MPT Project

Generated by Doxygen 1.8.1.2

Sun Jul 14 2013 14:52:49

Contents

Chapter 1

Todo List

Global exec_cmd (char *cmd)

Check result of commands and return some error code.

Chapter 2

Module Index

2.1 Modules

Here is a list of all modules:

Chapter 3

Data Structure Index

3.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 4

File Index

4.1 File List

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

Chapter 5

Module Documentation

5.1 Parser Errors

Return values of parser()

Macros

- #define PARSE_OK 0

  Successfully parsed and parameters seems good.
- #define PARSE_ERR_NOCMD 1

  No such command.
- #define PARSE_ERR_REGEX 2

  Invalid regexp format specified in c source.
- #define PARSE_ERR_IP 3

  Wrong IP address entered.
- #define PARSE_ERR_MASK 4

  Wrong network mask entered.
- #define PARSE_HELP_INT 5

  Wrong parameter list after command interface.
- #define PARSE_HELP_ADDR 6

  Wrong parameter list after command address.

5.1.1 Detailed Description

Return values of parser()

5.2 ParserCmdType

Parsed command type.

Macros

- #define **MPT_INT_UPDOWN** 0
- #define **MPT_ADDR_ADDDEL** 1
- #define **MPT_INT** 2
- #define **MPT_ADDR** 3
- #define **MPT_RELOAD** 4
- #define **MPT_DELETE** 5
- #define **MPT_SAVE** 6

5.2.1 Detailed Description

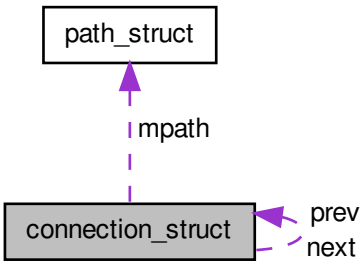
Parsed command type.

Chapter 6

Data Structure Documentation

6.1 connection_struct Struct Reference

Collaboration diagram for connection_struct:



Data Fields

- char name [128]

  The name of the connection.
- char filename [128]

  The configuraton file of the connection.
- bit_32 ip_local [4]

  The local IP address of the tunnel can be v4 or v6.
- bit_32 ip_remote [4]

  The remote IP address of the tunnel can be v4 or v6.
- bit_16 port_local

  The UDP local endpoint id (port number)
- bit_16 port_remote

  The UDP remote endpoint id (port number)
- bit_16 cmd_port_remote

  The cmd remote UDP port number.
- int socket

The local UDP socket ID of the conn. (IPv6 and IPv4 too)

- int socket_raw

  The local raw socket ID (if raw socket can be used)

- bit_32 path_packet

  The number of packets, sent on the actual path of the connection.

- bit_64 conn_packet

  The number of packets sent on the connection.

- bit_16 path_count

  The number of the existing paths.

- bit_16 path_index

  The index of the actual path in the mpath array.

- bit_8 status

  The status of the connection.

- bit_8 permission

  Allow connection updates (1: send bit, 2: receive bit)

- bit_8 ip_version

  The IP version of the connection (4 or 6)

- bit_8 keepalive

  Keepalive message interval in secs.

- bit_16 deadtimer

  Keepalive timeout value in secs.

- bit_32 waitrand

  Security token used at communication.

- bit_8 waitround

  Client phase in communication.

- char waithash [32]

  Expected checksum for receiving data (SHA-256 is 32 byte long)

- bit_8 auth_type

  The authentication code for the connection.

- char auth_key [128]

  The key value of the authentication.

- path_type mpath [MAX_PATH]

  The paths asociated to the connection.

- struct connection_struct * next

  Pointer to the next connection element.

- struct connection_struct * prev

  Pointer to the previous connection element.

- pthread_t socket_read

  The thread id of the connection socket reader thread.

6.1.1 Detailed Description

Definition at line 60 of file multipath.h.

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

- /home/kiskele/mptcp/src/include/multipath.h

6.2 interface_struct Struct Reference

Data Fields

- char name [128]

  The name of the interface.
- bit_32 ip4

  The IPv4 address.
- bit_8 ip4len

  The IPv4 prefix length.
- bit_32 ip6 [4]

  The IPv6 address.
- bit_8 ip6len

  The IPv6 prefix length.
- bit_32 mac_local [2]

  The MAC address.
- bit_32 ip4_gw

  The IPv4 address of the gateway.
- bit_32 ip6_gw [4]

  The IPv6 address of the gateway.
- bit_32 mac_gw [2]

  The MAC address of the gateway.

6.2.1 Detailed Description

Definition at line 110 of file multipath.h.

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

- /home/kiskele/mptcp/src/include/multipath.h

6.3 mpt_addr_adddel Struct Reference

```
#include <cli.h>
```

Data Fields

- char * op

  Operator can be add or del.
- char * ip

  IP Address.
- char * mask

  Network mask.
- char * dev

  Interface name.

6.3.1 Detailed Description

Structure to save arguments of command mpt address add|del

Definition at line 107 of file cli.h.

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

- /home/kiskele/mptcp/src/include/cli.h

6.4 mpt_int_updown Struct Reference

`#include <cli.h>`

Data Fields

- char * interface
  Interface name.
- char * mark
  Mark interface up or down.

6.4.1 Detailed Description

Structure to save arguments of command mpt interface up|down

Definition at line 99 of file cli.h.

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

- /home/kiskele/mptcp/src/include/cli.h

6.5 path_struct Struct Reference

Data Fields

- bit_32 ip_local [4]
  The local IP address of the path, can be v4 or v6.
- bit_32 ip_remote [4]
  The remote IP address of the path, can be v4 or v6.
- bit_32 ip_gw [4]
  The IP address of the outgoing gateway, can be v4 or v6.
- bit_32 mac_local [2]
  The MAC address of the local interface (for EUI-64 too)
- bit_32 mac_gw [2]
  The MAC address of the outgoing gateway (for EUI-64)
- bit_32 packet_max
  The maximum number of packets to send continously on the path.
- bit_32 header [32]
  buffer to hold the header (Eth, IP, UDP) data for raw socket
- bit_16 weight_in
  The weight value of the incoming traffic.
- bit_16 weight_out

The weight value of the outgoing traffic.

- bit_8 status

  The status of the path.
- bit_8 ip_version

  The IP version of the path.
- struct timeval last_keepalive

  Time of last keepalive received.
- struct sockaddr_in6 peer

  The sockaddr data for the peer.
- struct sockaddr_in6 peer_cmd

  The sockaddr data for the peer.
- char interface [128]

  The name of the local physical interface.

6.5.1 Detailed Description

Definition at line 40 of file multipath.h.

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

- /home/kiskele/mptcp/src/include/multipath.h

6.6 tunnel_struct Struct Reference

Data Fields

- char interface [128]

  The name of the tunnel interface e.g. tun1.
- char device [128]

  The name of the tunnel device.
- int fd

  The file descriptor of the opened tunnel interface.
- bit_32 ip4

  The IPv4 address of the tunnel.
- bit_16 ip4len

  The prefix length of the IPv4 address.
- bit_32 ip6 [4]

  The IPv6 address of the tunnel.
- bit_16 ip6len

  The pref.length of the IPv6 address.
- bit_16 cmd_port_rcv

  The port number for multipath commands to accept.
- int cmd_socket_rcv

  The socket id of the cmd communication to accept.
- bit_16 cmd_port_snd

  Port number to send multipath commands.
- int cmd_socket_snd

  Socket id of the cmd communication to start.
- pthread_t tunnel_read

  The thread id of the tunnel reader thread.
- pthread_t cmd_read

  The thread id of the cmd read thread.

6.6.1 Detailed Description

Definition at line 92 of file multipath.h.

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

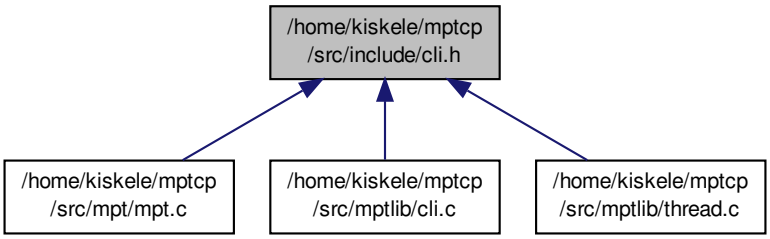
- /home/kiskele/mptcp/src/include/multipath.h

Chapter 7

File Documentation

7.1 /home/kiskele/mptcp/src/include/cli.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct mpt_int_updown
- struct mpt_addr_adddel

Macros

- #define strstart(src, pattern) strncmp(src, pattern, strlen(pattern))

  Src string begins with pattern.
- #define REGEX_IPV4 "([[:digit:]]{1,3})\\.([[:digit:]]{1,3})\\.([[:digit:]]{1,3})\\.([[:digit:]]{1,3})"

  Regexp pattern to match a part of IPv4 address.
- #define REGEX_IPV6 "[0-9a-fA-F:\\.]{1,4}"

  Regexp pattern to match a part of IPv6 address.
- #define PARSE_OK 0

  Successfully parsed and parameters seems good.
- #define PARSE_ERR_NOCMD 1

  No such command.
- #define PARSE_ERR_REGEX 2

  Invalid regexp format specified in c source.

- #define PARSE_ERR_IP 3

  Wrong IP address entered.
- #define PARSE_ERR_MASK 4

  Wrong network mask entered.
- #define PARSE_HELP_INT 5

  Wrong parameter list after command interface.
- #define PARSE_HELP_ADDR 6

  Wrong parameter list after command address.
- #define **MPT_INT_UPDOWN** 0
- #define **MPT_ADDR_ADDDEL** 1
- #define **MPT_INT** 2
- #define **MPT_ADDR** 3
- #define **MPT_RELOAD** 4
- #define **MPT_DELETE** 5
- #define **MPT_SAVE** 6
- #define HELP_INT "mpt int[erface] INTEFACE {up | down}\n\\n\INTEFACE: The name of the interface e.g. eth0\n\n"
- #define HELP_ADDR "mpt addr[ess] {add | del} IP_ADDRESS[/PREF_LEN] dev INTERFACE\n\\n\IP_ADDRESS: The IP address (can be v4 or v6) to manipulate\n\PREF_LEN: The prefix length of the manipulated address\n\ Default prefix length: 24 for IPv4, 64 for IPv6.\n\INTERFACE: The name of the interface related to the manipulated address\n\n"
- #define HELP_RELOAD "mpt reload [FILENAME]\n\\n\FILENAME: The file you have recently edited and needed to reload (Optional)\n\n"
- #define HELP_DELETE "mpt delete FILENAME\n\\n\FILENAME: The file that contains connections no longer needed\n\n"
- #define HELP_SAVE "mpt save [FILENAME]\n\\n\FILENAME: Save changed connection informations to config file (Optional)\n\n"
- #define HELP HELP_INT HELP_ADDR HELP_RELOAD HELP_DELETE HELP_SAVE

Functions

- int parse_cmd (char *cmd)
- int exec_cmd (char *cmd)

7.1.1 Detailed Description

Author

Kelemen Tamas kiskele@krc.hu

Definitions of cli.c

Definition in file cli.h.

7.1.2 Macro Definition Documentation

7.1.2.1 #define HELP HELP_INT HELP_ADDR HELP_RELOAD HELP_DELETE HELP_SAVE

Detailed help of all usable mpt commands

Definition at line 93 of file cli.h.

7.1.2.2 #define HELP_ADDR "mpt addr[ess] {add | del} IP_ADDRESS[/PREF_LEN] dev INTERFACE\n\n\IP_ADDRESS: The IP address (can be v4 or v6) to manipulate\n\PREF_LEN: The prefix length of the manipulated address\n\ Default prefix length: 24 for IPv4, 64 for IPv6.\n\INTERFACE: The name of the interface related to the manipulated address\n\n"

Help page of command mpt address

Definition at line 62 of file cli.h.

7.1.2.3 #define HELP_DELETE "mpt delete FILENAME\n\n\FILENAME: The file that contains connections no longer needed\n\n"

Help page of command mpt delete

Definition at line 79 of file cli.h.

7.1.2.4 #define HELP_INT "mpt int[erface] INTEFACE {up | down}\n\n\INTEFACE: The name of the interface e.g. eth0\n\n"

Help page of command mpt interface

Definition at line 55 of file cli.h.

7.1.2.5 #define HELP_RELOAD "mpt reload [FILENAME]\n\n\FILENAME: The file you have recently edited and needed to reload (Optional)\n\n"

Help page of command mpt reload

Definition at line 72 of file cli.h.

7.1.2.6 #define HELP_SAVE "mpt save [FILENAME]\n\n\FILENAME: Save changed connection informations to config file (Optional)\n\n"

Help page of command mpt save

Definition at line 86 of file cli.h.

7.1.3 Function Documentation

7.1.3.1 int exec_cmd (char * *cmd*)

Check CLI command syntax on server side and execute that

Todo Check result of commands and return some error code.

Parameters

cmd	CLI command received

Returns

Execution of command failed or succeeded

Definition at line 247 of file cli.c.

7.1.3.2 int parse_cmd (char * *cmd*)

Check CLI command syntax

Parameters

cmd	CLI command input

Returns

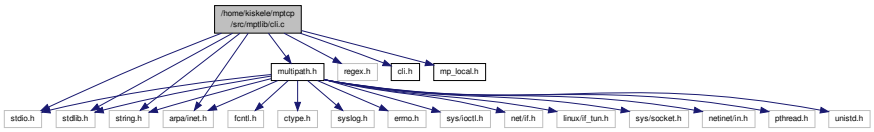
See Parser Errors

Definition at line 216 of file cli.c.

7.2 /home/kiskele/mptcp/src/mptlib/cli.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <regex.h>
#include <arpa/inet.h>
#include "cli.h"
#include "multipath.h"
#include "mp_local.h"
```

Include dependency graph for cli.c:



Functions

- void mpt_int_updown_constructor (struct mpt_int_updown **this)
- void mpt_int_updown_destructor (struct mpt_int_updown **this)
- void mpt_addr_adddel_constructor (struct mpt_addr_adddel **this)
- void mpt_addr_adddel_destructor (struct mpt_addr_adddel **this)
- void save_match (char **dst, char *src, regmatch_t pmatch)
- int parser (char *cmd, int *rettype, void **ret)
- int parse_cmd (char *cmd)
- int exec_cmd (char *cmd)

7.2.1 Detailed Description

Author

Kelemen Tamas kiskele@krc.hu

This file is used to check CLI commands and their parameters.

Definition in file cli.c.

7.2.2 Function Documentation

7.2.2.1 int exec_cmd (char * *cmd*)

Check CLI command syntax on server side and execute that

Todo Check result of commands and return some error code.

Parameters

cmd	CLI command received

Returns

Execution of command failed or succeeded

Definition at line 247 of file cli.c.

7.2.2.2 void mpt_addr_adddel_constructor (struct mpt_addr_adddel ** *this*)

Constructor that allocates memory and initializes the structure of "address add|del" command

Parameters

this	Pointer to an "mpt_addr_adddel *" variable

Definition at line 48 of file cli.c.

7.2.2.3 void mpt_addr_adddel_destructor (struct mpt_addr_adddel ** *this*)

Destructor that frees up memory after struct mpt_addr_adddel * is no longer needed

Parameters

this	Pointer to an "mpt_addr_adddel *" variable

Definition at line 61 of file cli.c.

7.2.2.4 void mpt_int_updown_constructor (struct mpt_int_updown ** *this*)

Constructor that allocates memory and initializes the structure of "interface up|down" command

Parameters

this	Pointer to an "mpt_int_updown *" variable

Definition at line 26 of file cli.c.

7.2.2.5 void mpt_int_updown_destructor (struct mpt_int_updown ** *this*)

Destructor that frees up memory after struct mpt_int_updown * is no longer needed

Parameters

this	Pointer to an "mpt_int_updown *" variable

Definition at line 37 of file cli.c.

7.2.2.6 int parse_cmd (char * *cmd*)

Check CLI command syntax

Parameters

cmd	CLI command input

Returns

See Parser Errors

Definition at line 216 of file cli.c.

7.2.2.7 int parser (char * *cmd*, int * *rettype*, void ** *ret*)

Function to parse CLI input and return command type and structure of parameters

Parameters

cmd	Input string which need to parse
rettype	Return value of the command type. See ParserCmdType
ret	Return a struct for that command that contains the parsed arguments

Returns

Parse or parameter errors. See Parser Errors for return values

Definition at line 98 of file cli.c.

7.2.2.8 void save_match (char ** *dst*, char * *src*, regmatch_t *pmatch*)

Function that creates string from regular expression match

Parameters

dst	Pointer to the newly allocated destination string
src	Original string which contains the matching string
pmatch	Which match need to be saved. N-th element of array return by regexec This contains index of the beginning and the end of the match

Definition at line 79 of file cli.c.

Index