GCCTrace

Generated by Doxygen 1.8.3.1

Thu Apr 17 2014 22:35:47

Contents

Chapter 1

Class Index

1.1 Class List

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

Chapter 2

File Index

2.1 File List

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

Chapter 3

Class Documentation

3.1 call_stack Struct Reference

Full calling stack.

`#include <gcctracer.h>`

Public Attributes

- unsigned int num_frames
- stack_frame * frames

3.1.1 Detailed Description

Full calling stack.

Structure that contains a full calling stack

3.1.2 Member Data Documentation

3.1.2.1 stack_frame* call_stack::frames

Stack frames array

3.1.2.2 unsigned int call_stack::num_frames

How deep is the stack

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

- include/gcctracer.h

3.2 stack_frame Struct Reference

Stack frame structure.

`#include <gcctracer.h>`

Public Attributes

- unsigned long int time
- unsigned long int thread
- unsigned long int used_bytes
- void * this_fn
- void * call_site

3.2.1 Detailed Description

Stack frame structure.

Stack frame containing the traced data

3.2.2 Member Data Documentation

3.2.2.1 void* stack_frame::call_site

Place in the code where this_fn was called

3.2.2.2 void* stack_frame::this_fn

Pointer to the invoked function

3.2.2.3 unsigned long int stack_frame::thread

Thread ID

3.2.2.4 unsigned long int stack_frame::time

Timestamp

3.2.2.5 unsigned long int stack_frame::used_bytes

Allocated memory in bytes

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

- include/gcctracer.h

Chapter 4

File Documentation

4.1 include/gcctracer.h File Reference

Main file to include gcctrace's functionalities.

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

Classes

- struct stack_frame

  Stack frame structure.
- struct call_stack

  Full calling stack.

Typedefs

- typedef struct stack_frame stack_frame

  Stack frame structure.
- typedef struct call_stack call_stack

  Full calling stack.

Functions

- void _gcc_trace_clone_current_call_stack (call_stack *stack)
- void _gcc_trace_free_call_stack (call_stack *stack)
- void _gcc_trace_print_call_stack (call_stack *stack)
- void _gcc_trace_dump_history_buffer (const char *file_name)
- unsigned long int _gcc_trace_get_heap_memory ()

4.1.1 Detailed Description

Main file to include gcctrace's functionalities.

Author

Renato Grottesi

Date

7 Apr 2014

4.1.2 Typedef Documentation

4.1.2.1 typedef struct call_stack call_stack

Full calling stack.

Structure that contains a full calling stack

4.1.2.2 typedef struct stack_frame stack_frame

Stack frame structure.

Stack frame containing the traced data

4.1.3 Function Documentation

4.1.3.1 void _gcc_trace_clone_current_call_stack (call_stack * *stack*)

Copy the current call stack inside the input parameter. This function allocates stack's internal memory. Please call _gcc_trace_free_call_stack to free the internal memory.

Parameters

stack	The call stack object where to clone the curren call stack

See Also

_gcc_trace_free_call_stack

4.1.3.2 void _gcc_trace_dump_history_buffer (const char * *file_name*)

Dump the internal circular buffer containing the history of the last n function invocations, where n is an hardcoded size for the buffer capacity.

Parameters

file_name	The name of the file where to dump the data

4.1.3.3 void _gcc_trace_free_call_stack (call_stack * *stack*)

Free the internal memory allocated by _gcc_trace_clone_current_call_stack

Parameters

stack	The call stack object to free

See Also

_gcc_trace_clone_current_call_stack

4.1.3.4 unsigned long int _gcc_trace_get_heap_memory ( )

Return the total heap memory allocated

Returns

total bytes allocated in heap

4.1.3.5 void _gcc_trace_print_call_stack (call_stack * *stack*)

Prints a call stack cloned by _gcc_trace_clone_current_call_stack.

Parameters

stack	The stack to print in stderr

See Also

_gcc_trace_clone_current_call_stack

Index