Assigment 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() 7

3.2.2.3 MyFIFOPeep() 7

3.2.2.4 MyFIFORemove() 8

3.2.2.5 MyFIFOSize() 8

3.3 test.c File Reference 9

3.3.1 Detailed Description 9

3.3.2 Function Documentation 9

3.3.2.1 main() 9

3.4 test2.c File Reference 10

3.4.1 Detailed Description 10

3.4.2 Function Documentation 10

3.4.2.1 main() 10

3.5 test3.c File Reference 11

3.5.1 Detailed Description 11

3.5.2 Function Documentation 11

3.5.2.1 main() 11

3.6 test4.c File Reference 12

3.6.1 Detailed Description 12

3.6.2 Function Documentation 12

3.6.2.1 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.

File test4.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 user interface: In this script we can create a FIFO with variable size and interact with it 9
test2.c
Test2.c file brief decription 10
test3.c
Test3.c file brief decription 11
test4.c
Test4.c Insert a value when the FIFO is Full 12

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)

  remove the last inserted element. This function removes the oldest element inserted in the FIFO and returns -1 if the FIFO is empty
- int MyFIFOPeep (void)

  Only see oldest FIFO element.
- int MyFIFOSize (void)

  Total number of elements This function returns the total numbers that the FIFO contains at the given time and returns this value.

3.2.1 Detailed Description

FIFO means First In First Out.

Contains the functions needed to create a FIFO as well as add or remove elements and it know what the last element.

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**.

The function initializes a FIFO ("Array") with input argument size **tamanho** and it doesn't return anything Example of usage:

```
   if (tamanho > MAX_SIZE)
    {
        size_T = MAX_SIZE;
    }
    else{
        size_T = tamanho;
    }
    head = 0;
    tail = 0;
    for (int i = 0; i < size_T; i++)
    {
        fifo_array[i] = 0;
    }
}
```

Parameters

tamanho	size of the FIFO.

Returns

it doesn't return anything.

3.2.2.2 MyFIFOInsert()

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

Insert an element in the FIFO.

This function adds a certain element inserted by the user at the rigth position of the FIFO.It also has the element to add to the FIFO as an arguement and doesn't return anything

```
   void MyFIFOInsert(int add)
{
    if (flag==0){
        fifo_array[head % size_T] = add;
        head++;
        flag=1;
    }
    else{
        if ((head % size_T) == (tail % size_T) ){
            printf("the oldest element are removed and inserted a new value\n");
            fifo_array[head % size_T] = add;
            head++;
            tail++;
        }
        else{
            fifo_array[head % size_T] = add;
            head++;
        }
    }


}
```

Parameters

add	element to add to FIFO.

Returns

it doesn't return anything.

3.2.2.3 MyFIFOPeep()

```
int MyFIFOPeep (
              void  )
```

Only see oldest FIFO element.

```
int MyFIFOPeep(void)
{
    int num;
    num = fifo_array[tail % size_T];
    //printf("O elemento mais antigo é %d",num);
    return num;
}
```

Parameters

NO_args	without arguments

Returns

Return the oldest FIFO value

3.2.2.4 MyFIFORemove()

```
int MyFIFORemove (
              void   )
```

remove the last inserted element. This function removes the oldest element inserted in the FIFO and returns -1 if the FIFO is empty

```
   int MyFIFORemove(void)
{
    int const1=0;
    if (tail == head)
    {   printf("O FIFO está vazio\n");
        return -1;

    }
    else
    {   const1= fifo_array[tail % size_T];
        fifo_array[tail % size_T] = 0;
        tail++;
        return const1;
    }
}
```

Parameters

No_param	No parameters

Returns

return -1 if there is no element

3.2.2.5 MyFIFOSize()

```
int MyFIFOSize (
              void   )
```

Total number of elements This function returns the total numbers that the FIFO contains at the given time and returns this value.

```
 *int MyFIFOSize(void)
{
    int size;
    size = head - tail;
    //printf("FIFO Size:  %d", size);
    return size;
}
```

Parameters

no_args	without arguments
arg2	Description of the second parameter of the function.

Returns

Returns the total number of FIFO elements.

3.3 test.c File Reference

test.c file user interface: In this script we can create a FIFO with variable size and interact with it

```
#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 user interface: In this script we can create a FIFO with variable size and interact with it

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 decription of main().

Main has no input arguments. The main has an infinite loop that you can create and interact with a FIFO in particular:

-insert elements

-remove elements

-peep the oldest element present in the FIFO

-know the size of the FIFO

Returns

main() always returns 0

3.4 test2.c File Reference

test2.c file brief decription

```
#include <stdio.h>
#include <stdlib.h>
#include "MY_FIFO.h"
```

Include dependency graph for test2.c:

Functions

- int main (void)

  Brief decription of main().

3.4.1 Detailed Description

test2.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.

3.4.2 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

3.6 test4.c File Reference

test4.c Insert a value when the FIFO is Full

```
#include <stdio.h>
#include <stdlib.h>
#include "MY_FIFO.h"
```

Include dependency graph for test4.c:

Functions

- int main (void)

  Brief decription of main().

3.6.1 Detailed Description

test4.c Insert a value when the FIFO is Full

In this script we want to know what happens when the FIFO are already full and we want insert a new element

Author

Ana Sousa, Frederico Moreira, Pedro Rodrigues

Date

22 March 2022

Bug No known bugs.

3.6.2 Function Documentation

3.6.2.1 main()

```
int main (
            void )
```

Brief decription of main().

Let's insert several values until filling the FIFO and then replace it with the last value inserted

Returns

main() always returns 0

Index

main
    test.c, 9
    test2.c, 10
    test3.c, 11
    test4.c, 12
MY_FIFO.c, 5
MY_FIFO.h, 5
    MyFIFOInit, 6
    MyFIFOInsert, 6
    MyFIFOPeep, 7
    MyFIFORemove, 8
    MyFIFOSize, 8
MyFIFOInit
    MY_FIFO.h, 6
MyFIFOInsert
    MY_FIFO.h, 6
MyFIFOPeep
    MY_FIFO.h, 7
MyFIFORemove
    MY_FIFO.h, 8
MyFIFOSize
    MY_FIFO.h, 8

test.c, 9
    main, 9
test2.c, 10
    main, 10
test3.c, 11
    main, 11
test4.c, 12
    main, 12