

Search,fill bit patterns

Generated by Doxygen 1.8.11

Contents

1	File Index	1
1.1	File List	1
2	File Documentation	3
2.1	bits.c File Reference	3
2.1.1	Detailed Description	3
2.1.2	Function Documentation	3
2.1.2.1	check_bit_pattern(unsigned char *start_addr)	3
2.1.2.2	fill_pattern(unsigned char *start_addr, unsigned char *end_addr)	4
2.1.2.3	init_zero(unsigned char *start_addr, unsigned char *end_addr)	4
2.1.2.4	process_pattern(unsigned char pattern, int plength)	4
2.2	bits.h File Reference	5
2.2.1	Detailed Description	5
2.2.2	Function Documentation	5
2.2.2.1	check_bit_pattern(unsigned char *start_addr)	5
2.2.2.2	fill_pattern(unsigned char *start_addr, unsigned char *end_addr)	6
2.2.2.3	init_zero(unsigned char *start_addr, unsigned char *end_addr)	6
2.2.2.4	process_pattern(unsigned char pattern, int plength)	6
	Index	7

Chapter 1

File Index

1.1 File List

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

bits.c	Function definitions for bit pattern check and fill	3
bits.h	Function prototypes for bit pattern check and fill	5

Chapter 2

File Documentation

2.1 bits.c File Reference

Function definitions for bit pattern check and fill.

```
#include "bits.h"
```

Functions

- void [process_pattern](#) (unsigned char pattern, int plength)
Computes and stores the byte sequences that repeat using the given pattern.
- unsigned char * [check_bit_pattern](#) (unsigned char *start_addr)
Checks whether the memory contains continuous repetition of the bit pattern from the given start address.
- void [fill_pattern](#) (unsigned char *start_addr, unsigned char *end_addr)
Fills the memory with continuous repetition of the bit pattern from start address to end address.
- void [init_zero](#) (unsigned char *start_addr, unsigned char *end_addr)
Initializes the memory with 0 from start address to end address.

Variables

- static unsigned char * [shifted_patterns](#)
An array to store all possible byte sequences which repeats. eg: 5 bit-pattern has 5 repeating byte sequences.
- static int [length](#)
Stores the length of the pattern (in bits)

2.1.1 Detailed Description

Function definitions for bit pattern check and fill.

This contains the function definitions for checking whether the memory is continuous repetition of given bit pattern and also to fill memory with the bit pattern.

Author

Freeze Francis

2.1.2 Function Documentation

2.1.2.1 unsigned char* check_bit_pattern (unsigned char * start_addr)

Checks whether the memory contains continuous repetition of the bit pattern from the given start address.

Parameters

<i>start_addr</i>	pointer to the starting address
-------------------	---------------------------------

Returns

pointer to the point where the bit pattern repetition fails

2.1.2.2 void fill_pattern (unsigned char * *start_addr*, unsigned char * *end_addr*)

Fills the memory with continuous repetition of the bit pattern from start address to end address.

Parameters

<i>start_addr</i>	pointer to the starting address
<i>end_addr</i>	pointer to the end address

Returns

void

2.1.2.3 void init_zero (unsigned char * *start_addr*, unsigned char * *end_addr*)

Initializes the memory with 0 from start address to end address.

Parameters

<i>start_addr</i>	pointer to the starting address
<i>end_addr</i>	pointer to the end address

Returns

void

2.1.2.4 void process_pattern (unsigned char *pattern*, int *plength*)

Computes and stores the byte sequences that repeat using the given pattern.

Parameters

<i>pattern</i>	the bit pattern to be processed
<i>plength</i>	length of the pattern in bits

Returns

void

2.2 bits.h File Reference

Function prototypes for bit pattern check and fill.

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

Macros

- `#define BYTE_SIZE 8`

Functions

- unsigned char * [check_bit_pattern](#) (unsigned char *start_addr)
Checks whether the memory contains continuous repetition of the bit pattern from the given start address.
- void [fill_pattern](#) (unsigned char *start_addr, unsigned char *end_addr)
Fills the memory with continuous repetition of the bit pattern from start address to end address.
- void [init_zero](#) (unsigned char *start_addr, unsigned char *end_addr)
Initializes the memory with 0 from start address to end address.
- void [process_pattern](#) (unsigned char pattern, int plength)
Computes and stores the byte sequences that repeat using the given pattern.

2.2.1 Detailed Description

Function prototypes for bit pattern check and fill.

This contains the prototypes for checking whether the memory is continuous repetition of given bitpattern and also to fill memory with the bitpattern.

Author

Freeze Francis

2.2.2 Function Documentation

2.2.2.1 unsigned char* check_bit_pattern (unsigned char * start_addr)

Checks whether the memory contains continuous repetition of the bit pattern from the given start address.

Parameters

<code>start_addr</code>	pointer to the starting address
-------------------------	---------------------------------

Returns

pointer to the point where the bit pattern repetition fails

2.2.2.2 void fill_pattern (unsigned char * *start_addr*, unsigned char * *end_addr*)

Fills the memory with continuos repetition of the bit pattern from start address to end address.

Parameters

<i>start_addr</i>	pointer to the starting address
<i>end_addr</i>	pointer to the end address

Returns

void

2.2.2.3 void init_zero (unsigned char * *start_addr*, unsigned char * *end_addr*)

Initializes the memory with 0 from start address to end address.

Parameters

<i>start_addr</i>	pointer to the starting address
<i>end_addr</i>	pointer to the end address

Returns

void

2.2.2.4 void process_pattern (unsigned char *pattern*, int *plength*)

Computes and stores the byte sequences that repeat using the given pattern.

Parameters

<i>pattern</i>	the bit pattern to be processed
<i>plength</i>	length of the pattern in bits

Returns

void

Index

bits.c, [3](#)
 check_bit_pattern, [3](#)
 fill_pattern, [4](#)
 init_zero, [4](#)
 process_pattern, [4](#)
bits.h, [5](#)
 check_bit_pattern, [5](#)
 fill_pattern, [6](#)
 init_zero, [6](#)
 process_pattern, [6](#)

check_bit_pattern
 bits.c, [3](#)
 bits.h, [5](#)

fill_pattern
 bits.c, [4](#)
 bits.h, [6](#)

init_zero
 bits.c, [4](#)
 bits.h, [6](#)

process_pattern
 bits.c, [4](#)
 bits.h, [6](#)