


 stefanct / **splib**

<> Code

 Issues 3 Pull requests 1 Actions Projects Security

Join GitHub today


[Dismiss](#)

GitHub is home to over 50 million developers working together to host and review code, manage projects, and build software together.

[Sign up](#)[New issue](#)[Jump to bottom](#)

bugfix #9

 Open**basetutu** opened this issue on 8 Oct 2018 · 2 comments

 **basetutu** commented on 8 Oct 2018 • edited ▼

[sglib/sglib.h](#)

Line 554 in 1d17ed7

```
554      if ((elem)->previous != NULL) (elem)->previous->next = (elem);\
```

Hi!

Would like to suggest a change. (bugfix)

After the indicated line above, there should be the line:

```
(place) = (elem)\
```

This allows consecutive use of `sglib_type_add_before(dllist_first)`, in the corner case where the `dllist_first` is `NULL` to begin with.

Current solution results in keeping the `dllist_first` pointer value the same as first call and place all new entries behind this first one which is continuously is moving further away from the 'front'.

Hence, new entries are placed at length-2 position!

This change fixed it. I can also push the change for review, if you like.

I am only working with the doubly ended linked list.

Best regards



ganwell commented on 8 Oct 2018

Contributor

Hi **@basetutu**

First, this repository is not maintained, either by me or **@stefanct**. Second, I sent my patches to Marian Vittek, and he isn't around anymore. I ended up doing my own [implementation](#) (only rbtree, queue and stack; I don't see the point of linked lists).

If I were the maintainer of `sglib`, I would ask you to extend the tests for this edge cases, to demonstrate how the patch improves things. So I can step through and convince myself, that there are no negative side-effects.

I like `sglib` very much, that's the reason I implemented parts of it from scratch, in order to learn how to implement generic C libraries. However, I won't maintain `sglib`. That is the sad truth.

Peace, out.



ganwell commented on 8 Oct 2018

Contributor

@basetutu I think I did almost the same thing here:

[1d17ed7](#)

Maybe with your change my change should be reverted. Because your fix is general.

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

2 participants



SMG

Make SGLIB_DL_LIST_ADD consistent with SGLIB_LIST_ADD

[Browse files](#)


SGLIB_DL_LIST_ADD should also update the list pointer so it behaves just like SGLIB_LIST_ADD. Only SGLIB_DL_LIST_ADD_AFTER/BEFORE should not update the list pointer.

After:

```
sglib_test_t_add(&list, &a);
sglib_test_t_add(&list, &b);
sglib_test_t_add(&list, &c);
sglib_test_t_add(&list, &d);
```

The list should be d <-> c <-> b <-> a and not
a <-> d <-> c <-> b.

master (#7)

 Jean-Louis Fuchs authored and stefanct committed on 1 May 2017

1 parent 5d559da commit 1d17ed7533fd7f8157f82e1b7ac4b4910f3e0c79

Showing 1 changed file with 1 addition and 0 deletions.

Unified

Split

1 sglib.h

557	557	
558	558	#define SGLIB_DL_LIST_ADD(type, list, elem, previous, next) {\
559	559	SGLIB_DL_LIST_ADD_BEFORE(type, list, elem, previous, next)\
	560	+ (list) = (elem);\
560	561	}
561	562	
562	563	#define SGLIB___DL_LIST_GENERIC_ADD_IF_NOT_MEMBER(type, list, elem, comparator, pr

0 comments on commit 1d17ed7

Please [sign in](#) to comment.