

My Project

構築: Doxygen 1.8.11

Contents

1	ファイル索引	1
1.1	ファイル一覧	1
2	ファイル詳解	3
2.1	src/main.c ファイル	3
2.1.1	詳解	4
2.1.2	関数詳解	4
2.1.2.1	main()	4
2.1.3	変数詳解	4
2.1.3.1	sharedVariable	4
2.2	src/thread1.c ファイル	4
2.2.1	詳解	5
2.2.2	関数詳解	5
2.2.2.1	thread1_function(void *arg)	5
2.2.3	変数詳解	5
2.2.3.1	sharedVariable	5
2.3	src/thread1.h ファイル	5
2.3.1	詳解	6
2.3.2	関数詳解	6
2.3.2.1	thread1_function(void *arg)	6
2.4	src/thread2.c ファイル	7
2.4.1	詳解	8
2.4.2	関数詳解	8
2.4.2.1	thread2_function(void *arg)	8
2.4.3	変数詳解	8
2.4.3.1	sharedVariable	8
2.5	src/thread2.h ファイル	8
2.5.1	詳解	9
2.5.2	関数詳解	9
2.5.2.1	thread2_function(void *arg)	9
	索引	11

Chapter 1

ファイル索引

1.1 ファイル一覧

詳解が付けられているファイルの一覧です。

src/main.c	メインプログラムファイル。スレッドを作成および管理します。	??
src/thread1.c	スレッド 1 の実装	??
src/thread1.h	スレッド 1 のヘッダーファイル	??
src/thread2.c	スレッド 2 の実装	??
src/thread2.h	スレッド 2 のヘッダーファイル	??

Chapter 2

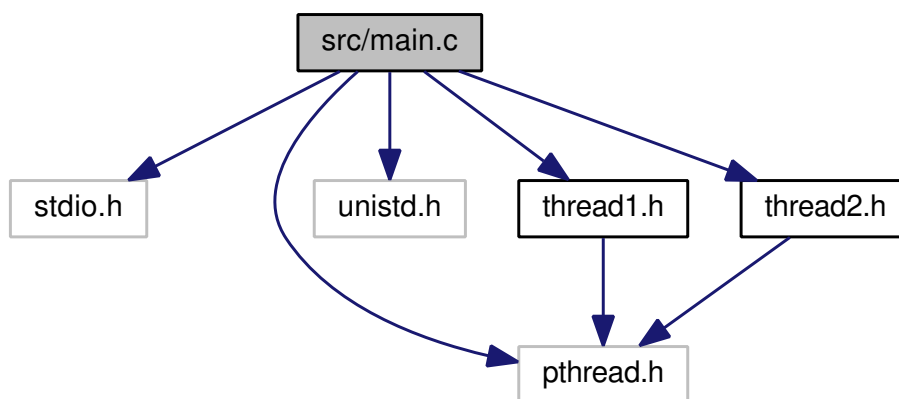
ファイル詳解

2.1 src/main.c ファイル

メインプログラムファイル。スレッドを作成および管理します。

```
#include <stdio.h>
#include <pthread.h>
#include <unistd.h>
#include "thread1.h"
#include "thread2.h"
```

main.c の依存先関係図:



関数

- void displayValue ()
共有変数の値を表示する関数。
- int main ()
スレッドを作成および管理するメイン関数

変数

- int sharedVariable = 0
スレッドによって増分される共有変数。

2.1.1 詳解

メインプログラムファイル。スレッドを作成および管理します。

2.1.2 関数詳解

2.1.2.1 int main()

スレッドを作成および管理するメイン関数

戻り値

成功時に 0 を返す

2.1.3 変数詳解

2.1.3.1 int sharedVariable = 0

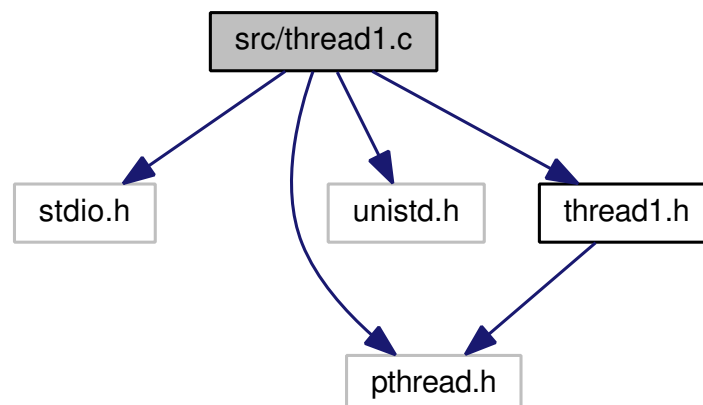
スレッドによって増分される共有変数。

共有変数への参照

2.2 src/thread1.c ファイル

スレッド 1 の実装

```
#include <stdio.h>
#include <pthread.h>
#include <unistd.h>
#include "thread1.h"
thread1.c の依存先関係図:
```



関数

- `void * thread1_function (void *arg)`
スレッド 1 の実装。

変数

- `int sharedVariable`
共有変数への参照

2.2.1 詳解

スレッド 1 の実装

2.2.2 関数詳解

2.2.2.1 `void* thread1_function (void * arg)`

スレッド 1 の実装。

スレッド 1 の関数プロトタイプ

引数

<code>arg</code>	引数へのポインタ
------------------	----------

戻り値

NULL

2.2.3 変数詳解

2.2.3.1 `int sharedVariable`

共有変数への参照

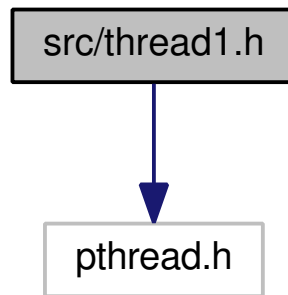
共有変数への参照

2.3 src/thread1.h ファイル

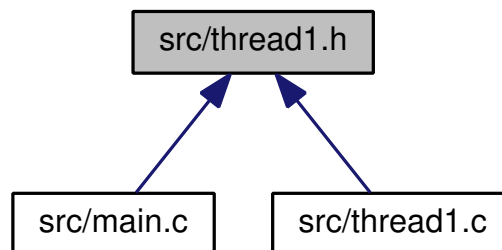
スレッド 1 のヘッダーファイル

```
#include <pthread.h>
```

thread1.h の依存先関係図:



被依存関係図:



関数

- `void * thread1_function (void *arg)`
スレッド 1 の関数プロトタイプ

2.3.1 詳解

スレッド 1 のヘッダーファイル

2.3.2 関数詳解

2.3.2.1 `void* thread1_function (void * arg)`

スレッド 1 の関数プロトタイプ

引数

<code>arg</code>	引数へのポインタ (この例では使用されません)
------------------	-------------------------

戻り値

NULL

スレッド 1 の関数プロトタイプ

引数

<code>arg</code>	引数へのポインタ
------------------	----------

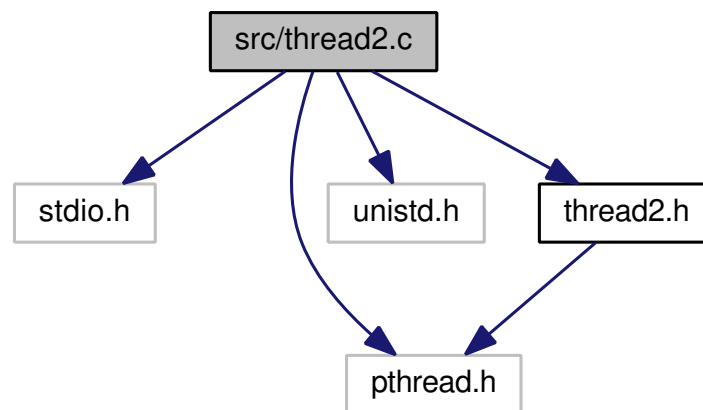
戻り値

NULL

2.4 src/thread2.c ファイル

スレッド 2 の実装

```
#include <stdio.h>
#include <pthread.h>
#include <unistd.h>
#include "thread2.h"
thread2.c の依存先関係図:
```



関数

- `void * thread2_function (void *arg)`
スレッド 2 の実装

変数

- `int sharedVariable`
共有変数への参照

2.4.1 詳解

スレッド 2 の実装

2.4.2 関数詳解

2.4.2.1 void* thread2_function (void * arg)

スレッド 2 の実装

スレッド 2 の関数プロトタイプ

引数

arg	引数へのポインタ
-----	----------

戻り値

NULL

2.4.3 変数詳解

2.4.3.1 int sharedVariable

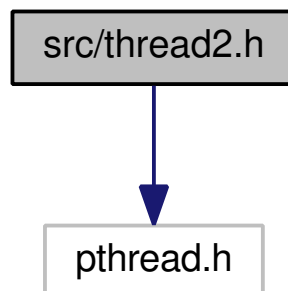
共有変数への参照

共有変数への参照

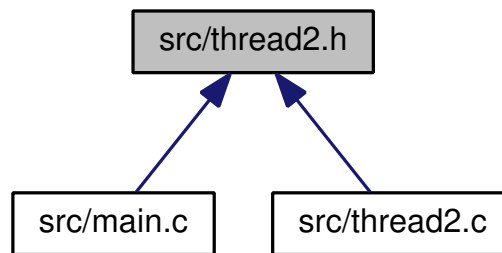
2.5 src/thread2.h ファイル

スレッド 2 のヘッダーファイル

```
#include <pthread.h>  
thread2.h の依存先関係図:
```



被依存関係図:



関数

- `void * thread2_function (void *arg)`
スレッド 2 の関数プロトタイプ

2.5.1 詳解

スレッド 2 のヘッダーファイル

2.5.2 関数詳解

2.5.2.1 `void* thread2_function (void * arg)`

スレッド 2 の関数プロトタイプ

引数

<code>arg</code>	引数へのポインタ
------------------	----------

戻り値

NULL

スレッド 2 の関数プロトタイプ

引数

<code>arg</code>	引数へのポインタ
------------------	----------

戻り値

NULL

Index

- main
 - main.c, 4
- main.c
 - main, 4
 - sharedVariable, 4
- sharedVariable
 - main.c, 4
 - thread1.c, 5
 - thread2.c, 8
- src/main.c, 3
- src/thread1.c, 4
- src/thread1.h, 5
- src/thread2.c, 7
- src/thread2.h, 8
- thread1.c
 - sharedVariable, 5
 - thread1_function, 5
- thread1.h
 - thread1_function, 6
- thread1_function
 - thread1.c, 5
 - thread1.h, 6
- thread2.c
 - sharedVariable, 8
 - thread2_function, 8
- thread2.h
 - thread2_function, 9
- thread2_function
 - thread2.c, 8
 - thread2.h, 9