UDP

Generated by Doxygen 1.16.1

Chapter 1

udp

Chapter 2

Topic Index

2.1 Topics

Here is a list of all topics with brief descriptions:

Chapter 3

Directory Hierarchy

3.1 Directories

Chapter 4

Class Index

4.1 Class List

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

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

Chapter 6

Topic Documentation

6.1 work for udp

Group function for work udp package.

Classes

- struct udp_head

  Struct is header UDP pack.
- struct udp_pack

  Struct is UDP package.
- struct pseudo_header

  Struct pseudo_header for calculate checksum for UDP package.

Macros

- #define PACKED __attribute__((packed))

Typedefs

- typedef struct udp_pack * udp_pack_t

  UDP packet descriptor.

Functions

- static uint16_t inet_port (const char *const port)

  Function support for parsing port.
- static void set_size_udp_pack (udp_pack_t pack, const uint16_t size)

  Function raw write size UDP package.
- static uint32_t sum_compute (void *ptr, uint16_t nbytes)

  Function calculate sum in big endian.
- static uint16_t checksum_compute (uint32_t sum)

  Function calculating from sum big endian to checksum big endian.

- static void calculate_checksum_udp_pack (udp_pack_t pack)

  Function calculate checksum for ip header and UDP package.
- static void * get_pack_udp_pack (udp_pack_t pack)

  Function raw get pointer on UDP package.
- static void * inet_mac (const char *const mac_address)

  Function return mac address in big endian.
- udp_pack_t init_udp_pack (void)

  Function for create object UDP package.
- ssize_t set_port_source_udp_pack (udp_pack_t pack, const char *const port)

  Function for setting source port in UDP package.
- ssize_t set_port_destantion_udp_pack (udp_pack_t pack, const char *const port)

  Function for setting destantion port in UDP package.
- ssize_t set_ip_address_source_udp_pack (udp_pack_t pack, const char *const ip)

  Function for setting source ip address in UDP package.
- ssize_t set_ip_address_destantion_udp_pack (udp_pack_t pack, const char *const ip)

  Function for setting destantion ip address in UDP package.
- ssize_t add_data_udp_pack (udp_pack_t pack, void *data, uint16_t size)

  Function addition data in UDP package.
- ssize_t set_data_udp_pack (udp_pack_t pack, void *data, const uint16_t size)

  Function overriding data in UDP package.
- ssize_t add_byte_udp_pack (udp_pack_t pack, const uint8_t byte)

  Function addition byte in UDP package.
- ssize_t set_input_data_udp_pack (udp_pack_t pack)

  Function read data from stdin in UDP package.
- ssize_t set_file_data_udp_pack (udp_pack_t pack, const char *const file_name)

  Function read data from file in UDP package.
- ssize_t set_interface_udp_pack (udp_pack_t pack, const char *const interface)

  Function for pick interface to send UDP package.
- ssize_t send_udp_pack (udp_pack_t pack)

  Function to send UDP package.
- char * get_mac_address_source_udp_pack (udp_pack_t pack)

  Function getting mac address for source.
- char * get_mac_address_destantion_udp_pack (udp_pack_t pack)

  Function setting mac address for destantion.
- ssize_t set_mac_address_source_udp_pack (udp_pack_t pack, const char *const mac_address)

  Function setting mac address for source.
- ssize_t set_mac_address_destantion_udp_pack (udp_pack_t pack, const char *const mac_address)

  Function setting mac address for destantion.
- char * get_interface_udp_pack (udp_pack_t pack)

  Function for getting interface.
- char * get_port_source_udp_pack (udp_pack_t pack)

  Function for getting port source.
- char * get_port_destantion_udp_pack (udp_pack_t pack)

  Function for getting port destantion.
- char * get_ip_address_source_udp_pack (udp_pack_t pack)

  Function for getting ip address source.
- char * get_ip_address_destantion_udp_pack (udp_pack_t pack)

  Function for getting ip address destantion.
- uint16_t get_size_data_udp_pack (udp_pack_t pack)

  Function for getting size data.
- char * get_data_udp_pack (udp_pack_t pack)

Function for getting data.

- char * get_data_hex_udp_pack (udp_pack_t pack)

  Function for getting data hex.

- ssize_t print_udp_pack (udp_pack_t pack)

  Function print all info about udp pack.

- void destroy_udp_pack (udp_pack_t pack)

  Function free UDP package.

Variables

- struct udp_head PACKED

6.1.1 Detailed Description

Group function for work udp package.

6.1.2 Macro Definition Documentation

6.1.2.1 PACKED

```
struct pseudo_header PACKED __attribute__((packed))
```

6.1.3 Typedef Documentation

6.1.3.1 udp_pack_t

```
typedef struct udp_pack* udp_pack_t
```

UDP packet descriptor.

The primary type for working with the library. All functions accept this pointer as an argument.

6.1.4 Function Documentation

6.1.4.1 add_byte_udp_pack()

```
ssize_t add_byte_udp_pack (
            udp_pack_t pack,
            const uint8_t byte)
```

Function addition byte in UDP package.

Note

You must call init_udp_pack before this.

Parameters

in,out	*pack*	UDP package for work.

in	*byte*	Addition byte in UDP package.

Returns

0 or -1 on error. Usage example.

```
ssize_t ret = 0;
udp_pack_t pack = init_udp_pack();
if (pack == NULL) {
    ret = -1;
    goto get_not_udp_pack;
}
ret = add_byte_udp_pack(pack, 'n');
if (ret)
    goto add_not_byte;
ret = add_byte_udp_pack(pack, '\0');
    goto add_not_byte;
add_not_byte:
get_not_udp_pack:
destroy_udp_pack(pack);
```

6.1.4.2 add_data_udp_pack()

```
ssize_t add_data_udp_pack (
            udp_pack_t pack,
            void * data,
            uint16_t size)
```

Function addition data in UDP package.

Note

You must call init_udp_pack before this.

Parameters

in,out	*pack*	UDP package for work.
in	*data*	Addition data in UDP package.
in	*size*	Size addition data for UDP package.

Returns

0 or -1 on error. Usage example.

```
ssize_t ret = 0;
const char * const message = "Hello, world";
udp_pack_t pack = init_udp_pack();
if (pack == NULL) {
    ret = -1;
    goto get_not_udp_pack;
}
ret = add_data_udp_pack(pack, message, strlen(message));
if (ret == -1)
    gotov add_not_data;
add_not_data:
get_not_udp_pack:
destroy_udp_pack(pack);
```

6.1.4.3 calculate_checksum_udp_pack()

```
void calculate_checksum_udp_pack (
            udp_pack_t pack)  [static]
```

Function calculate checksum for ip header and UDP package.

Parameters

in,out	*pack*	UDP package for work.

Note

This function is private. Not used outside udp_lib/udp.c

6.1.4.4 checksum_compute()

```
uint16_t checksum_compute (
            uint32_t sum)  [static]
```

Function calculating from sum big endian to checksum big endian.

Parameters

in	*sum*	Sum in big endian.

Returns

Checksum in big endian.

Note

This function is private. Not used outside udp_lib/udp.c

6.1.4.5 destroy_udp_pack()

```
void destroy_udp_pack (
            udp_pack_t pack)
```

Function free UDP package.

Note

You must call init_udp_pack before this.

Parameters

in,out	*pack*	UDP package for work. Usage example.
		`ssize_t ret = 0;`
		`udp_pack_t pack = init_udp_pack();`
		`if (pack == NULL) {`
		`  ret = -1;`
		`  goto get_not_udp_pack;`
		`}`
		`// other code whit using udp_pack_t`
		`get_not_udp_pack:`
		`destroy_udp_pack(pack);`

6.1.4.6 get_data_hex_udp_pack()

```
char * get_data_hex_udp_pack (
            udp_pack_t pack)
```

Function for getting data hex.

Note

You must call init_udp_pack before this.

Parameters

in,out	*pack*	UDP package for work.

Returns

Data hex in UDP package or NULL pointer on error. Usage example.

```
ssize_t ret = 0;
udp_pack_t pack = init_udp_pack();
if (pack == NULL) {
    ret = -1;
    goto get_not_udp_pack;
}
{
    char * data = get_data_hex_udp_pack(pack);
    if (data == NULL) {
        ret = -1;
        goto get_not_data_hex_udp_pack;
    }
    // other code whit udp_pack_t
}
// other code whit udp_pack_t
get_not_data_hex_udp_pack:
get_not_udp_pack:
destroy_udp_pack(pack);
```

6.1.4.7 get_data_udp_pack()

```
char * get_data_udp_pack (
            udp_pack_t pack)
```

Function for getting data.

Note

You must call init_udp_pack before this.

Parameters

in,out	*pack*	UDP package for work.

Returns

Data in UDP package. Usage example.

```
ssize_t ret = 0;
udp_pack_t pack = init_udp_pack();
if (pack == NULL) {
    ret = -1;
    goto get_not_udp_pack;
}
{
    void * data = get_data_udp_pack(pack);
    // other code whit udp_pack_t
}
// other code whit udp_pack_t
get_not_udp_pack:
destroy_udp_pack(pack);
```

6.1.4.8 get_interface_udp_pack()

```
char * get_interface_udp_pack (
            udp_pack_t pack)
```

Function for getting interface.

Note

You must call init_udp_pack before this.

You can call set_interface_udp_pack before this.

Parameters

in,out	*pack*	UDP package for work.

Returns

String or NULL pointer is fail. Usage example.

```
ssize_t ret = 0;
udp_pack_t pack = init_udp_pack();
if (pack == NULL) {
    ret = -1;
    goto get_not_udp_pack;
}
ret = set_interface_udp_pack(pack, "lo");
if (ret == -1)
    goto setting_not_interface;
{
    char * interface = NULL;
    interface = get_interface_udp_pack(pack);
    if (interface == NULL)
        getting_not_interface;
    // other code whit udp_pack_t
}
// other code whit udp_pack_t
getting_not_interface:
setting_not_interface:
get_not_udp_pack:
destroy_udp_pack(pack);
```

6.1.4.9 get_ip_address_destantion_udp_pack()

```
char * get_ip_address_destantion_udp_pack (
            udp_pack_t pack)
```

Function for getting ip address destantion.

Note

You must call init_udp_pack before this.

You can call set_ip_address_destantion_udp_pack before this.

Parameters

in,out	*pack*	UDP package for work.

Returns

String or NULL pointer is fail. Usage example.

```
ssize_t ret = 0;
udp_pack_t pack = init_udp_pack();
if (pack == NULL) {
    ret = -1;
    goto get_not_udp_pack;
}
ret = set_ip_address_source_udp_pack(pack, "127.0.0.1");
if (ret == -1)
    goto setting_not_ip_destantion;
{
    char * ip_address = NULL;
    ip_address = get_ip_destantion_udp_pack(pack);
    if (ip_address == NULL)
        getting_not_ip_address;
    // other code whit udp_pack_t
}
// other code whit udp_pack_t
getting_not_ip_address:
setting_not_ip_address:
get_not_udp_pack:
destroy_udp_pack(pack);
```

6.1.4.10 get_ip_address_source_udp_pack()

```
char * get_ip_address_source_udp_pack (
            udp_pack_t pack)
```

Function for getting ip address source.

Note

You must call init_udp_pack before this.

You can call set_ip_address_source_udp_pack before this.

Parameters

in,out	*pack*	UDP package for work.

Returns

String or NULL pointer is fail. Usage example.

```
ssize_t ret = 0;
udp_pack_t pack = init_udp_pack();
if (pack == NULL) {
    ret = -1;
    goto get_not_udp_pack;
}
ret = set_ip_address_source_udp_pack(pack, "127.0.0.1");
if (ret == -1)
    goto setting_not_ip_address;
{
    char * ip_address = NULL;
    ip_address = get_ip_address_source_udp_pack(pack);
    if (ip_source == NULL)
        getting_not_ip_address;
    // other code whit udp_pack_t
}
// other code whit udp_pack_t
getting_not_ip_address:
setting_not_ip_address:
get_not_udp_pack:
destroy_udp_pack(pack);
```

6.1.4.11 get_mac_address_destantion_udp_pack()

```
char * get_mac_address_destantion_udp_pack (
            udp_pack_t pack)
```

Function setting mac address for destantion.

Note

You must call init_udp_pack before this.

You can call set_mac_address_destantion_udp_pack before this.

Parameters

in,out	*pack*	UDP package for work.

Returns

String or NULL pointer is fail. Usage example.

```
ssize_t ret = 0;
udp_pack_t pack = init_udp_pack();
if (pack == NULL) {
    ret = -1;
    goto get_not_udp_pack;
}
ret = set_mac_address_destantion_udp_pack(pack, "ff:ff:ff:ff:ff:ff");
if (ret == -1)
    goto setting_not_mac_address;
{
    char * mac_address = NULL;
    mac_address = get_mac_address_destantion_udp_pack(udp_pack pack);
    // other code whit udp_pack_t
}
// other code whit udp_pack_t
setting_not_mac_address:
get_not_udp_pack:
destroy_udp_pack(pack);
```

6.1.4.12 get_mac_address_source_udp_pack()

```
char * get_mac_address_source_udp_pack (
            udp_pack_t pack)
```

Function getting mac address for source.

Note

You must call init_udp_pack before this.

You can call set_mac_address_source_udp_pack before this.

Parameters

in,out	*pack*	UDP package for work.

Returns

String or NULL pointer is fail. Usage example.

```
ssize_t ret = 0;
udp_pack_t pack = init_udp_pack();
if (pack == NULL) {
    ret = -1;
    goto get_not_udp_pack;
}
ret = set_mac_address_source_udp_pack(pack, "ff:ff:ff:ff:ff:ff");
if (ret == -1)
    goto setting_not_mac_address;
{
    char * mac_address =  NULL;
    mac_address = get_mac_address_source_udp_pack(pack);
    // other code whit udp_pack_t
}
// other code whit udp_pack_t
setting_not_mac_address:
get_not_udp_pack:
destroy_udp_pack(pack);
```

6.1.4.13 get_pack_udp_pack()

```
void * get_pack_udp_pack (
            udp_pack_t pack)  [static]
```

Function raw get pointer on UDP package.

Parameters

in,out	*pack*	UDP package for work.

Returns

Getting UDP package.

Note

This function is private. Not used outside udp_lib/udp.c

6.1.4.14 get_port_destantion_udp_pack()

```
char * get_port_destantion_udp_pack (
            udp_pack_t pack)
```

Function for getting port destantion.

Note

You must call init_udp_pack before this.

You can call set_port_destantion_udp_pack before this.

Parameters

in,out	*pack*	UDP package for work.

Returns

String or NULL pointer is fail. Usage example.

```
ssize_t ret = 0;
udp_pack_t pack = init_udp_pack();
if (pack == NULL) {
    ret = -1;
    goto get_not_udp_pack;
}
ret = set_port_destantion_udp_pack(pack, "8001");
if (ret == -1)
    goto setting_not_port_destantion;
{
    char * port_destantion = NULL;
    port_destantion = get_port_destantion_udp_pack(pack);
    if (port_destantion == NULL)
        getting_not_port_destantion;
    // other code whit udp_pack_t
}
// other code whit udp_pack_t
getting_not_port_destantion:
setting_not_port_destantion:
get_not_udp_pack:
destroy_udp_pack(pack);
```

6.1.4.15 get_port_source_udp_pack()

```
char * get_port_source_udp_pack (
            udp_pack_t pack)
```

Function for getting port source.

Note

You must call init_udp_pack before this.

You can call set_port_source_udp_pack before this.

Parameters

in,out	*pack*	UDP package for work.

Returns

String or NULL pointer is fail. Usage example.

```
ssize_t ret = 0;
udp_pack_t pack = init_udp_pack();
if (pack == NULL) {
    ret = -1;
    goto get_not_udp_pack;
}
ret = set_port_source_udp_pack(pack, "8003");
if (ret == -1)
    goto setting_not_port_source;
{
    char * port_source = NULL;
    port_source = get_port_source_udp_pack(pack);
    if (port_source == NULL)
        getting_not_port_source;
    // other code whit udp_pack_t
}
// other code whit udp_pack_t
getting_not_port_source:
setting_not_port_source:
get_not_udp_pack:
destroy_udp_pack(pack);
```

6.1.4.16 get_size_data_udp_pack()

```
uint16_t get_size_data_udp_pack (
            udp_pack_t pack)
```

Function for getting size data.

Note

You must call init_udp_pack before this.

You can call set_data_udp_pack before this.

You can call add_data_udp_pack before this.

You can call add_byte_udp_pack before this.

Parameters

in,out	*pack*	UDP package for work.

Returns

Length data in UDP package. Usage example.

```
ssize_t ret = 0;
udp_pack_t pack = init_udp_pack();
if (pack == NULL) {
    ret = -1;
    goto get_not_udp_pack;
}
{
    uint16_t size_data = get_size_data_udp_pack(pack);
    // other code whit udp_pack_t
}
// other code whit udp_pack_t
getting_not_size_data:
get_not_udp_pack:
destroy_udp_pack(pack);
```

6.1.4.17 inet_mac()

```
void * inet_mac (
            const char *const mac_address)  [static]
```

Function return mac address in big endian.

Parameters

in	*mac_address*	String mac addres.

Returns

Getting buffer mac address.

Note

This function is private. Not used outside udp_lib/udp.c

6.1.4.18 inet_port()

```
uint16_t inet_port (
            const char *const port)  [static]
```

Function support for parsing port.

Parameters

in	*port*	Port in string format.

Returns

Port for big indian architect.

Note

This function is private. Not used outside udp_lib/udp.c

6.1.4.19 init_udp_pack()

```
udp_pack_t init_udp_pack (
            void )
```

Function for create object UDP package.

Note

You must call destroy_udp_pack after this.

Returns

pointer or null on error. Usage example.

```
ssize_t ret = 0;
udp_pack_t pack = init_udp_pack();
if (pack == NULL) {
    ret = -1;
    goto get_not_udp_pack;
}
// other code whit using udp_pack_t
get_not_udp_pack:
destroy_udp_pack(pack);
```

6.1.4.20 print_udp_pack()

```
ssize_t print_udp_pack (
            udp_pack_t pack)
```

Function print all info about udp pack.

Note

You must call init_udp_pack before this.

Parameters

in,out	*pack*	UDP package for work.

Returns

0 or -1 on error. Usage example.

```
ssize_t ret = 0;
udp_pack_t pack = init_udp_pack();
if (pack == NULL) {
    ret = -1;
    goto get_not_udp_pack;
}
ret = print_udp_pack(pack);
if (ret)
    goto print_not_udp_pack;
// other code whit udp_pack_t
print_not_udp_pack:
get_not_udp_pack:
destroy_udp_pack(pack);
```

6.1.4.21 send_udp_pack()

```
ssize_t send_udp_pack (
            udp_pack_t pack)
```

Function to send UDP package.

Note

You must call init_udp_pack before this.

Parameters

in,out	*pack*	UDP package for work.

Returns

0 or -1 on error. Usage example.

```
ssize_t ret = 0;
udp_pack_t pack = init_udp_pack();
if (pack == NULL) {
    ret = -1;
    goto get_not_udp_pack;
}
send_udp_pack(pack);
get_not_udp_pack:
destroy_udp_pack(pack);
```

6.1.4.22 set_data_udp_pack()

```
ssize_t set_data_udp_pack (
            udp_pack_t pack,
            void * data,
            const uint16_t size)
```

Function overriding data in UDP package.

Note

You must call init_udp_pack before this.

Parameters

in,out	*pack*	UDP package for work.
in	*data*	New data in UDP package.
in	*size*	Size new data for UDP package.

Returns

0 or -1 on error. Usage example.

```
ssize_t ret = 0;
const char * const message = "Hello, world";
udp_pack_t pack = init_udp_pack();
if (pack == NULL) {
    ret = -1;
    goto get_not_udp_pack;
}
set_data_udp_pack(pack, message, strlen(message));
get_not_udp_pack:
destroy_udp_pack(pack);
```

6.1.4.23 set_file_data_udp_pack()

```
ssize_t set_file_data_udp_pack (
            udp_pack_t pack,
            const char *const file_name)
```

Function read data from file in UDP package.

Note

You must call init_udp_pack before this.

Parameters

in,out	*pack*	UDP package for work.
in	*file_name*	File name for read in UDP pack.

Returns

0 or -1 on error. Usage example.

```
ssize_t ret = 0;
udp_pack_t pack = init_udp_pack();
if (pack == NULL) {
    ret = -1;
    goto get_not_udp_pack;
}
set_file_data_udp_pack(pack, "Hello.txt");
get_not_udp_pack:
destroy_udp_pack(pack);
```

6.1.4.24 set_input_data_udp_pack()

```
ssize_t set_input_data_udp_pack (
            udp_pack_t pack)
```

Function read data from stdin in UDP package.

Note

You must call init_udp_pack before this.

Parameters

in,out	*pack*	UDP package for work.

Returns

0 or -1 on error. Usage example.

```
ssize_t ret = 0;
udp_pack_t pack = init_udp_pack();
if (pack == NULL) {
    ret = -1;
    goto get_not_udp_pack;
}
set_input_data_udp_pack(pack);
get_not_udp_pack:
destroy_udp_pack(pack);
```

6.1.4.25 set_interface_udp_pack()

```
ssize_t set_interface_udp_pack (
            udp_pack_t pack,
            const char *const interface)
```

Function for pick interface to send UDP package.

Note

You must call init_udp_pack before this.

Parameters

in,out	*pack*	UDP package for work.
in	*interface*	Interface to send UDP pack.

Returns

0 or -1 on error. Usage example.

```
ssize_t ret = 0;
udp_pack_t pack = init_udp_pack();
if (pack == NULL) {
    ret = -1;
    goto get_not_udp_pack;
}
set_interface_udp_pack(pack, "lo");
get_not_udp_pack:
destroy_udp_pack(pack);
```

6.1.4.26 set_ip_address_destantion_udp_pack()

```
ssize_t set_ip_address_destantion_udp_pack (
            udp_pack_t pack,
            const char *const ip)
```

Function for setting destantion ip address in UDP package.

Note

You must call init_udp_pack before this.

Parameters

in,out	*pack*	UDP package for work.
in	*ip*	Destination ip address to send.

Returns

0 or -1 on error. Usage example.

```
ssize_t ret = 0;
udp_pack_t pack = init_udp_pack();
if (pack == NULL) {
    ret = -1;
    goto get_not_udp_pack;
}
ret = set_ip_address_destantion_udp_pack(pack, "127.0.0.1");
if (ret == -1)
    goto setting_not_ip_address;
// other code whit udp_pack_t
setting_not_ip_address:
get_not_udp_pack:
destroy_udp_pack(pack);
```

6.1.4.27 set_ip_address_source_udp_pack()

```
ssize_t set_ip_address_source_udp_pack (
            udp_pack_t pack,
            const char *const ip)
```

Function for setting source ip address in UDP package.

Note

You must call init_udp_pack before this.

Parameters

in,out	*pack*	UDP package for work.
in	*ip*	Source ip to send.

Returns

0 or -1 on error. Usage example.

```
ssize_t ret = 0;
udp_pack_t pack = init_udp_pack();
if (pack == NULL) {
    ret = -1;
    goto get_not_udp_pack;
}
ret = set_ip_address_source_udp_pack(pack, "127.0.0.1");
if (ret == -1)
    goto set_not_ip_address;
// other code whit udp_pack_t
set_not_ip_address:
get_not_udp_pack:
destroy_udp_pack(pack);
```

6.1.4.28 set_mac_address_destantion_udp_pack()

```
ssize_t set_mac_address_destantion_udp_pack (
            udp_pack_t pack,
            const char *const mac_address)
```

Function setting mac address for destantion.

Note

You must call init_udp_pack before this.

Parameters

in,out	*pack*	UDP package for work.
in	*mac_address*	String is mac address destantion.

Returns

0 or -1 on error. Usage example.

```
ssize_t ret = 0;
udp_pack_t pack = init_udp_pack();
if (pack == NULL) {
    ret = -1;
    goto get_not_udp_pack;
}
ret = set_mac_address_destantion_udp_pack(pack, "ff:ff:ff:ff:ff:ff");
if (ret == -1)
    goto setting_not_mac_address;
// other code whit udp_pack_t
setting_not_mac_address:
get_not_udp_pack:
destroy_udp_pack(pack);
```

6.1.4.29 set_mac_address_source_udp_pack()

```
ssize_t set_mac_address_source_udp_pack (
            udp_pack_t pack,
            const char *const mac_address)
```

Function setting mac address for source.

Note

You must call init_udp_pack before this.

Parameters

in,out	*pack*	UDP package for work.

in	*mac_address*	String is mac address source.

Returns

0 or -1 on error. Usage example.

```
ssize_t ret = 0;
udp_pack_t pack = init_udp_pack();
if (pack == NULL) {
    ret = -1;
    goto get_not_udp_pack;
}
ret = set_mac_address_source_udp_pack(pack, "ff:ff:ff:ff:ff:ff");
if (ret == -1)
    goto setting_not_mac_address;
// other code whit udp_pack_t
setting_not_mac_address:
get_not_udp_pack:
destroy_udp_pack(pack);
```

6.1.4.30 set_port_destantion_udp_pack()

```
ssize_t set_port_destantion_udp_pack (
            udp_pack_t pack,
            const char *const port)
```

Function for setting destantion port in UDP package.

Note

You must call init_udp_pack before this.

Parameters

in,out	*pack*	UDP package for work.
in	*port*	Destination port to send.

Returns

0 or -1 on error. Usage example.

```
ssize_t ret = 0;
udp_pack_t pack = init_udp_pack();
if (pack == NULL) {
    ret = -1;
    goto get_not_udp_pack;
}
set_port_destantion_udp_pack(pack, "8003");
get_not_udp_pack:
destroy_udp_pack(pack);
```

6.1.4.31 set_port_source_udp_pack()

```
ssize_t set_port_source_udp_pack (
            udp_pack_t pack,
            const char *const port)
```

Function for setting source port in UDP package.

Note

You must call init_udp_pack before this.

Parameters

in,out	*pack*	UDP package for work.

in	*port*	Source port to send.

Returns

0 or -1 on error. Usage example.

```
ssize_t ret = 0;
udp_pack_t pack = init_udp_pack();
if (pack == NULL) {
    ret = -1;
    goto get_not_udp_pack;
}
set_port_source_udp_pack(pack, "8001");
get_not_udp_pack:
destroy_udp_pack(pack);
```

6.1.4.32 set_size_udp_pack()

```
void set_size_udp_pack (
            udp_pack_t pack,
            const uint16_t size)  [static]
```

Function raw write size UDP package.

Parameters

in,out	*pack*	UDP package for work.
in	*size*	New size UDP package.

Note

This function is private. Not used outside udp_lib/udp.c

6.1.4.33 sum_compute()

```
uint32_t sum_compute (
            void * ptr,
            uint16_t nbytes)  [static]
```

Function calculate sum in big endian.

Parameters

in	*ptr*	Buffer for calculating sum in big endian.
in	*nbytes*	Size buffer for calculating sum.

Returns

Sum buffer in big endian.

Note

This function is private. Not used outside udp_lib/udp.c

6.1.5 Variable Documentation

6.1.5.1 PACKED

```
struct pseudo_header PACKED
```

Chapter 7

Directory Documentation

7.1 udp_lib Directory Reference

Files

- file udp.c

  Code file for work udp package.
- file udp.h

  Header file for work udp package.

Chapter 8

Class Documentation

8.1 pseudo_header Struct Reference

Struct pseudo_header for calculate checksum for UDP package.

Public Attributes

- uint32_t source_address
- uint32_t dest_address
- uint8_t placeholder
- uint8_t protocol
- uint16_t udp_length

8.1.1 Detailed Description

Struct pseudo_header for calculate checksum for UDP package.

Note

This struct is private. Not used outside udp_lib/udp.c

8.1.2 Member Data Documentation

8.1.2.1 dest_address

```
uint32_t pseudo_header::dest_address
```

Destination ip addresses

8.1.2.2 placeholder

```
uint8_t pseudo_header::placeholder
```

Placeholder is 0x00.

8.1.2.3 protocol

```
uint8_t pseudo_header::protocol
```

Constant protocol.

8.1.2.4 source_address

```
uint32_t pseudo_header::source_address
```

Source ip addresses

8.1.2.5 udp_length

```
uint16_t pseudo_header::udp_length
```

Length UDP package.

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

- udp_lib/udp.c

8.2 udp_head Struct Reference

Struct is header UDP pack.

Public Attributes

- uint16_t m_port_source
- uint16_t m_port_destantion
- uint16_t m_length
- uint16_t m_checksum

8.2.1 Detailed Description

Struct is header UDP pack.

Note

This struct is private. Not used outside udp_lib/udp.c

8.2.2 Member Data Documentation

8.2.2.1 m_checksum

```
uint16_t udp_head::m_checksum
```

Calculated software checksum

8.2.2.2 m_length

```
uint16_t udp_head::m_length
```

Length udp pack

8.2.2.3 m_port_destantion

```
uint16_t udp_head::m_port_destantion
```

Port destantion

8.2.2.4 m_port_source

```
uint16_t udp_head::m_port_source
```

Port source

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

- udp_lib/udp.c

8.3 udp_pack Struct Reference

Struct is UDP package.

Public Attributes

- char m_interface [IFNAMSIZ]
- struct ethhdr m_ethhdr
- struct iphdr m_iphdr
- struct udp_head m_head
- uint8_t m_data [MAX_SIZE_DATA]

8.3.1 Detailed Description

Struct is UDP package.

Note

This struct is private. Not used outside udp_lib/udp.c

8.3.2 Member Data Documentation

8.3.2.1 m_data

```
uint8_t udp_pack::m_data[MAX_SIZE_DATA]
```

Data in UDP package.

8.3.2.2 m_ethhdr

```
struct ethhdr udp_pack::m_ethhdr
```

Ethernet header start UDP package.

8.3.2.3 m_head

```
struct udp_head udp_pack::m_head
```

UDP header

8.3.2.4 m_interface

```
char udp_pack::m_interface[IFNAMSIZ]
```

Name ethrnet interface.

8.3.2.5 m_iphdr

```
struct iphdr udp_pack::m_iphdr
```

IP header

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

- udp_lib/udp.c

Chapter 9

File Documentation

9.1 main.c File Reference

```
#include "udp_lib/udp.h"
#include <getopt.h>
#include <stddef.h>
#include <string.h>
```

Functions

- int main (int argc, char **argv)

  Entry point for the UDP packet crafting and transmission utility.

9.1.1 Function Documentation

9.1.1.1 main()

```
int main (
            int argc,
            char ** argv)
```

Entry point for the UDP packet crafting and transmission utility.

This program parses command-line arguments to build a custom UDP packet. It allows configuration of MAC addresses, IP addresses, ports, and payload data.

Command-line Options:

- `-w`, `--stdio` Read payload data from standard input.
- `-e`, `--print` Print the packet structure to the console before sending.
- `-i`, `--ip-address-destination` Set the destination IPv4 address.
- `-s`, `--ip-address-source` Set the source IPv4 address.

- `-p`, `--port-destanition` Set the destination UDP port.
- `-o`, `--port-source` Set the source UDP port.
- `-n`, `--interface` Specify the network interface (e.g., eth0).
- `-f`, `--file` Read payload data from a specified file.
- `-m`, `--mac-address-destantion` Set the destination MAC address.
- `-a`, `--mac-address-source` Set the source MAC address.

Payload Logic:

1. If `-w` or `-f` is provided, the data is pulled from those sources.
2. If no source flag is provided, the program concatenates all remaining positional arguments (argv) into a single space-separated string payload.

Parameters

argc	The number of command-line arguments.
argv	The array of command-line argument strings.

Returns

- `0` on successful transmission.
- `-1` if memory allocation fails (`init_udp_pack`).
- Non-zero error code if packet configuration or sending fails.

9.2 README.md File Reference

9.3 udp_lib/udp.c File Reference

Code file for work udp package.

```
#include "udp_lib/udp.h"
#include <stdint.h>
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <sys/ioctl.h>
#include <sys/stat.h>
#include <unistd.h>
#include <fcntl.h>
#include <errno.h>
#include <stddef.h>
#include <net/if.h>
#include <linux/if_packet.h>
#include <net/ethernet.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#include <netinet/ip.h>
```

Classes

- struct udp_head

  Struct is header UDP pack.
- struct udp_pack

  Struct is UDP package.
- struct pseudo_header

  Struct pseudo_header for calculate checksum for UDP package.

Macros

- #define PACKED __attribute__((packed))
- #define HEAD_ETH sizeof(struct ethhdr)
- #define HEAD_UDP sizeof(struct udp_head)
- #define HEAD_IP sizeof(struct iphdr)
- #define HEAD_UDP_IP (HEAD_IP + HEAD_UDP)
- #define MAX_SIZE_DATA (0xFFFF - HEAD_UDP_IP)
- #define NULL_CHECKSUM 0x0000
- #define MIN(left, rigth)
- #define HEAD_PSEUDO sizeof(struct pseudo_header)

Functions

- udp_pack_t init_udp_pack (void)

  Function for create object UDP package.
- static uint16_t inet_port (const char *const port)

  Function support for parsing port.
- ssize_t set_port_source_udp_pack (udp_pack_t pack, const char *const port)

  Function for setting source port in UDP package.
- ssize_t set_port_destantion_udp_pack (udp_pack_t pack, const char *const port)

  Function for setting destantion port in UDP package.
- ssize_t set_ip_address_source_udp_pack (udp_pack_t pack, const char *const ip)

  Function for setting source ip address in UDP package.
- ssize_t set_ip_address_destantion_udp_pack (udp_pack_t pack, const char *const ip)

  Function for setting destantion ip address in UDP package.
- static void set_size_udp_pack (udp_pack_t pack, const uint16_t size)

  Function raw write size UDP package.
- uint16_t get_size_data_udp_pack (udp_pack_t pack)

  Function for getting size data.
- ssize_t add_data_udp_pack (udp_pack_t pack, void *data, const uint16_t size)

  Function addition data in UDP package.
- ssize_t set_data_udp_pack (udp_pack_t pack, void *data, uint16_t size)

  Function overriding data in UDP package.
- ssize_t add_byte_udp_pack (udp_pack_t pack, uint8_t byte)

  Function addition byte in UDP package.
- ssize_t set_input_data_udp_pack (udp_pack_t pack)

  Function read data from stdin in UDP package.
- ssize_t set_file_data_udp_pack (udp_pack_t pack, const char *const file_name)

  Function read data from file in UDP package.
- static uint32_t sum_compute (void *ptr, uint16_t nbytes)

Function calculate sum in big endian.

- static uint16_t checksum_compute (uint32_t sum)

  Function calculating from sum big endian to checksum big endian.
- static void calculate_checksum_udp_pack (udp_pack_t pack)

  Function calculate checksum for ip header and UDP package.
- ssize_t set_interface_udp_pack (udp_pack_t pack, const char *const interface)

  Function for pick interface to send UDP package.
- static void * get_pack_udp_pack (udp_pack_t pack)

  Function raw get pointer on UDP package.
- ssize_t send_udp_pack (udp_pack_t pack)

  Function to send UDP package.
- static void * inet_mac (const char *const mac_address)

  Function return mac address in big endian.
- ssize_t set_mac_address_source_udp_pack (udp_pack_t pack, const char *const mac_address)

  Function setting mac address for source.
- ssize_t set_mac_address_destantion_udp_pack (udp_pack_t pack, const char *const mac_address)

  Function setting mac address for destantion.
- char * get_ip_address_source_udp_pack (udp_pack_t pack)

  Function for getting ip address source.
- char * get_ip_address_destantion_udp_pack (udp_pack_t pack)

  Function for getting ip address destantion.
- char * get_interface_udp_pack (udp_pack_t pack)

  Function for getting interface.
- char * get_mac_address_source_udp_pack (udp_pack_t pack)

  Function getting mac address for source.
- char * get_mac_address_destantion_udp_pack (udp_pack_t pack)

  Function setting mac address for destantion.
- char * get_port_destantion_udp_pack (udp_pack_t pack)

  Function for getting port destantion.
- char * get_port_source_udp_pack (udp_pack_t pack)

  Function for getting port source.
- ssize_t print_udp_pack (udp_pack_t pack)

  Function print all info about udp pack.
- char * get_data_udp_pack (udp_pack_t pack)

  Function for getting data.
- void destroy_udp_pack (udp_pack_t pack)

  Function free UDP package.

Variables

- struct udp_head PACKED

9.3.1 Detailed Description

Code file for work udp package.

Author

Vladsanin777

9.3.2 Macro Definition Documentation

9.3.2.1 HEAD_ETH

```
#define HEAD_ETH sizeof(struct ethhdr)
```

9.3.2.2 HEAD_IP

```
#define HEAD_IP sizeof(struct iphdr)
```

9.3.2.3 HEAD_PSEUDO

```
#define HEAD_PSEUDO sizeof(struct pseudo_header)
```

9.3.2.4 HEAD_UDP

```
#define HEAD_UDP sizeof(struct udp_head)
```

9.3.2.5 HEAD_UDP_IP

```
#define HEAD_UDP_IP (HEAD_IP + HEAD_UDP)
```

9.3.2.6 MAX_SIZE_DATA

```
#define MAX_SIZE_DATA (0xFFFF - HEAD_UDP_IP)
```

9.3.2.7 MIN

```
#define MIN(
              left,
              rigth)
```

Value:

```
(((left) < ((typeof(left))rigth)) ? (left) : ((typeof(left))rigth))
```

9.3.2.8 NULL_CHECKSUM

```
#define NULL_CHECKSUM 0x0000
```

9.4 udp_lib/udp.h File Reference

Header file for work udp package.

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

Typedefs

- typedef struct udp_pack * udp_pack_t
  UDP packet descriptor.

Functions

- udp_pack_t init_udp_pack (void)
  Function for create object UDP package.
- ssize_t set_port_source_udp_pack (udp_pack_t pack, const char *const port)
  Function for setting source port in UDP package.
- ssize_t set_port_destantion_udp_pack (udp_pack_t pack, const char *const port)
  Function for setting destantion port in UDP package.
- ssize_t set_ip_address_source_udp_pack (udp_pack_t pack, const char *const ip)
  Function for setting source ip address in UDP package.
- ssize_t set_ip_address_destantion_udp_pack (udp_pack_t pack, const char *const ip)
  Function for setting destantion ip address in UDP package.
- ssize_t add_data_udp_pack (udp_pack_t pack, void *data, uint16_t size)
  Function addition data in UDP package.
- ssize_t set_data_udp_pack (udp_pack_t pack, void *data, const uint16_t size)
  Function overriding data in UDP package.
- ssize_t add_byte_udp_pack (udp_pack_t pack, const uint8_t byte)
  Function addition byte in UDP package.
- ssize_t set_input_data_udp_pack (udp_pack_t pack)
  Function read data from stdin in UDP package.
- ssize_t set_file_data_udp_pack (udp_pack_t pack, const char *const file_name)
  Function read data from file in UDP package.
- ssize_t set_interface_udp_pack (udp_pack_t pack, const char *const interface)
  Function for pick interface to send UDP package.
- ssize_t send_udp_pack (udp_pack_t pack)
  Function to send UDP package.
- char * get_mac_address_source_udp_pack (udp_pack_t pack)
  Function getting mac address for source.
- char * get_mac_address_destantion_udp_pack (udp_pack_t pack)
  Function setting mac address for destantion.
- ssize_t set_mac_address_source_udp_pack (udp_pack_t pack, const char *const mac_address)
  Function setting mac address for source.
- ssize_t set_mac_address_destantion_udp_pack (udp_pack_t pack, const char *const mac_address)
  Function setting mac address for destantion.
- char * get_interface_udp_pack (udp_pack_t pack)
  Function for getting interface.
- char * get_port_source_udp_pack (udp_pack_t pack)

Function for getting port source.

- char * get_port_destantion_udp_pack (udp_pack_t pack)

  Function for getting port destantion.
- char * get_ip_address_source_udp_pack (udp_pack_t pack)

  Function for getting ip address source.
- char * get_ip_address_destantion_udp_pack (udp_pack_t pack)

  Function for getting ip address destantion.
- uint16_t get_size_data_udp_pack (udp_pack_t pack)

  Function for getting size data.
- char * get_data_udp_pack (udp_pack_t pack)

  Function for getting data.
- char * get_data_hex_udp_pack (udp_pack_t pack)

  Function for getting data hex.
- ssize_t print_udp_pack (udp_pack_t pack)

  Function print all info about udp pack.
- void destroy_udp_pack (udp_pack_t pack)

  Function free UDP package.

9.4.1 Detailed Description

Header file for work udp package.

Author

Vladsanin777

9.5 udp.h

Go to the documentation of this file.

```
00001
00006
00007 #include <stdint.h>
00008 #include <stdlib.h>
00009
00015
00019 struct udp_pack;
00020
00026 typedef struct udp_pack * udp_pack_t;
00027
00045 udp_pack_t init_udp_pack(void);
00046
00066 ssize_t set_port_source_udp_pack(udp_pack_t pack, const char * const port);
00067
00087 ssize_t set_port_destantion_udp_pack(udp_pack_t pack, const char * const port);
00088
00112 ssize_t set_ip_address_source_udp_pack( \
00113         udp_pack_t pack, const char * const ip);
00114
00138 ssize_t set_ip_address_destantion_udp_pack( \
00139         udp_pack_t pack, const char * const ip);
00140
00165 ssize_t add_data_udp_pack(udp_pack_t pack, void * data, uint16_t size);
00166
00188 ssize_t set_data_udp_pack(udp_pack_t pack, void * data,  \
00189         const uint16_t size);
00190
00215 ssize_t add_byte_udp_pack(udp_pack_t pack, \
00216         const uint8_t byte);
00217
00236 ssize_t set_input_data_udp_pack(udp_pack_t pack);
00237
00257 ssize_t set_file_data_udp_pack( \
```

```
00258         udp_pack_t pack, const char * const file_name);
00259
00279 ssize_t set_interface_udp_pack( \
00280         udp_pack_t pack, const char * const interface);
00281
00300 ssize_t send_udp_pack(udp_pack_t pack);
00301
00330 char * get_mac_address_source_udp_pack(udp_pack_t pack);
00331
00360 char * get_mac_address_destantion_udp_pack(udp_pack_t pack);
00361
00385 ssize_t set_mac_address_source_udp_pack(udp_pack_t pack, \
00386         const char * const mac_address);
00387
00411 ssize_t set_mac_address_destantion_udp_pack(udp_pack_t pack, \
00412         const char * const mac_address);
00413
00445 char * get_interface_udp_pack(udp_pack_t pack);
00446
00478 char * get_port_source_udp_pack(udp_pack_t pack);
00479
00511 char * get_port_destantion_udp_pack(udp_pack_t pack);
00512
00544 char * get_ip_address_source_udp_pack(udp_pack_t pack);
00545
00577 char * get_ip_address_destantion_udp_pack(udp_pack_t pack);
00578
00605 uint16_t get_size_data_udp_pack(udp_pack_t pack);
00606
00629 char * get_data_udp_pack(udp_pack_t pack);
00630
00658 char * get_data_hex_udp_pack(udp_pack_t pack);
00659
00682 ssize_t print_udp_pack(udp_pack_t pack);
00683
00701 void destroy_udp_pack(udp_pack_t pack);
00702
```

Index