

Assignment 1- FIFO

V3.0

Generated by Doxygen 1.8.17

1 Bug List	1
2 File Index	3
2.1 File List	3
3 File Documentation	5
3.1 MY_FIFO.c File Reference	5
3.2 MY_FIFO.h File Reference	5
3.2.1 Detailed Description	6
3.2.2 Function Documentation	6
3.2.2.1 MyFIFOInit()	6
3.2.2.2 MyFIFOInsert()	6
3.2.2.3 MyFIFOPeep()	8
3.2.2.4 MyFIFORemove()	8
3.2.2.5 MyFIFOSize()	9
3.3 test.c File Reference	9
3.3.1 Detailed Description	9
3.3.2 Function Documentation	10
3.3.2.1 main()	10
3.4 test2.c File Reference	10
3.4.1 Detailed Description	10
3.4.2 Function Documentation	11
3.4.2.1 main()	11
3.5 test3.c File Reference	11
3.5.1 Detailed Description	11
3.5.2 Function Documentation	12
3.5.2.1 main()	12
Index	13

Chapter 1

Bug List

File [MY_FIFO.h](#)

No known bugs.

File [test.c](#)

No known bugs.

File [test2.c](#)

No known bugs.

File [test3.c](#)

No known bugs.

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

MY_FIFO.c	5
MY_FIFO.h		
	FIFO means First In First Out	5
test.c		
	Test.c file brief decription	9
test2.c		
	Test2.c file brief decription	10
test3.c		
	Test3.c file brief decription	11

Chapter 3

File Documentation

3.1 MY_FIFO.c File Reference

```
#include "MY_FIFO.h"
#include <stdio.h>
#include <stdlib.h>
Include dependency graph for MY_FIFO.c:
```

3.2 MY_FIFO.h File Reference

FIFO means First In First Out.

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

Functions

- void [MyFIFOInit](#) (int tamanho)
*Initialize a FIFO with size **tamanho**.*
- void [MyFIFOInsert](#) (int add)
Insert an element in the FIFO.
- int [MyFIFORemove](#) (void)
remover o ultimo elemento inserido. Esta função remove o elemento mais antigo inserido no FIFO e devolve -1 se o FIFO estiver vazio
- int [MyFIFOPeep](#) (void)
Apenas ver o elemento mais antigo do FIFO.
- int [MyFIFOSize](#) (void)
numero total de elemntos Esta função devovolve o numero total de elentos que o FIFO contém no determinado momento e devolve este valor

3.2.1 Detailed Description

FIFO means First In First Out.

Contém as funções necessárias para criar um FIFO assim como adicionar ou remover elementos e saber qual o ultimo elemento.

Author

Frederico Moreira, Ana Sousa, Pedro Rodrigues

Date

22 March 2022

Bug No known bugs.

3.2.2 Function Documentation

3.2.2.1 MyFIFOInit()

```
void MyFIFOInit (
    int tamanho )
```

Initialize a FIFO with size **tamanho**.

A função inicializa um FIFO ("Array") com tamanho do argumento de entrada **tamanho** e não retorna nada Example of usage:

```
res = function1(param1, param2);
printf("res=%d\n",res);
```

Parameters

<i>tamanho</i>	size of the FIFO.
----------------	-------------------

Returns

Não retorna nada.

3.2.2.2 MyFIFOInsert()

```
void MyFIFOInsert (
    int add )
```

Insert an element in the FIFO.

Esta função adiciona um determinado elemento inserido pelo utilizador na posição certa do FIFO. Tem assim como argumento o elemento a adicionar ao fifo e não retorna nada

```
res = function1(param1, param2);  
printf("res=%d\n", res);
```

Parameters

<i>add</i>	elemento a adicionar ao FIFO.
------------	-------------------------------

Returns

Não retorna nada.

3.2.2.3 MyFIFOPeep()

```
int MyFIFOPeep (  
    void )
```

Apenas ver o elemento mais antigo do FIFO.

```
res = function1(param1, param2);  
printf("res=%d\n", res);
```

Parameters

<i>NO_args</i>	sem argumentos
----------------	----------------

Returns

Retorna o valor mais antigo do FIFO

3.2.2.4 MyFIFORemove()

```
int MyFIFORemove (  
    void )
```

remover o ultimo elemento inserido. Esta função remove o elemento mais antigo inserido no FIFO e devolve -1 se o FIFO estiver vazio

```
res = function1(param1, param2);  
printf("res=%d\n", res);
```

Parameters

<i>No_param</i>	No parameters
-----------------	---------------

Returns

retorna -1 se nao existir elementos

3.2.2.5 MyFIFOSize()

```
int MyFIFOSize (
    void )
```

numero total de elemntos Esta função devovlle o numero total de elentos que o FIFO contém no determinado momento e devolve este valor

```
res = function1(param1, param2);
printf("res=%d\n", res);
```

Parameters

<i>no_args</i>	nao tem argumentos
<i>arg2</i>	Description of the second parameter of the function.

Returns

retorna o numero de elemontos total do FIFO

3.3 test.c File Reference

[test.c](#) file brief decription

```
#include <stdio.h>
#include <stdlib.h>
#include "MY_FIFO.h"
Include dependency graph for test.c:
```

Functions

- int [main](#) (void)
Brief decription of [main\(\)](#).

3.3.1 Detailed Description

[test.c](#) file brief decription

Follows the detailed description of [MY_FIFO.c](#). It is separated from the brief one by a blank line. In this case [test.c](#) is the file that contains the [main\(\)](#) function.

Author

Ana Sousa, Frederico Moreira, Pedro Rodrigues

Date

22 March 2022

Bug No known bugs.

3.3.2 Function Documentation

3.3.2.1 main()

```
int main (  
    void )
```

Brief description of [main\(\)](#).

Here it goes the long description of [main\(\)](#) main has no input arguments. It then prints the result and returns.

Returns

[main\(\)](#) always returns 0

3.4 test2.c File Reference

[test2.c](#) file brief description

```
#include <stdio.h>  
#include <stdlib.h>  
#include "MY_FIFO.h"  
Include dependency graph for test2.c:
```

Functions

- int [main](#) (void)
Brief description of [main\(\)](#).

3.4.1 Detailed Description

[test2.c](#) file brief description

Follows the detailed description of [MY_FIFO.c](#). It is separated from the brief one by a blank line. In this case [test.c](#) is the file that contains the [main\(\)](#) function.

Author

Ana Sousa, Frederico Moreira, Pedro Rodrigues

Date

22 March 2022

Bug No known bugs.

3.4.2 Function Documentation

3.4.2.1 main()

```
int main (  
    void )
```

Brief decription of [main\(\)](#).

Here it goes the long description of [main\(\)](#) main has no input arguments. It then prints the result and returns.

Returns

[main\(\)](#) always returns 0

3.5 test3.c File Reference

[test3.c](#) file brief decription

```
#include <stdio.h>  
#include <stdlib.h>  
#include "MY_FIFO.h"  
Include dependency graph for test3.c:
```

Functions

- int [main](#) (void)
Brief decription of [main\(\)](#).

3.5.1 Detailed Description

[test3.c](#) file brief decription

Follows the detailed description of [MY_FIFO.c](#). It is separated from the brief one by a blank line. In this case [test.c](#) is the file that contains the [main\(\)](#) function.

Author

Ana Sousa, Frederico Moreira, Pedro Rodrigues

Date

22 March 2022

Bug No known bugs.

3.5.2 Function Documentation

3.5.2.1 `main()`

```
int main (  
    void )
```

Brief decription of `main()`.

Here it goes the long description of `main()` main has no input arguments. It then prints the result and returns.

Returns

`main()` always returns 0

Index

- main
 - test.c, [10](#)
 - test2.c, [11](#)
 - test3.c, [12](#)
- MY_FIFO.c, [5](#)
- MY_FIFO.h, [5](#)
 - MyFIFOInit, [6](#)
 - MyFIFOInsert, [6](#)
 - MyFIFOPeep, [8](#)
 - MyFIFORemove, [8](#)
 - MyFIFOSize, [8](#)
- MyFIFOInit
 - MY_FIFO.h, [6](#)
- MyFIFOInsert
 - MY_FIFO.h, [6](#)
- MyFIFOPeep
 - MY_FIFO.h, [8](#)
- MyFIFORemove
 - MY_FIFO.h, [8](#)
- MyFIFOSize
 - MY_FIFO.h, [8](#)
- test.c, [9](#)
 - main, [10](#)
- test2.c, [10](#)
 - main, [11](#)
- test3.c, [11](#)
 - main, [12](#)