

## My Project

Generated by Doxygen 1.8.13



# Contents

<b>1</b>	<b>Hierarchical Index</b>	<b>1</b>
1.1	Class Hierarchy . . . . .	1
<b>2</b>	<b>Class Index</b>	<b>3</b>
2.1	Class List . . . . .	3
<b>3</b>	<b>Class Documentation</b>	<b>5</b>
3.1	binary< N > Struct Template Reference . . . . .	5
3.1.1	Member Data Documentation . . . . .	5
3.1.1.1	value . . . . .	5
3.2	binary< 0 > Struct Template Reference . . . . .	5
3.3	Distance Class Reference . . . . .	6
3.4	func Struct Reference . . . . .	6
3.5	is_incrementable< typename, typename > Struct Template Reference . . . . .	6
3.6	is_incrementable< T, std::void_t< decltype(++std::declval< T >())> > Struct Template Reference . . . . .	7
3.7	less< T > Struct Template Reference . . . . .	8
3.8	thread_guard Class Reference . . . . .	8
	<b>Index</b>	<b>9</b>



# Chapter 1

## Hierarchical Index

### 1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

binary< N > . . . . .	5
binary< 0 > . . . . .	5
Distance . . . . .	6
false_type	
is_incrementable< typename, typename > . . . . .	6
func . . . . .	6
less< T > . . . . .	8
thread_guard . . . . .	8
true_type	
is_incrementable< T, std::void_t< decltype(++std::declval< T & >())> > . . . . .	7



## Chapter 2

# Class Index

### 2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">binary&lt; N &gt;</a>	5
<a href="#">binary&lt; 0 &gt;</a>	5
<a href="#">Distance</a>	6
<a href="#">func</a>	6
<a href="#">is_incrementable&lt; typename, typename &gt;</a>	6
<a href="#">is_incrementable&lt; T, std::void_t&lt; decltype(++std::declval&lt; T &gt;())&gt;&gt;</a>	7
<a href="#">less&lt; T &gt;</a>	8
<a href="#">thread_guard</a>	8





## Chapter 3

# Class Documentation

### 3.1 `binary< N >` Struct Template Reference

#### Static Public Attributes

- static unsigned const **value**

#### 3.1.1 Member Data Documentation

##### 3.1.1.1 value

```
template<unsigned long N>
unsigned const binary< N >::value [static]
```

#### Initial value:

```
= binary<N/10>::value << 1
    | N%10
```

The documentation for this struct was generated from the following file:

- `/home/oohnohnoh1/Desktop/GIT/Research/Pthreads_work/Metaprogramming1.cxx`

### 3.2 `binary< 0 >` Struct Template Reference

#### Static Public Attributes

- static unsigned const **value** = 0

The documentation for this struct was generated from the following file:

- `/home/oohnohnoh1/Desktop/GIT/Research/Pthreads_work/Metaprogramming1.cxx`

### 3.3 Distance Class Reference

#### Public Member Functions

- **Distance** (int f, int i)
- **Distance operator()** (int a, int b, int c)
- void **displayDistance** ()

The documentation for this class was generated from the following file:

- /home/oohnohnoh1/Desktop/GIT/Research/Pthreads\_work/ManagingThreads1.cxx

### 3.4 func Struct Reference

#### Public Member Functions

- **func** (int &i\_)
- void **operator()** ()
- **func** (int &i\_)
- void **operator()** ()

#### Public Attributes

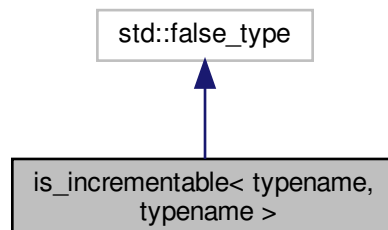
- int & i

The documentation for this struct was generated from the following files:

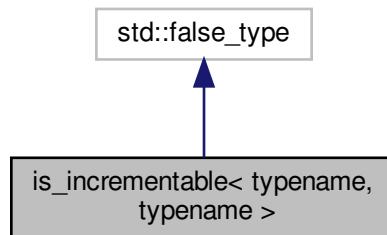
- /home/oohnohnoh1/Desktop/GIT/Research/Pthreads\_work/ManagingThreads1.cxx
- /home/oohnohnoh1/Desktop/GIT/Research/Pthreads\_work/SYN\_thread1.cxx

### 3.5 is\_incrementable< typename, typename > Struct Template Reference

Inheritance diagram for is\_incrementable< typename, typename >:



Collaboration diagram for `is_incrementable< typename, typename >`:

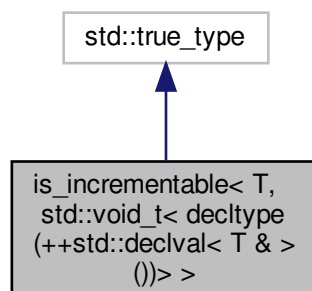


The documentation for this struct was generated from the following file:

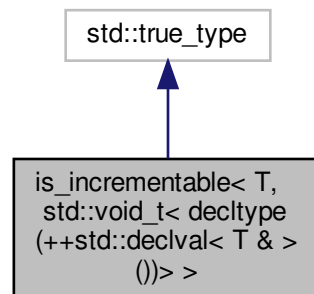
- `/home/oohnohnoh1/Desktop/GIT/Research/Pthreads_work/Metaprogramming1.cxx`

### 3.6 `is_incrementable< T, std::void_t< decltype(++std::declval< T >())>>` > Struct Template Reference

Inheritance diagram for `is_incrementable< T, std::void_t< decltype(++std::declval< T >())>>` >:



Collaboration diagram for `is_incrementable< T, std::void_t< decltype(++std::declval< T >())> >`:



The documentation for this struct was generated from the following file:

- `/home/oohnohnoh1/Desktop/GIT/Research/Pthreads_work/Metaprogramming1.cxx`

### 3.7 `less< T >` Struct Template Reference

#### Public Member Functions

- `bool operator() (T a, T b) const`
- `bool operator() (T a, T b) const`

The documentation for this struct was generated from the following file:

- `/home/oohnohnoh1/Desktop/GIT/Research/Pthreads_work/Metaprogramming2.cxx`

### 3.8 `thread_guard` Class Reference

#### Public Member Functions

- `thread_guard (std::thread &t_)`

The documentation for this class was generated from the following file:

- `/home/oohnohnoh1/Desktop/GIT/Research/Pthreads_work/ManagingThreads1.cxx`

# Index

binary

value, [5](#)

binary< 0 >, [5](#)

binary< N >, [5](#)

Distance, [6](#)

func, [6](#)

is\_incrementable< T, std::void\_t< decltype(++std::declval< T & >())> >, [7](#)

is\_incrementable< typename, typename >, [6](#)

less< T >, [8](#)

thread\_guard, [8](#)

value

binary, [5](#)