

Reference Manual

Generated by Doxygen 1.7.1

Tue Oct 25 2011 20:34:34

Contents

1	Namespace Index	1
1.1	Namespace List	1
2	File Index	3
2.1	File List	3
3	Namespace Documentation	5
3.1	gdb_utils Namespace Reference	5
3.1.1	Detailed Description	6
3.1.2	Function Documentation	6
3.1.2.1	assemble_instructions	6
3.1.2.2	disassemble_count	6
3.1.2.3	disassemble_current_instruction	7
3.1.2.4	disassemble_current_instructions	7
3.1.2.5	disassemble_function	7
3.1.2.6	disassemble_range	7
3.1.2.7	execute_external	8
3.1.2.8	execute_external_output	8
3.1.2.9	execute_output	8
3.1.2.10	normalized_argv	8
3.1.2.11	parse_disassembled_output	8
3.1.2.12	process_mappings	9
3.1.2.13	read_string	9
3.1.2.14	search_functions	9
3.1.2.15	search_processes	9
4	File Documentation	11
4.1	gdb_utils.py File Reference	11

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

[gdb_utils](#) (Various utility functions to work with GDB) 5

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

gdb_utils.py	11
--	----

Chapter 3

Namespace Documentation

3.1 gdb_utils Namespace Reference

Various utility functions to work with GDB.

Functions

- def [read_string](#)
Read an ASCII string from memory.
- def [execute_output](#)
Execute a GDB command with output capture.
- def [execute_external](#)
Execute external command.
- def [execute_external_output](#)
Execute external command with output capture.
- def [search_functions](#)
Search program functions and return their names and addresses.
- def [search_processes](#)
Search running processes and return their info.
- def [parse_disassembled_output](#)
Parse disassembled output (internal function).
- def [disassemble_function](#)
Disassemble a function.
- def [disassemble_range](#)
Disassemble a range.
- def [disassemble_count](#)

Disassemble a variable number of instruction.

- def [disassemble_current_instruction](#)
Disassemble and return the current instruction (pointed by the program counter register).
- def [disassemble_current_instructions](#)
Disassemble a variable number of instruction starting from the current instruction (pointed by the program counter register).
- def [process_mappings](#)
Get process memory mapping.
- def [assemble_instructions](#)
Assemble x86/x64 assembly instructions and return a buffer containing the assembled machine code.
- def [normalized_argv](#)
Get the normalized system arguments to fix a little (IMHO) gdb bug: when the program is executed with no arguments sys.argv is equal to [""], in this case the function returns [], otherwise returns sys.argv immutated.

3.1.1 Detailed Description

Various utility functions to work with GDB. This package provides functions not included in the default gdb module.

3.1.2 Function Documentation

3.1.2.1 def gdb_utils::assemble_instructions (*instructions*)

Assemble x86/x64 assembly instructions and return a buffer containing the assembled machine code.

Parameters

instructions (str) assembly instructions separated by a newline (basically an assembly listing)

Returns

a buffer containing the assembled machine code

3.1.2.2 def gdb_utils::disassemble_count (*start*, *count*, *regex* = " ")

Disassemble a variable number of instruction.

Parameters

start (int) start address

count (int) total number of instructions to disassemble

regex (str) optional regular expression applied to the instruction mnemonic

Returns

list of instructions represented by a dictionary address->instr_code

3.1.2.3 def gdb_utils::disassemble_current_instruction (*regex* = ")

Disassemble and return the current instruction (pointed by the program counter register).

Parameters

regex (str) optional regular expression applied to the instruction mnemonic

Returns

the current instruction represented by a dictionary address->instr_code

3.1.2.4 def gdb_utils::disassemble_current_instructions (*count*, *regex* = ")

Disassemble a variable number of instruction starting from the current instruction (pointed by the program counter register).

Parameters

count (int) total number of instructions to disassemble

regex (str) optional regular expression applied to the instruction mnemonic

Returns

list of instructions represented by a dictionary address->instr_code

3.1.2.5 def gdb_utils::disassemble_function (*func_name*, *regex* = ")

Disassemble a function.

Parameters

func_name (str) name of the function to disassemble

regex (str) optional regular expression applied to the instruction mnemonic

Returns

list of instructions represented by a dictionary address->instr_code

3.1.2.6 def gdb_utils::disassemble_range (*start*, *end*, *regex* = ")

Disassemble a range.

Parameters

start (int) start address

end (int) end address

regex (str) optional regular expression applied to the instruction mnemonic

Returns

list of instructions represented by a dictionary address->instr_code

3.1.2.7 def gdb_utils::execute_external (*command*)

Execute external command.

Parameters

command (str) command string to execute (command + arguments)

3.1.2.8 def gdb_utils::execute_external_output (*command*)

Execute external command with output capture.

Parameters

command (str) command string to execute (command + arguments)

Returns

command output as list of strings

3.1.2.9 def gdb_utils::execute_output (*command*)

Execute a GDB command with output capture.

Parameters

command (str) GDB command

Returns

command output (str)

3.1.2.10 def gdb_utils::normalized_argv ()

Get the normalized system arguments to fix a little (IMHO) gdb bug: when the program is executed with no arguments sys.argv is equal to [""], in this case the function returns [], otherwise returns sys.argv immutated.

Returns

the normalized system arguments

3.1.2.11 def gdb_utils::parse_disassembled_output (*output*, *regex* = " ")

Parse disassembled output (internal function).

Parameters

output (list of strings) disassembled output

regex (str) optional regular expression applied to the instruction mnemonic

Returns

list of instructions represented by a dictionary address->instr_code

3.1.2.12 def gdb_utils::process_mappings (*regex* = ")

Get process memory mapping.

Parameters

regex (str) optional regular expression applied name of the memory area

Returns

a list of hash maps, where every hash map contains informations about a memory area

3.1.2.13 def gdb_utils::read_string (*address*, *count*)

Read an ASCII string from memory.

Parameters

address (int) memory address of the string

count (int) maximum string length

Returns

string read (str)

3.1.2.14 def gdb_utils::search_functions (*regex* = ")

Search program functions and return their names and addresses.

Parameters

regex (str) optional regular expression to search for specific functions

Returns

dictionary of the type func_name->address

3.1.2.15 def gdb_utils::search_processes (*regex* = ")

Search running processes and return their info.

Parameters

regex (str) optional regular expression applied to the process name

Returns

a list of hash maps, where every hash map contains informations about a process

Chapter 4

File Documentation

4.1 gdb_utils.py File Reference

Namespaces

- namespace `gdb_utils`
Various utility functions to work with GDB.

Functions

- def `gdb_utils::read_string`
Read an ASCII string from memory.
- def `gdb_utils::execute_output`
Execute a GDB command with output capture.
- def `gdb_utils::execute_external`
Execute external command.
- def `gdb_utils::execute_external_output`
Execute external command with output capture.
- def `gdb_utils::search_functions`
Search program functions and return their names and addresses.
- def `gdb_utils::search_processes`
Search running processes and return their info.
- def `gdb_utils::parse_disassembled_output`
Parse disassembled output (internal function).
- def `gdb_utils::disassemble_function`
Disassemble a function.

- def `gdb_utils::disassemble_range`
Disassemble a range.
- def `gdb_utils::disassemble_count`
Disassemble a variable number of instruction.
- def `gdb_utils::disassemble_current_instruction`
Disassemble and return the current instruction (pointed by the program counter register).
- def `gdb_utils::disassemble_current_instructions`
Disassemble a variable number of instruction starting from the current instruction (pointed by the program counter register).
- def `gdb_utils::process_mappings`
Get process memory mapping.
- def `gdb_utils::assemble_instructions`
Assemble x86/x64 assembly instructions and return a buffer containing the assembled machine code.
- def `gdb_utils::normalized_argv`
Get the normalized system arguments to fix a little (IMHO) gdb bug: when the program is executed with no arguments `sys.argv` is equal to `[""]`, in this case the function returns `[]`, otherwise returns `sys.argv` immutated.

Index

- assemble_instructions
 - [gdb_utils](#), [6](#)
- disassemble_count
 - [gdb_utils](#), [6](#)
- disassemble_current_instruction
 - [gdb_utils](#), [6](#)
- disassemble_current_instructions
 - [gdb_utils](#), [7](#)
- disassemble_function
 - [gdb_utils](#), [7](#)
- disassemble_range
 - [gdb_utils](#), [7](#)
- execute_external
 - [gdb_utils](#), [7](#)
- execute_external_output
 - [gdb_utils](#), [8](#)
- execute_output
 - [gdb_utils](#), [8](#)
- [gdb_utils](#), [5](#)
 - [assemble_instructions](#), [6](#)
 - [disassemble_count](#), [6](#)
 - [disassemble_current_instruction](#), [6](#)
 - [disassemble_current_instructions](#), [7](#)
 - [disassemble_function](#), [7](#)
 - [disassemble_range](#), [7](#)
 - [execute_external](#), [7](#)
 - [execute_external_output](#), [8](#)
 - [execute_output](#), [8](#)
 - [normalized_argv](#), [8](#)
 - [parse_disassembled_output](#), [8](#)
 - [process_mappings](#), [8](#)
 - [read_string](#), [9](#)
 - [search_functions](#), [9](#)
 - [search_processes](#), [9](#)
- [gdb_utils.py](#), [11](#)
- normalized_argv
 - [gdb_utils](#), [8](#)
- parse_disassembled_output
 - [gdb_utils](#), [8](#)
- process_mappings
 - [gdb_utils](#), [8](#)
- read_string
 - [gdb_utils](#), [9](#)
- search_functions
 - [gdb_utils](#), [9](#)
- search_processes
 - [gdb_utils](#), [9](#)