

# Libreria Traceback

1.0

Generated by Doxygen 1.8.6

Thu May 7 2015 18:47:01



# Contents

<b>1</b>	<b>Bug List</b>	<b>1</b>
<b>2</b>	<b>Data Structure Index</b>	<b>3</b>
2.1	Data Structures . . . . .	3
<b>3</b>	<b>File Index</b>	<b>5</b>
3.1	File List . . . . .	5
<b>4</b>	<b>Data Structure Documentation</b>	<b>7</b>
4.1	argsym_t Struct Reference . . . . .	7
4.1.1	Detailed Description . . . . .	7
4.2	functsym_t Struct Reference . . . . .	7
4.2.1	Detailed Description . . . . .	7
<b>5</b>	<b>File Documentation</b>	<b>9</b>
5.1	add_one.h File Reference . . . . .	9
5.1.1	Detailed Description . . . . .	9
5.2	add_one_test.c File Reference . . . . .	9
5.2.1	Detailed Description . . . . .	9
5.3	alarming_test.c File Reference . . . . .	10
5.3.1	Detailed Description . . . . .	10
5.4	evil_test.c File Reference . . . . .	10
5.4.1	Detailed Description . . . . .	11
5.5	func_aux.h File Reference . . . . .	11
5.5.1	Detailed Description . . . . .	11
5.5.2	Function Documentation . . . . .	12
5.5.2.1	get_char_value . . . . .	12
5.5.2.2	get_double_value . . . . .	12
5.5.2.3	get_ebp . . . . .	12
5.5.2.4	get_float_value . . . . .	12
5.5.2.5	get_function_return . . . . .	13
5.5.2.6	get_int_value . . . . .	13
5.5.2.7	get_next_ebp . . . . .	13

5.5.2.8	<a href="#">get_string_array_value</a>	14
5.5.2.9	<a href="#">get_string_value</a>	15
5.5.2.10	<a href="#">get_void_value</a>	15
5.6	<a href="#">string_aux.c File Reference</a>	15
5.6.1	Detailed Description	16
5.6.2	Function Documentation	16
5.6.2.1	<a href="#">ischar</a>	16
5.6.2.2	<a href="#">strlen</a>	16
5.6.2.3	<a href="#">substr</a>	16
5.7	<a href="#">string_aux.h File Reference</a>	17
5.7.1	Detailed Description	17
5.7.2	Function Documentation	17
5.7.2.1	<a href="#">ischar</a>	17
5.7.2.2	<a href="#">strlen</a>	18
5.7.2.3	<a href="#">substr</a>	18
5.8	<a href="#">traceback.c File Reference</a>	18
5.8.1	Detailed Description	19
5.8.2	Function Documentation	19
5.8.2.1	<a href="#">isprint_string</a>	19
5.8.2.2	<a href="#">locate_function</a>	19
5.8.2.3	<a href="#">print_args</a>	20
5.8.2.4	<a href="#">print_func</a>	20
5.8.2.5	<a href="#">search_function</a>	21
5.8.2.6	<a href="#">traceback</a>	21
5.9	<a href="#">traceback.h File Reference</a>	21
5.9.1	Detailed Description	21
5.9.2	Function Documentation	22
5.9.2.1	<a href="#">traceback</a>	22
5.10	<a href="#">traceback_globals.c File Reference</a>	22
5.10.1	Detailed Description	22
5.10.2	Variable Documentation	22
5.10.2.1	<a href="#">functions</a>	22
5.11	<a href="#">traceback_internal.h File Reference</a>	23
5.11.1	Detailed Description	23
5.11.2	Macro Definition Documentation	23
5.11.2.1	<a href="#">ARGS_MAX_NAME</a>	23
5.11.2.2	<a href="#">ARGS_MAX_NUM</a>	24
5.11.2.3	<a href="#">FUNCTS_MAX_NAME</a>	24
5.11.2.4	<a href="#">FUNCTS_MAX_NUM</a>	24
5.11.2.5	<a href="#">MAX_FUNCTION_SIZE_BYTES</a>	24

<b>CONTENTS</b>	<b>v</b>
5.12 <a href="#">verify_test.c File Reference</a> . . . . .	24
5.12.1 <a href="#">Detailed Description</a> . . . . .	24
<b><a href="#">Index</a></b>	<b>25</b>



# Chapter 1

## Bug List

File [func\\_aux.h](#)

No conocidos

File [string\\_aux.c](#)

No conocidos

File [string\\_aux.h](#)

No conocidos

File [traceback.c](#)

No conocidos

File [traceback.h](#)

No conocidos

File [traceback\\_globals.c](#)

No known bugs.





## Chapter 2

# Data Structure Index

### 2.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">argsym_t</a>	Structure to hold that data for a function argument . . . . .	<a href="#">7</a>
<a href="#">functsym_t</a>	Structure to hold the data representing a function . . . . .	<a href="#">7</a>



## Chapter 3

# File Index

### 3.1 File List

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

<a href="#">add_one.h</a>	Definitions for the add_one functions . . . . .	9
<a href="#">add_one_test.c</a>	Test program for the add_one functions . . . . .	9
<a href="#">alarming_test.c</a>	Test code for the traceback function . . . . .	10
<a href="#">contracts.h</a>	. . . . .	??
<a href="#">evil_test.c</a>	Test code for the traceback function . . . . .	10
<a href="#">func_aux.h</a>	Funciones auxiliares assembly . . . . .	11
<a href="#">string_aux.c</a>	Funciones auxiliares de String . . . . .	15
<a href="#">string_aux.h</a>	Funciones prototipo auxiliares de String . . . . .	17
<a href="#">traceback.c</a>	La funcion Traceback . . . . .	18
<a href="#">traceback.h</a>	Funciones prototipo de la libreria traceback . . . . .	21
<a href="#">traceback_globals.c</a>	Global "functions" variable for traceback . . . . .	22
<a href="#">traceback_internal.h</a>	Function prototype for traceback library and types . . . . .	23
<a href="#">verify_test.c</a>	Test output format for the traceback function . . . . .	24



## Chapter 4

# Data Structure Documentation

### 4.1 `argsym_t` Struct Reference

a structure to hold that data for a function argument

```
#include <traceback_internal.h>
```

#### Data Fields

- int **type**
- int **offset**
- char **name** [ARGS\_MAX\_NAME]

#### 4.1.1 Detailed Description

a structure to hold that data for a function argument

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

- [traceback\\_internal.h](#)

### 4.2 `functsym_t` Struct Reference

a structure to hold the data representing a function

```
#include <traceback_internal.h>
```

#### Data Fields

- void \* **addr**
- char **name** [FUNCTS\_MAX\_NAME]
- [argsym\\_t](#) **args** [ARGS\_MAX\_NUM]

#### 4.2.1 Detailed Description

a structure to hold the data representing a function

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

- [traceback\\_internal.h](#)

## Chapter 5

# File Documentation

### 5.1 `add_one.h` File Reference

provides the definitions for the `add_one` functions

#### Functions

- int **add\_one1** (int)
- int **add\_one2** (int)

#### 5.1.1 Detailed Description

provides the definitions for the `add_one` functions

#### Author

Michael Ashley-Rollman(mpa)

### 5.2 `add_one_test.c` File Reference

a test program for the `add_one` functions

```
#include <stdio.h>
#include "add_one.h"
```

#### Functions

- int **main** ()

#### 5.2.1 Detailed Description

a test program for the `add_one` functions This program provides a brief demonstration of the usage of the `add_one` functions.

#### Author

Michael Ashley-Rollman(mpa)

## 5.3 alarming\_test.c File Reference

Test code for the traceback function.

```
#include "traceback.h"
#include <errno.h>
#include <stdio.h>
#include <unistd.h>
#include <string.h>
#include <stdlib.h>
#include <signal.h>
#include <sys/mman.h>
```

### Macros

- #define **MEGABYTE** (1024\*1024)

### Functions

- void **alarming** (char \*str, char \*notstr)
- void **dingdong** (int sig)
- int **main** ()

### Variables

- int **work\_done** = 0

#### 5.3.1 Detailed Description

Test code for the traceback function. This test calls [traceback\(\)](#) from an alternate signal stack.

Don't try to understand the operation of sigaltstack() from the Linux man page since it is incoherent (at least through glibc 2.6). This is better: <http://www.opengroup.org/onlinepubs/000095399/functions/sigaltstack.-html>

### Author

Dave Eckhardt (de0u)

## 5.4 evil\_test.c File Reference

Test code for the traceback function.

```
#include "traceback.h"
#include <errno.h>
#include <stdio.h>
#include <unistd.h>
#include <string.h>
#include <stdlib.h>
```



## Functions

- void **foo** (char \*str)
- int **main** ()

### 5.4.1 Detailed Description

Test code for the traceback function. This test calls a function foo() with an unterminated string. foo() in turn calls traceback.

#### Author

Michael Ashley-Rollman (mpa)

## 5.5 func\_aux.h File Reference

Funciones auxiliares assembly.

## Functions

- int \* [get\\_ebp](#) ()  
*funcion get\_ebp*
- int \* [get\\_next\\_ebp](#) (int \*)  
*funcion get\_next\_ebp*
- int [get\\_int\\_value](#) (int \*, int)  
*funcion get\_int\_value*
- char [get\\_char\\_value](#) (int \*, int)  
*funcion get\_char\_value*
- float [get\\_float\\_value](#) (int \*, int)  
*funcion get\_float\_value*
- double [get\\_double\\_value](#) (int \*, int)  
*funcion get\_double\_value*
- char \* [get\\_string\\_value](#) (int \*, int)  
*funcion get\_string\_value*
- void \* [get\\_void\\_value](#) (int \*, int)  
*funcion get\_void\_value*
- char \*\* [get\\_string\\_array\\_value](#) (int \*, int)  
*funcion get\_string\_array\_value*
- char \* [get\\_function\\_return](#) (int \*)  
*funcion get\_function\_return*

### 5.5.1 Detailed Description

Funciones auxiliares assembly. Este archivo consta de las implementacion de las funciones definidas en los prototipos

#### Author

Calvo Alfaro, Angelo Alejandro (acalvoa)

**Bug** No conocidos

## 5.5.2 Function Documentation

### 5.5.2.1 char get\_char\_value ( int \*, int )

funcion get\_char\_value

funcion que recupera el valor del parametro relativo a ebp a partir de un offset

#### Parameters

<i>int*</i>	proporciona un puntero al stack frame
<i>int</i>	correponde al offset relativo al ebp

#### Returns

la funcion retorna el valor char del argumento

Referenced by print\_args().

### 5.5.2.2 double get\_double\_value ( int \*, int )

funcion get\_double\_value

funcion que recupera el valor del parametro relativo a ebp a partir de un offset

#### Parameters

<i>int*</i>	proporciona un puntero al stack frame
<i>int</i>	correponde al offset relativo al ebp

#### Returns

la funcion retorna el valor double del argumento

Referenced by print\_args().

### 5.5.2.3 int\* get\_ebp ( )

funcion get\_ebp

funcion que recupera un puntero a la posicion del stack frame

#### Returns

la funcion retorna un puntero al stack frame que llamo a la funcion traceback

Referenced by traceback().

### 5.5.2.4 float get\_float\_value ( int \*, int )

funcion get\_float\_value

funcion que recupera el valor del parametro relativo a ebp a partir de un offset

#### Parameters

<i>int*</i>	proporciona un puntero al stack frame
<i>int</i>	correponde al offset relativo al ebp

**Returns**

la funcion retorna el valor float del argumento

Referenced by print\_args().

**5.5.2.5 char\* get\_function\_return ( int \* )**

funcion get\_function\_return

funcion que recupera el valor de retorno de la funcion de ebp a partir de un stack frame

**Parameters**

<i>int*</i>	proporciona un puntero al stack frame
-------------	---------------------------------------

**Returns**

la funcion retorna un puntero al vaor de retorno

Referenced by search\_function().

**5.5.2.6 int get\_int\_value ( int \* , int )**

funcion get\_int\_value

funcion que recupera el valor del parametro relativo a ebp a partir de un offset

**Parameters**

<i>int*</i>	proporciona un puntero al stack frame
<i>int</i>	correponde al offset relativo al ebp

**Returns**

la funcion retorna el valor int del argumento

Referenced by print\_args().

**5.5.2.7 int\* get\_next\_ebp ( int \* )**

funcion get\_next\_ebp

funcion que recupera un puntero al siguiente posicion del stack frame a partir de uno ya existente

**Parameters**

<i>int*</i>	proporciona un puntero al stack frame
-------------	---------------------------------------

**Returns**

la funcion retorna un puntero al siguiente stack frame

Referenced by traceback().

5.5.2.8 `char** get_string_array_value ( int *, int )`

funcion `get_string_array_value`

funcion que recupera el valor del parametro relativo a ebp a partir de un offset

## Parameters

<i>int*</i>	proporciona un puntero al stack frame
<i>int</i>	correponde al offset relativo al ebp

## Returns

la funcion retorna el valor char\*\* del argumento

Referenced by print\_args().

## 5.5.2.9 char\* get\_string\_value ( int \*, int )

funcion get\_string\_value

funcion que recupera el valor del parametro relativo a ebp a partir de un offset

## Parameters

<i>int*</i>	proporciona un puntero al stack frame
<i>int</i>	correponde al offset relativo al ebp

## Returns

la funcion retorna el valor string del argumento

Referenced by print\_args().

## 5.5.2.10 void\* get\_void\_value ( int \*, int )

funcion get\_void\_value

funcion que recupera el valor del parametro relativo a ebp a partir de un offset

## Parameters

<i>int*</i>	proporciona un puntero al stack frame
<i>int</i>	correponde al offset relativo al ebp

## Returns

la funcion retorna el valor void\* del argumento

Referenced by print\_args().

## 5.6 string\_aux.c File Reference

Funciones auxiliares de String.

```
#include "string.h"
#include "stdlib.h"
```

### Functions

- `size_t strlen (const char *str)`  
*funcion para contar largo de strings - strlen*

- char \* **substr** (char \*cadena, int comienzo, int longitud)  
*funcion para cortar strings - substr*
- int **ischar** (char \*cadena)  
*funcion para identificar char string*

### 5.6.1 Detailed Description

Funciones auxiliares de String. Este archivo consta de las implementacion de las funciones definidas en los prototipos

#### Author

Calvo Alfaro, Angelo Alejandro (acalvoa)

**Bug** No conocidos

### 5.6.2 Function Documentation

#### 5.6.2.1 int ischar ( char \* *cadena* )

funcion para identificar char string

Esta funcion identifica si el char\* es un string

#### Parameters

<i>cadena</i>	Representa el string a analizar
---------------	---------------------------------

#### Returns

La funcion retorna un 1 si es char o un 0 si es string

Referenced by print\_args().

#### 5.6.2.2 size\_t strlen ( const char \* *str* )

funcion para contar largo de strings - strlen

Esta funcion cuenta la cantidad de caracteres validos presentes en un string.

#### Parameters

<i>str</i>	representa el string que se obtendra su largo.
------------	--

#### Returns

La funcion retorna el largo del string suministrado.

Referenced by isprint\_string(), locate\_function(), print\_args(), and substr().

#### 5.6.2.3 char\* substr ( char \* *cadena*, int *comienzo*, int *longitud* )

funcion para cortar strings - substr

Esta funcion corta un string en una cantidad definida por el usuario.

## Parameters

<i>cadena</i>	Representa el string a cortar
<i>comienzo</i>	Indica la posicion del arreglo char* en donde se comienza a cortar la cadena.
<i>longitud</i>	Indica el largo que tendra la cadena

## Returns

La funcion retorna el string cortado de acuerdo al largo especificado y se añade al final de este puntos suspensivos.

References strlen().

Referenced by print\_args().

## 5.7 string\_aux.h File Reference

Funciones prototipo auxiliares de String.

```
#include "string.h"
#include "stdlib.h"
```

## Functions

- `size_t strlen` (const char \*str)  
*funcion para contar largo de strings - strlen*
- `char * substr` (char \*cadena, int comienzo, int longitud)  
*funcion para cortar strings - substr*
- `int ischar` (char \*cadena)  
*funcion para identificar char string*

### 5.7.1 Detailed Description

Funciones prototipo auxiliares de String. Este archivo consta de dos funciones auxiliares para la manipulacion de String que son strlen y substr. Estas funciones son necesarias para el correcto funcionamiento de traceback.

## Author

Calvo Alfaro, Angelo Alejandro (acalvoa)

**Bug** No conocidos

### 5.7.2 Function Documentation

#### 5.7.2.1 int ischar ( char \* *cadena* )

funcion para identificar char string

Esta funcion identifica si el char\* es un string

**Parameters**

<i>cadena</i>	Representa el string a analizar
---------------	---------------------------------

**Returns**

La funcion retorna un 1 si es char o un 0 si es string

Referenced by `print_args()`.

**5.7.2.2 `size_t strlen ( const char * str )`**

funcion para contar largo de strings - `strlen`

Esta funcion cuenta la cantidad de caracteres validos presentes en un string.

**Parameters**

<i>str</i>	representa el string que se obtendra su largo.
------------	--

**Returns**

La funcion retorna el largo del string suministrado.

Referenced by `isprint_string()`, `locate_function()`, `print_args()`, and `substr()`.

**5.7.2.3 `char* substr ( char * cadena, int comienzo, int longitud )`**

funcion para cortar strings - `substr`

Esta funcion corta un string en una cantidad definida por el usuario.

**Parameters**

<i>cadena</i>	Representa el string a cortar
<i>comienzo</i>	Indica la posicion del arreglo <code>char*</code> en donde se comienza a cortar la cadena.
<i>longitud</i>	Indica el largo que tendra la cadena

**Returns**

La funcion retorna el string cortado de acuerdo al largo especificado y se añade al final de este puntos suspensivos.

References `strlen()`.

Referenced by `print_args()`.

## 5.8 `traceback.c` File Reference

La funcion `Traceback`.

```
#include "traceback_internal.h"
#include "func_aux.h"
#include "ctype.h"
#include "string_aux.h"
```



## Functions

- int [locate\\_function](#) (int num, char \*instruction)  
*funcion locate\_function*
- int [search\\_function](#) (int \*ebp)  
*funcion search\_function*
- int [isprint\\_string](#) (char \*string)  
*funcion isprint\_string*
- void [print\\_args](#) (const [functsym\\_t](#) func, int i, int \*ebp, int first)  
*funcion print\_args*
- void [print\\_func](#) (int pos, int \*ebp, int \*main, int \*first)  
*funcion print\_func*
- void [traceback](#) (FILE \*fp)  
*funcion print\_func*

### 5.8.1 Detailed Description

La funcion Traceback. Este archivo contiene la funcion traceback y todas las funciones auxiliares adicionales necesarias para la ejecución de traceback. las funciones del archivo corresponden a funciones necesarias por el la funcion traceback para operar de forma recursiva a traves de la tabla de funciones generada. Las busquedas en esta tabla son efectuadas a traves de metodos recursivos en base a las especificaciones indicadas en el documento de requerimiento.

#### Author

Calvo Alfaro, Angelo Alejandro (acalvoa)

**Bug** No conocidos

### 5.8.2 Function Documentation

#### 5.8.2.1 int isprint\_string ( char \* string )

funcion isprint\_string

funcion que indica si un string valido apra ser impresa

#### Parameters

<i>string</i>	provee el string que debe ser analizado
---------------	---

#### Returns

la funcion retorna un 1 en caso de que el string se pueda imprimir o un 0 en caso de que no pueda ser impreso.

References strlen().

Referenced by print\_args().

#### 5.8.2.2 int locate\_function ( int num, char \* instruction )

funcion locate\_function

funcion que busca de forma iterativa en la tabla de funciones a partir de una direccion de memoria obtenida de las instrucciones previas a la direccion de retorno con respecto a ebp

**Parameters**

<i>num</i>	indica el numero incremental recursivo. Siempre debe ser cero al efectuarse la primera llamada.
<i>instruction</i>	corresponde a un puntero de 1 byte a la direccion de llamada de la funcion a buscar

**Returns**

la funcion retorna el indice de coincidencia con la tabla de funciones.

References strlen().

Referenced by search\_function().

**5.8.2.3 void print\_args ( const functsym\_t func, int i, int \* ebp, int first )**

funcion print\_args

print\_args es una funcion recursiva que se encarga de imprimir en pantalla todos los argumentos de una funcion que ha sido identificada a partir de su stack frame el formato bajo el cual se imprimen dichos parametros corresponde al requerido por el documento de requerimientos.

**Parameters**

<i>func</i>	es la funcion que ha sido identificada de la tabla
<i>i</i>	corresponde al valor recursivo, siempre debe ser 0
<i>ebp</i>	correponde al puntero hacia el stack frame
<i>first</i>	es un puntero que debe ser siempre 1.

**Returns**

la funcion no retorna valores.

References get\_char\_value(), get\_double\_value(), get\_float\_value(), get\_int\_value(), get\_string\_array\_value(), get\_string\_value(), get\_void\_value(), ischar(), isprint\_string(), strlen(), and substr().

Referenced by print\_func().

**5.8.2.4 void print\_func ( int pos, int \* ebp, int \* main, int \* first )**

funcion print\_func

print\_func es una funcion que a partir de la posicion de la funcion en la tabla y del stack frame idetificado imprime la funcion de acuerdo al formato planteado en la hoja de requerimientos.

**Parameters**

<i>pos</i>	correponde a la posicion dentro de la tabla
<i>ebp</i>	correponde al puntero hacia el stack frame
<i>main</i>	es un puntero que indica si ya hubo coicidencia con main siempre debe ser 0
<i>first</i>	es un puntero que debe ser siempre 1.

**Returns**

la funcion no retorna valores.

References print\_args().

Referenced by traceback().

5.8.2.5 int search\_function ( int \* *ebp* )

funcion search\_function

funcion que busca a que funcion de la tabla corresponde un stack frame determinado.

## Parameters

<i>ebp</i>	indica el stack frame a analizar
------------	----------------------------------

## Returns

la funcion retorna la posicion de la funcion que coincide de la tabla de funciones o un -1 en caso de no existir coincidencias

References get\_function\_return(), and locate\_function().

Referenced by traceback().

5.8.2.6 void traceback ( FILE \* *fp* )

funcion print\_func

print\_func es una funcion recursiva que se encarga de recorrer el stack, recorriendo los stack frames hasta el wrapper. Para esto hace uso del registro ebp y mueve el puntero hasta que no se encuentran mas funciones despues del wrapper que corresponde a un indice -1.

## Parameters

* <i>fp</i>	correponde al puntero a un stack
-------------	----------------------------------

## Returns

la funcion no retorna valores.

References get\_ebp(), get\_next\_ebp(), print\_func(), and search\_function().

## 5.9 traceback.h File Reference

funciones prototipo de la libreria traceback

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

## Functions

- void [traceback](#) (FILE \*)  
funcion print\_func

## 5.9.1 Detailed Description

funciones prototipo de la libreria traceback No borrar este archivo. Contiene las definiciones abstractas de la libreria trace

## Author

Calvo Alfaro, Angelo Alejandro (acalvoa)

**Bug** No conocidos

## 5.9.2 Function Documentation

### 5.9.2.1 void traceback ( FILE \* fp )

funcion print\_func

print\_func es una funcion recursiva que se encarga de recorrer el stack, recorriendo los stack frames hasta el wrapper. Para esto hace uso del registro ebp y mueve el puntero hasta que no se encuentran mas funciones despues del wrapper que corresponde a un indice -1.

#### Parameters

*fp	correponde al puntero a un stack
-----	----------------------------------

#### Returns

la funcion no retorna valores.

References get\_ebp(), get\_next\_ebp(), print\_func(), and search\_function().

## 5.10 traceback\_globals.c File Reference

global "functions" variable for traceback

```
#include "traceback_internal.h"
#include <stdio.h>
```

#### Variables

- const [functsym\\_t](#) **functions** [FUNCTS\_MAX\_NUM]

### 5.10.1 Detailed Description

global "functions" variable for traceback This contains the actual definition of the global traceback variable. It is filled with the information necessary for symtabgen.pl to later fill the table in.

#### Author

Michael Ashley-Rollman (mpa)

**Bug** No known bugs.

### 5.10.2 Variable Documentation

#### 5.10.2.1 const functsym\_t functions[FUNCTS\_MAX\_NUM]

#### Initial value:

```
=
{{(void *)sizeof(functsym_t),
{(unsigned char) (FUNCTS_MAX_NUM % 256),
(unsigned char) (FUNCTS_MAX_NUM / 256),
(char)ARGS_MAX_NUM,
(char)sizeof(argsym_t),
(char)FUNCTS_MAX_NAME,
(char)ARGS_MAX_NAME }}}}
```

## 5.11 traceback\_internal.h File Reference

Function prototype for traceback library and types.

```
#include "traceback.h"
```

### Data Structures

- struct [argsym\\_t](#)  
*a structure to hold that data for a function argument*
- struct [functsym\\_t](#)  
*a structure to hold the data representing a function*

### Macros

- #define **TYPE\_CHAR** 0
- #define **TYPE\_INT** 1
- #define **TYPE\_FLOAT** 2
- #define **TYPE\_DOUBLE** 3
- #define **TYPE\_STRING** 4
- #define **TYPE\_STRING\_ARRAY** 5
- #define **TYPE\_VOIDSTAR** 6
- #define **TYPE\_UNKNOWN** -1
- #define **ARGS\_MAX\_NUM**
- #define **ARGS\_MAX\_NAME**
- #define **FUNCTS\_MAX\_NUM**
- #define **FUNCTS\_MAX\_NAME**
- #define **MAX\_FUNCTION\_SIZE\_BYTES**

### Variables

- const [functsym\\_t](#) **functions** [FUNCTS\_MAX\_NUM]

#### 5.11.1 Detailed Description

Function prototype for traceback library and types. This file should only be included by the files that form the traceback library. Test programs should include [traceback.h](#) instead.

DO NOT EDIT THIS FILE DO NOT EDIT THIS FILE DO NOT EDIT THIS FILE DO NOT EDIT THIS FILE DO NOT EDIT THIS FILE

Copyright (c) 2014 Carnegie Mellon University

Author

Michael Ashley-Rollman(mpa)

#### 5.11.2 Macro Definition Documentation

##### 5.11.2.1 #define ARGS\_MAX\_NAME

**Value:**

```
24      /* The maximum length of the name of
        an argument */
```

#### 5.11.2.2 #define ARGS\_MAX\_NUM

##### Value:

```
6      /* The maximum number of arguments
        a single function can have */
```

#### 5.11.2.3 #define FUNCTS\_MAX\_NAME

##### Value:

```
60     /* The maximum length of the name of
        a function */
```

#### 5.11.2.4 #define FUNCTS\_MAX\_NUM

##### Value:

```
4096  /* The maximum number of functions
        that traceback will support */
```

#### 5.11.2.5 #define MAX\_FUNCTION\_SIZE\_BYTES

##### Value:

```
1048576 /* The maximum size in bytes
        of a function in memory */
```

## 5.12 verify\_test.c File Reference

Test output format for the traceback function.

```
#include "traceback.h"
```

### Functions

- void **f8** (int i, char \*\*f)
- int **main** ()

#### 5.12.1 Detailed Description

Test output format for the traceback function. This test calls a few functions to test a bunch of different outputs.

##### Author

Michael Ashley-Rollman(mpa)

# Index

ARGS\_MAX\_NAME  
    [traceback\\_internal.h, 23](#)

ARGS\_MAX\_NUM  
    [traceback\\_internal.h, 23](#)

[add\\_one.h, 9](#)

[add\\_one\\_test.c, 9](#)

[alarming\\_test.c, 10](#)

[argsym\\_t, 7](#)

[evil\\_test.c, 10](#)

FUNCTS\_MAX\_NAME  
    [traceback\\_internal.h, 24](#)

FUNCTS\_MAX\_NUM  
    [traceback\\_internal.h, 24](#)

[func\\_aux.h, 11](#)

[get\\_char\\_value, 12](#)

[get\\_double\\_value, 12](#)

[get\\_ebp, 12](#)

[get\\_float\\_value, 12](#)

[get\\_function\\_return, 13](#)

[get\\_int\\_value, 13](#)

[get\\_next\\_ebp, 13](#)

[get\\_string\\_array\\_value, 13](#)

[get\\_string\\_value, 15](#)

[get\\_void\\_value, 15](#)

[functions](#)  
    [traceback\\_globals.c, 22](#)

[functsym\\_t, 7](#)

[get\\_char\\_value](#)  
    [func\\_aux.h, 12](#)

[get\\_double\\_value](#)  
    [func\\_aux.h, 12](#)

[get\\_ebp](#)  
    [func\\_aux.h, 12](#)

[get\\_float\\_value](#)  
    [func\\_aux.h, 12](#)

[get\\_function\\_return](#)  
    [func\\_aux.h, 13](#)

[get\\_int\\_value](#)  
    [func\\_aux.h, 13](#)

[get\\_next\\_ebp](#)  
    [func\\_aux.h, 13](#)

[get\\_string\\_array\\_value](#)  
    [func\\_aux.h, 13](#)

[get\\_string\\_value](#)  
    [func\\_aux.h, 15](#)

[get\\_void\\_value](#)  
    [func\\_aux.h, 15](#)

[ischar](#)  
    [string\\_aux.c, 16](#)  
    [string\\_aux.h, 17](#)

[isprint\\_string](#)  
    [traceback.c, 19](#)

[locate\\_function](#)  
    [traceback.c, 19](#)

[print\\_args](#)  
    [traceback.c, 20](#)

[print\\_func](#)  
    [traceback.c, 20](#)

[search\\_function](#)  
    [traceback.c, 20](#)

[string\\_aux.c, 15](#)

[ischar, 16](#)

[strlen, 16](#)

[substr, 16](#)

[string\\_aux.h, 17](#)

[ischar, 17](#)

[strlen, 18](#)

[substr, 18](#)

[strlen](#)  
    [string\\_aux.c, 16](#)  
    [string\\_aux.h, 18](#)

[substr](#)  
    [string\\_aux.c, 16](#)  
    [string\\_aux.h, 18](#)

[traceback](#)  
    [traceback.c, 21](#)  
    [traceback.h, 22](#)

[traceback.c, 18](#)

[isprint\\_string, 19](#)

[locate\\_function, 19](#)

[print\\_args, 20](#)

[print\\_func, 20](#)

[search\\_function, 20](#)

[traceback, 21](#)

[traceback.h, 21](#)

[traceback, 22](#)

[traceback\\_globals.c, 22](#)  
    [functions, 22](#)

[traceback\\_internal.h, 23](#)

[ARGS\\_MAX\\_NAME, 23](#)

[ARGS\\_MAX\\_NUM, 23](#)

[FUNCTS\\_MAX\\_NAME, 24](#)

[FUNCTS\\_MAX\\_NUM, 24](#)

verify\_test.c, [24](#)