.c File Reference 5

4.1.1 Function Documentation

4.1.1.1 Ilist_t* llist_create (size_t width)

Definition at line 7 of file llist.c.

4.1.1.2 void llist_delete ($llist_t * l$, int n)

Definition at line 39 of file Ilist.c.

```
00039
00040     if (1->count == 0) return;
00041     lnode_t* x = llist_lsearch(1, n);
00042     x->prev->next = x->next;
00043     x->next->prev = x->prev;
00044     free (x->data);
00045     free (x);
00046     l->count--;
00047 }
```

Here is the call graph for this function:



4.1.1.3 void llist_destruct (llist_t * /)

Definition at line 19 of file llist.c.

```
00019
00020
         lnode_t* x = 1->nil->next;
         while (x != 1->nil) {
00021
           free(x->data);
00023
              x = x->next;
             free(x->prev);
00024
00025
00026
          free(l->nil);
00027
          free(1);
00028 }
```

```
4.1.1.4 Inode_t* llist_insert ( llist_t * l, int n, void * e )
```

Definition at line 57 of file Ilist.c.

Here is the call graph for this function:



4.1.1.5 void llist_int_print (llist_t * I)

Definition at line 67 of file llist.c.

4.1.1.6 Inode_t* llist_lsearch (llist_t * l, int n)

Definition at line 30 of file llist.c.

4.2 llist.c 7

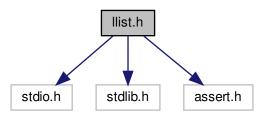
4.2 Ilist.c

```
00001 #include "llist.h"
00002
00003 /*
00004 *
00005
00006
00007 llist_t* llist_create(size_t width) {
80000
        llist_t* 1 = malloc(sizeof(llist_t));
00009
          assert(1);
          1->nil = malloc(sizeof(lnode_t));
00010
          assert(l->nil);
00011
          1->ni1->prev = 1->ni1;
1->ni1->next = 1->ni1;
00012
00013
          l->width = width;
l->count = 0;
00014
00015
00016
           return 1:
00017 }
00018
00019 void llist_destruct(llist_t* 1) {
          lnode_t* x = 1->nil->next;
while(x != 1->nil) {
00020
00021
00022
              free(x->data);
               x = x->next;
00023
00024
               free(x->prev);
00025
00026
           free(l->nil);
00027
           free(1);
00028 }
00029
00030 lnode_t* llist_lsearch(llist_t* 1, int n) {
         assert (n >= 0 || n < 1->count) ;
           lnode_t* x = l->nil->next;
for(int i = 0; i < n; i++) {</pre>
00032
00033
          x = x->next;
00034
00035
00036
           return x;
00037 }
00038
00039 void llist_delete(llist_t* 1, int n) {
          if (1->count == 0) return;
lnode_t* x = llist_lsearch(l, n);
00040
00041
          x->prev->next = x->next;
x->next->prev = x->prev;
00042
00043
00044
           free(x->data);
00045
           free(x);
00046
           1->count--;
00047 }
00048
00049 static void llist_insert_ptr(lnode_t* node, lnode_t* x) {
        lnode_t* pn = node->prev;
x->next = pn->next;
pn->next->prev = x;
00050
00051
00052
00053
           pn->next = x;
00054
           x->prev = pn;
00055 }
00060
           assert (node);
00061
           node->data = e;
00062
           llist_insert_ptr(x, node);
00063
00064
           return node;
00065 }
00066
00067 void llist_int_print(llist_t* 1) {
        printf("%d nodes : nil<->", l->count);
           lnode_t* x = 1->nil->next;
for(int i = 0; i < 1->count; i++) {
    printf("[%d]<->", *((int*)x->data));
00069
00070
00071
00072
               x = x->next;
00073
00074
           puts("nil");
00075 }
```

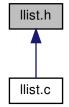
4.3 Ilist.h File Reference

Circular doubly linked list definition and basic operations.

```
#include <stdio.h>
#include <stdlib.h>
#include <assert.h>
Include dependency graph for llist.h:
```



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



Data Structures

- struct Inode_t
- struct llist_t

Typedefs

- typedef struct Inode_t Inode_t
- typedef struct llist_t llist_t

Functions

- llist_t * llist_create (size_t width)
- void llist_destruct (llist_t *I)
- Inode_t * Ilist_insert (Ilist_t *I, int n, void *e)
- void llist_delete (llist_t *I, int n)
- void llist_int_print (llist_t *I)
- Inode_t * Ilist_Isearch (Ilist_t *I, int n)

4.3 Ilist.h File Reference 9

4.3.1 Detailed Description

Circular doubly linked list definition and basic operations.

Author

Firmin MARTIN

Version

0.1

Date

01/01/2018

Definition in file Ilist.h.

- 4.3.2 Typedef Documentation
- 4.3.2.1 typedef struct llist_t llist_t
- 4.3.2.2 typedef struct Inode_t Inode_t
- 4.3.3 Function Documentation
- 4.3.3.1 Ilist_t* llist_create (size_t width)

Definition at line 7 of file llist.c.

4.3.3.2 void llist_delete (llist_t * I, int n)

Definition at line 39 of file llist.c.

```
00039

00040 if (1->count == 0) return;

00041 lnode_t* x = llist_lsearch(1, n);

00042 x->prev->next = x->next;

00043 x->next->prev = x->prev;

00044 free(x->data);

00045 free(x);

00046 l->count--;
```

Here is the call graph for this function:



```
4.3.3.3 void llist_destruct ( llist_t * I )
```

Definition at line 19 of file llist.c.

```
00019
           lnode_t* x = 1->nil->next;
while(x != 1->nil) {
00020
00021
               free(x->data);
00022
00023
               x = x->next;
00024
               free(x->prev);
00025
           free(l->nil);
00026
00027
           free(1);
00028 }
```

4.3.3.4 Inode_t* llist_insert (llist_t * I, int n, void * e)

Definition at line 57 of file Ilist.c.

Here is the call graph for this function:



4.3.3.5 void llist_int_print ($llist_t * I$)

Definition at line 67 of file llist.c.

4.4 Ilist.h

4.3.3.6 Inode_t* llist_lsearch (llist_t * l, int n)

Definition at line 30 of file llist.c.

4.4 Ilist.h

```
00001 #ifndef LLIST_H
00002 #define LLIST_H
00003
00004 #include <stdio.h>
00005 #include <stdlib.h>
00006 #include <assert.h>
00007
00016 typedef struct lnode_t {
          struct lnode_t* prev;
00018
            void* data;
00019
            struct lnode_t* next;
00020 } lnode_t;
00021
00022 typedef struct llist_t {
         size_t width;
lnode_t* nil;
00023
00024
00025
             int count;
00026 } llist_t;
00027
00028 llist_t* llist_create(size_t width);
00029 void llist_destruct(llist_t* 1);
00030 lnode_t* llist_insert(llist_t* 1);
00031 void llist_delete(llist_t* 1, int n, void* e);
00032 void llist_int_print(llist_t* 1);
00033 lnode_t* llist_lsearch(llist_t* 1, int n);
00034
00035 #endif /* ifndef LLIST_H */
```

Index

count llist_t, 3	prev Inode_t, 4
data Inode_t, 4	width llist_t, 3
llist.c, 4	
llist_create, 5	
llist_delete, 5	
llist_destruct, 5	
llist_insert, 5	
llist_int_print, 6	
llist_lsearch, 6 llist.h, 7	
llist_create, 9	
llist_delete, 9	
llist_destruct, 9	
llist_insert, 10	
llist_int_print, 10	
llist_lsearch, 10	
llist_t, 9	
Inode_t, 9	
llist_create	
llist.c, 5	
llist.h, 9	
llist.c, 5	
llist.h, 9	
llist_destruct	
llist.c, 5	
llist.h, 9	
Ilist_insert	
llist.c, 5	
llist.h, 10	
llist_int_print	
llist.c, 6 llist.h, 10	
llist_Isearch	
llist.c, 6	
llist.h, 10	
llist_t, 2	
count, 3	
llist.h, 9	
nil, 3	
width, 3	
Inode_t, 3 data, 4	
llist.h, 9	
next, 4	
prev, 4	
•	
next	
Inode_t, 4	
nil	
llist_t, 3	