



Class SignalHandler

Table of contents

Class Documentation

- Defined in [File signal_handler.hpp](#)

Class Documentation

class SignalHandler

The [SignalHandler](#) class provides a mechanism to handle signals in a C++ program.

The [SignalHandler](#) class provides a way to handle signals in a C++ program. It allows registering global signal handlers and context-specific signal handlers. The class is implemented as a singleton, and its instance can be obtained using the [get_instance\(\)](#) method.

Public Types

```
using GlobalSignalHandlerMap = std::unordered_map<int, std::function<void(int)>>
```

Type definition for a global signal handler map.

```
using SignalHandlerMap = std::unordered_map<int, std::function<void(void*, int)>>
```

Type definition for a signal handler map.

```
using ContextSignalHandlerMap = std::unordered_map<void*, SignalHandlerMap>
```

Type definition for a context-specific signal handler map.

Public Static Functions

```
static SignalHandler &get_instance()
```

Returns the singleton instance of the [SignalHandler](#) class.

Returns

[SignalHandler](#)& The singleton instance of the [SignalHandler](#) class.

```
static void static_handle_signal(int signal)
```

The static method to handles the specified signal.

Parameters

signal – The signal to handle.

```
static void install_signal_handler(int signal = 0)
```

Installs the signal handler for the specified signal.

Parameters

signal – The signal to install the signal handler for. If signal is 0, the signal handler is installed for all existing signals.

```
static void register_global_signal_handler(int signal, std::function<void(int)> handler,  
bool overwrite = false)
```

Registers a global signal handler for the specified signal.

Parameters

- **signal** – The signal to register the global signal handler for.
- **handler** – The global signal handler function.
- **overwrite** – If true, overwrites any existing global signal handler for the specified signal.

```
static void register_signal_handler(void *context, int signal, std::function<void(void*,  
int)> handler, bool overwrite = false)
```

Registers a context-specific signal handler for the specified signal.

Parameters

- **context** – The context to register the signal handler for.
- **signal** – The signal to register the signal handler for.
- **handler** – The signal handler function.

- **overwrite** – If true, overwrites any existing signal handler for the specified context and signal.

```
static void unregister_global_signal_handler(int signal)
```

Unregisters the global signal handler for the specified signal.

Parameters

signal – The signal to unregister the global signal handler for.

```
static void unregister_signal_handler(void *context, int signal)
```

Unregisters the context-specific signal handler for the specified context and signal.

Parameters

- **context** – The context to unregister the signal handler for.
- **signal** – The signal to unregister the signal handler for.

```
static void clear_all_signal_handlers()
```

Clears all signal handlers.

```
static void clear_global_signal_handlers()
```

Clears all global signal handlers.

```
static void clear_signal_handlers()
```

Clears all context-specific signal handlers.

© Copyright 2022-2024, NVIDIA.. PDF Generated on 06/06/2024