libgdbc

Generated by Doxygen 1.8.3.1

Tue Oct 29 2013 15:21:39

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 libgdbc_message_stack_t Struct Reference

`#include <core.h>`

Public Attributes

- int **top**
- libgdbc_message_t message_stack [128]

3.1.1 Detailed Description

Message stack

3.1.2 Member Data Documentation

3.1.2.1 libgdbc_message_t libgdbc_message_stack_t::message_stack[128]

Top of the message stack (index)

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

- /home/rene/libgdbc/include/core.h

3.2 libgdbc_message_t Struct Reference

`#include <core.h>`

Public Attributes

- ssize_t **len**
- char * msg
- uint8_t chk

3.2.1 Detailed Description

Structure that saves a gdb message

3.2.2 Member Data Documentation

3.2.2.1 uint8_t libgdbc_message_t::chk

Pointer to the buffer that contains the message

3.2.2.2 char* libgdbc_message_t::msg

Len of the message

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

- /home/rene/libgdbc/include/core.h

3.3 libgdbc_t Struct Reference

```
#include <core.h>
```

Public Attributes

- char * **send_buff**
- ssize_t **max_send_len**
- char * **read_buff**
- ssize_t **max_read_len**
- libgdbc_message_stack_t **message_stack**
- int **fd**
- int **connected**
- int **acks**
- uint8_t * **data**
- ssize_t **data_len**
- uint8_t **architecture**
- register_t * **registers**

3.3.1 Detailed Description

Core "object" that saves the instance of the lib

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

- /home/rene/libgdbc/include/core.h

3.4 libgdbc_x86_64_t Struct Reference

Public Attributes

- uint64_t * **registers**

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

- /home/rene/libgdbc/include/target.h

3.5 parsing_object_t Struct Reference

Public Attributes

- char * **buffer**
- ssize_t **length**
- int **start**
- int **end**
- int **position**
- uint8_t **checksum**
- int **acks**

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

- /home/rene/libgdbc/include/packet.h

3.6 registers_t Struct Reference

```
#include <arch.h>
```

Public Attributes

- char **name** [32]
- uint64_t offset
- uint64_t size
- uint64_t value

3.6.1 Detailed Description

This struct defines a generic register view

3.6.2 Member Data Documentation

3.6.2.1 uint64_t registers_t::offset

The Name of the current register

3.6.2.2 uint64_t registers_t::size

Offset in the data block

3.6.2.3 uint64_t registers_t::value

Size of the register

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

- /home/rene/libgdbc/include/arch.h

Chapter 4

File Documentation

4.1 /home/rene/libgdbc/include/arch.h File Reference

Classes

- struct registers_t

Typedefs

- typedef struct registers_t registers_t

Variables

- registers_t **x86_64** []
- registers_t **x86_32** []

4.1.1 Typedef Documentation

4.1.1.1 typedef struct registers_t registers_t

This struct defines a generic register view

4.2 /home/rene/libgdbc/include/core.h File Reference

```
#include <stdint.h>
#include <netinet/in.h>
#include <sys/socket.h>
#include <netdb.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <stdio.h>
#include "utils.h"
#include "arch.h"
```

Classes

- struct libgdbc_message_t
- struct libgdbc_message_stack_t
- struct libgdbc_t

Macros

- #define **CMD_CONTINUE** "c"
- #define **CMD_READREG** "g"

Typedefs

- typedef struct libgdbc_message_t libgdbc_message_t
- typedef struct libgdbc_message_stack_t libgdbc_message_stack_t
- typedef struct libgdbc_t libgdbc_t

Functions

- int send_command (libgdbc_t *instance, char *command)

  Function sends a command to the gdbserver.
- int send_packet (libgdbc_t *instance)

  sends a packet sends a packet to the established connection
- int read_packet (libgdbc_t *instance)

  Function reads data from the established connection.
- int create_instance (libgdbc_t *instance, uint8_t architecture)

  creates a new instance object (allocates buffers and such)
- int delete_instance (libgdbc_t *instance)

  deletes the given instance (frees buffers)
- int connect_instance (libgdbc_t *instance, const char *host, int port)

  connects to the gdbserver
- int disconnect_instance (libgdbc_t *instance)

  disconnects the instance
- int dump_message_stack (libgdbc_t *instance)

  dumps the whole message stack
- int **step_instance** (libgdbc_t *instance)
- int memread_instance (libgdbc_t *instance, uint64_t address, uint64_t len)

  sends a 'm' packet to the gdbserver and reads the result
- int regread_instance (libgdbc_t *instance)

  sends a 'g' packet to the gdbserver and reads the result
- int continue_instance (libgdbc_t *instance)

  sends a 'c' packet to the gdbserver

4.2.1 Typedef Documentation

4.2.1.1 typedef struct libgdbc_message_stack_t libgdbc_message_stack_t

Message stack

4.2.1.2 typedef struct libgdbc_message_t libgdbc_message_t

Structure that saves a gdb message

4.2.1.3 typedef struct libgdbc_t libgdbc_t

Core "object" that saves the instance of the lib

4.2.2 Function Documentation

4.2.2.1 int connect_instance (libgdbc_t * *instance*, const char * *host*, int *port*)

connects to the gdbserver

Parameters

instance	the "instance" of the current libgdbc session
host	defines the host in string representation
port	of the connection

Returns

a failure code (currently -1) or 0 if call successfully

Function connects the defined host:port kombination to the existing gdbserver instance TODO add connect function with parameters (i.e. qSupported...)

4.2.2.2 int continue_instance (libgdbc_t * *instance*)

sends a 'c' packet to the gdbserver

Parameters

instance	the "instance" of the current libgdbc session

Returns

a failure code (currently -1) or 0 if call successfully

4.2.2.3 int create_instance (libgdbc_t * *instance*, uint8_t *architecture*)

creates a new instance object (allocates buffers and such)

Parameters

instance	the "instance" of the current libgdbc session
architecture	defines the architecure used (registersize, and such)

Returns

a failure code (currently -1) or 0 if call successfully

Function creates a new instance of libgdbc_t

4.2.2.4 int delete_instance (libgdbc_t * *instance*)

deletes the given instance (frees buffers)

Parameters

instance	the "instance" of the current libgdbc session

Returns

a failure code (currently -1) or 0 if call successfully

Function deletes existing instance i.e. frees all allocated memory inside the instance remember it does not free the instance itself

4.2.2.5 int disconnect_instance (libgdbc_t * *instance*)

disconnects the instance

Parameters

instance	the "instance" of the current libgdbc session

Returns

a failure code (currently -1) or 0 if call successfully

4.2.2.6 int dump_message_stack (libgdbc_t * *instance*)

dumps the whole message stack

Parameters

instance	the "instance" of the current libgdbc session

Returns

a failure code (currently -1) or 0 if call successfully

4.2.2.7 int memread_instance (libgdbc_t * *instance*, uint64_t *address*, uint64_t *len*)

sends a 'm' packet to the gdbserver and reads the result

Parameters

instance	the "instance" of the current libgdbc session

Returns

a failure code (currently -1) or 0 if call successfully

4.2.2.8 int read_packet (libgdbc_t * *instance*)

Function reads data from the established connection.

Parameters

instance	the "instance" of the current libgdbc session

Returns

a failure code (currently -1) or 0 if call successfully

4.2.2.9 int regread_instance (libgdbc_t * *instance*)

sends a 'g' packet to the gdbserver and reads the result

Parameters

instance	the "instance" of the current libgdbc session

Returns

a failure code (currently -1) or 0 if call successfully

4.2.2.10 int send_command (libgdbc_t * *instance*, char * *command*)

Function sends a command to the gdbserver.

Parameters

instance	the "instance" of the current libgdbc session
command	the command that will be sent

Returns

a failure code (currently -1) or 0 if call successfully

This function sends the given command to the gdb server it creates the needet checksum and creates the packet i.e. command = 'g' will end in $g#67 instance : instance that defines the current gdb session command : defines the given command

4.2.2.11 int send_packet (libgdbc_t * *instance*)

sends a packet sends a packet to the established connection

Parameters

instance	the "instance" of the current libgdbc session

Returns

a failure code (currently -1) or 0 if call successfully

This function sends the packet that lays in instance->buff and checks the ack from the server instance : defines the current gdb session

4.3 /home/rene/libgdbc/include/libgdbc.h File Reference

```
#include <stdint.h>
```

Macros

- #define **ARCH_X86_64** 0
- #define **ARCH_X86_32** 1

Functions

- int libgdbc_init (uint8_t architecture)
- int **libgdbc_cleanup** ()
- int **libgdbc_connect** (char *server, int port)
- int **libgdbc_disconnect** ()
- int **libgdbc_continue** ()
- int **libgdbc_read_registers** ()
- int **libgdbc_read_memory** (uint64_t address, uint64_t len)
- int **libgdbc_send_cmd** (char *command)

4.3.1 Function Documentation

4.3.1.1 int libgdbc_init (uint8_t *architecture*)

Initializes the blah

4.4 /home/rene/libgdbc/include/messages.h File Reference

```
#include "core.h"
```

Functions

- int handle_g (libgdbc_t *instance)
- int **handle_m** (libgdbc_t *instance)

4.4.1 Function Documentation

4.4.1.1 int handle_g (libgdbc_t * *instance*)

See Appendix E in the gdb manual (GDB Remote Serial Protocol) Packets look following: $ starts a command/-packet, the end is indicated with # and a final checksum $<command>#<checksum>

4.5 /home/rene/libgdbc/include/packet.h File Reference

```
#include "core.h"
#include <stdint.h>
#include <unistd.h>
```

Classes

- struct parsing_object_t

Typedefs

- typedef struct parsing_object_t **parsing_object_t**

Functions

- int **parse_packet** (libgdbc_t *instance)
- int **push_message** (libgdbc_t *instance, parsing_object_t *parsed)
- char * **pop_message** (libgdbc_t *instance)
- void **handle_data** (parsing_object_t *current)
- void **handle_chk** (parsing_object_t *current)
- void **handle_packet** (parsing_object_t *current)
- void **handle_escape** (parsing_object_t *current)
- char **get_next_token** (parsing_object_t *current)

4.6 /home/rene/libgdbc/include/target.h File Reference

```
#include "libgdbc.h"
```

Classes

- struct libgdbc_x86_64_t

Typedefs

- typedef struct libgdbc_x86_64_t **libgdbc_x86_64_t**

4.7 /home/rene/libgdbc/include/utils.h File Reference

```
#include <stdint.h>
```

Functions

- uint8_t cmd_checksum (const char *command)
- uint64_t unpack_uint64 (char *buff, int len)
- uint64_t unpack_uint64_co (char *buff, int len)
- int hex2int (int ch)

4.7.1 Function Documentation

4.7.1.1 uint8_t cmd_checksum (const char * *command*)

Function creates the checksum for the given command

- command : is used to calculate the checksum needs to be null terminated

Returns

: calculated checksum

4.7.1.2 int hex2int (int *ch*)

Converts a given hex character into its int value

Returns

value of hex or -1 on error

4.7.1.3 uint64_t unpack_uint64 (char * *buff,* int *len*)

Converts str to uint64_t

4.7.1.4 uint64_t unpack_uint64_co (char * *buff,* int *len*)

Changed byte order and converts the value into uint64_t

Index