Practica 1 - Redes de Comunicaciones II

Generado por Doxygen 1.8.9.1

Martes, 15 de Marzo de 2016 23:53:56

Índice general

Capítulo 1

Indice de archivos

1.1. Lista de archivos

Lista de todos los archivos documentados y con descripciones breves:

Capítulo 2

Documentación de archivos

2.1. Referencia del Archivo src/G-2301-01-P1-irc_server.c

Servidor IRC.

```
#include "G-2301-01-P1-server.h"
#include "G-2301-01-P1-irc_server.h"
#include <G-2301-01-P1-irc.h>
#include <G-2301-01-P1-tools.h>
#include <G-2301-01-P1-daemon.h>
#include <redes2/irc.h>
#include <stdlib.h>
#include <stdio.h>
#include <syslog.h>
```

'typedefs'

- typedef int(∗ **comm_t**) (char ∗, void ∗)

Funciones

- void **repair_command** (char ∗command)
- char ∗ get_motd_path ()

  Devuelve el path del MOTD.
- int set_user (int socketd, char ∗user)

  Asocia a un socketd el usuario.
- int set_nick (int socketd, char ∗nick)

  Asocia a un socketd el nick.
- int get_socketd_byuser (char ∗user)

  Asocia a un socketd el nick.
- int get_socketd_bynick (char ∗nick)

  Busca el socket de un usuario por nick.
- char ∗ get_nick (int socketd)

  Devuelve el nick de un usuario.
- char ∗ get_user (int socketd)

  Devuelve el user de un usuario.
- int update_timeout (int socketd)

  Actualiza el contador de timeout de un socketd.

- void * handler (void *data)
  Manejador de las conexiones.
- void **do_on_disconnect** (void *data)
- int init_commands ()
  Inicializa las llamadas de atencion de los comandos.
- int init_memspace ()
  Inicializa las estructuras.
- int free_memspace ()
  Borra las estructuras auxiliares del servidor.
- int process_command (char *command, void *data)
  Procesa un comando.
- void * timeout_thread (void *data)
  Hilo de desconexion por timeout.
- int **read_channels** ()
- int main (int argc, char **argv)
  Llamada principal del servidor.

Variables

- comm_t **commands** [NCOMMANDS]
- char ** **users**
- char ** **nicks**
- char * **timeout**
- char **running** =1
- char **path_motd** [256] ="/tmp/irc_tmp"
- pthread_t **t_timeout**
- pthread_mutex_t **m_users**
- pthread_mutex_t **m_nicks**
- pthread_mutex_t **m_timeout**

2.1.1. Descripción detallada

Servidor IRC.

Autor

Sergio Fuentes sergio.fuentesd@estudiante.uam.es
Daniel Perdices daniel.perdices@estudiante.uam.es

Fecha

2016/02/10

2.1.2. Documentación de las funciones

2.1.2.1. char* get_motd_path ( )

Devuelve el path del MOTD.

Devuelve

Dicho path al fichero

2.1.2.2. char* get_nick (int *socketd*)

Devuelve el nick de un usuario.

Parámetros

socketd	El socket del usuario

Devuelve

El nick del usuario

2.1.2.3. int get_socketd_bynick (char * *nick*)

Busca el socket de un usuario por nick.

Parámetros

nick	Nick del usuario

Devuelve

el socket del usuario

2.1.2.4. int get_socketd_byuser (char * *user*)

Asocia a un socketd el nick.

Parámetros

socketd	Socket del usuario
user	Nick del usuario

Devuelve

IRCSVROK en caso adecuado, IRCSVERR (<0) en otro caso

2.1.2.5. char* get_user (int *socketd*)

Devuelve el user de un usuario.

Parámetros

socketd	El socket del usuario

Devuelve

El user del usuario

2.1.2.6. void* handler (void * *data*)

Manejador de las conexiones.

Parámetros

data	Datos de la conexion y el mensaje TCP

Devuelve

IRCSVR_OK

2.1.2.7. int main (int *argc,* char ∗∗ *argv*)

Llamada principal del servidor.

Parámetros

argc	Num de argumentos
argv	Argumentos

Devuelve

0

2.1.2.8. int process_command (char * *command,* void * *data*)

Procesa un comando.

Parámetros

command	El comando
data	Datos de las conexion

Devuelve

IRCSVROK en caso adecuado, IRCSVERR (<0) en otro caso

2.1.2.9. int set_nick (int *socketd,* char * *nick*)

Asocia a un socketd el nick.

Parámetros

socketd	Socket del usuario
user	Nick del usuario

Devuelve

IRCSVROK en caso adecuado, IRCSVERR (<0) en otro caso

2.1.2.10. int set_user (int *socketd,* char * *user*)

Asocia a un socketd el usuario.

Parámetros

socketd	Socket del usuario
user	Nombre del usuario

Devuelve

IRCSVROK en caso adecuado, IRCSVERR (<0) en otro caso

2.1.2.11. void* timeout_thread (void * *data*)

Hilo de desconexion por timeout.

Parámetros

data	NULL

Devuelve

No devuelve ningun valor

2.1.2.12. int update_timeout (int *socketd*)

Actualiza el contador de timeout de un socketd.

Parámetros

socketd	El socket del usuario

Devuelve

IRCSVROK

2.2. Referencia del Archivo srclib/G-2301-01-P1-daemon.c

Libreria de creacion de un daemon.

```
#include <stdio.h>
#include <stdlib.h>
#include <syslog.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
#include <string.h>
#include "G-2301-01-P1-daemon.h"
```

Funciones

- int daemonize (char *srvcid)
  Crea un servicio daemon identificado.

2.2.1. Descripción detallada

Libreria de creacion de un daemon.

Autor

Sergio Fuentes sergio.fuentesd@estudiante.uam.es
Daniel Perdices daniel.perdices@estudiante.uam.es

Fecha

2016/02/10

2.2.2. Documentación de las funciones

2.2.2.1. int daemonize (char * *srvcid*)

Crea un servicio daemon identificado.

Parámetros

srvcid	Identificacion del daemon

Devuelve

DMNOK o DMNERR_∗ en caso de error

2.3. Referencia del Archivo srclib/G-2301-01-P1-irc.c

Libreria de manejo de mensajes IRC.

```
#include <G-2301-01-P1-irc.h>
#include <redes2/irc.h>
#include <syslog.h>
#include <G-2301-01-P1-server.h>
#include <G-2301-01-P1-tools.h>
#include "G-2301-01-P1-irc_server.h"
```

Funciones

- int nick (char ∗command, void ∗more)
  Atiende el comando NICK.
- int user (char ∗command, void ∗more)
  Atiende el comando USER.
- int list (char ∗command, void ∗more)
  Atiende el comando LIST.
- int ping (char ∗command, void ∗more)
  Atiende el comando PING.
- int pong (char ∗command, void ∗more)
  "Atiende" el comando PONG
- int join (char ∗command, void ∗more)
  Atiende el comando JOIN.
- int privmsg (char ∗command, void ∗more)
  Atiende el comando PRIVMSG.
- int mode (char ∗command, void ∗more)
  Atiende el comando MODE.
- int part (char ∗command, void ∗more)
  Atiende el comando PART.
- int names (char ∗command, void ∗more)
  Atiende el comando NAMES.
- int no_command (char ∗command, void ∗more)
  Atiende comandos desconocidos.
- int topic (char ∗command, void ∗more)
  Atiende el comando TOPIC.
- int whois (char ∗command, void ∗more)
  Atiende el comando WHOIS.
- int quit (char ∗command, void ∗more)
  Atiende el comando QUIT.
- int motd (char ∗command, void ∗more)
  Atiende el comando MOTD.

- int away (char *command, void *more)
  Atiende el comando AWAY.
- int kick (char *command, void *more)
  Atiende el comando KICK.
- int time_c (char *command, void *more)
  Atiende el comando TIME.
- int version (char *command, void *more)
  Atiende el comando VERSION.
- int notice (char *command, void *more)
  Atiende el comando NOTICE.

2.3.1. Descripción detallada

Libreria de manejo de mensajes IRC.

Autor

Sergio Fuentes sergio.fuentesd@estudiante.uam.es
Daniel Perdices daniel.perdices@estudiante.uam.es

Fecha

2016/02/10

2.3.2. Documentación de las funciones

2.3.2.1. int away (char * *command,* void * *more*)

Atiende el comando AWAY.

Parámetros

command	El comando recibido
more	Puntero a estructura conn_data auxiliar

Devuelve

Ningún valor definido, la función controlan el error de manera interna

2.3.2.2. int join (char * *command,* void * *more*)

Atiende el comando JOIN.

Parámetros

command	El comando recibido
more	Puntero a estructura conn_data auxiliar

Devuelve

Ningún valor definido, la función controlan el error de manera interna

2.3.2.3. int kick (char * *command,* void * *more*)

Atiende el comando KICK.

Parámetros

command	El comando recibido
more	Puntero a estructura conn_data auxiliar

Devuelve

Ningún valor definido, la función controlan el error de manera interna

2.3.2.4. int list (char * *command,* void * *more*)

Atiende el comando LIST.

Parámetros

command	El comando recibido
more	Puntero a estructura conn_data auxiliar

Devuelve

Ningún valor definido, la función controlan el error de manera interna

2.3.2.5. int mode (char * *command,* void * *more*)

Atiende el comando MODE.

Parámetros

command	El comando recibido
more	Puntero a estructura conn_data auxiliar

Devuelve

Ningún valor definido, la función controlan el error de manera interna

2.3.2.6. int motd (char * *command,* void * *more*)

Atiende el comando MOTD.

Parámetros

command	El comando recibido
more	Puntero a estructura conn_data auxiliar

Devuelve

Ningún valor definido, la función controlan el error de manera interna

2.3.2.7. int names (char * *command,* void * *more*)

Atiende el comando NAMES.

Parámetros

command	El comando recibido
more	Puntero a estructura conn_data auxiliar

Devuelve

Ningún valor definido, la función controlan el error de manera interna

2.3.2.8. int nick (char * *command,* void * *more*)

Atiende el comando NICK.

Parámetros

command	El comando recibido
more	Puntero a estructura conn_data auxiliar

Devuelve

Ningún valor definido, la función controlan el error de manera interna

2.3.2.9. int no_command (char * *command,* void * *more*)

Atiende comandos desconocidos.

Parámetros

command	El comando recibido
more	Puntero a estructura conn_data auxiliar

Devuelve

Ningún valor definido, la función controlan el error de manera interna

2.3.2.10. int notice (char * *command,* void * *more*)

Atiende el comando NOTICE.

Parámetros

command	El comando recibido
more	Puntero a estructura conn_data auxiliar

Devuelve

Ningún valor definido, la función controlan el error de manera interna

2.3.2.11. int part (char * *command,* void * *more*)

Atiende el comando PART.

Parámetros

command	El comando recibido
more	Puntero a estructura conn_data auxiliar

Devuelve

Ningún valor definido, la función controlan el error de manera interna

2.3.2.12. int ping (char * *command,* void * *more*)

Atiende el comando PING.

Parámetros

command	El comando recibido
more	Puntero a estructura conn_data auxiliar

Devuelve

Ningún valor definido, la función controlan el error de manera interna

2.3.2.13. int pong (char * *command,* void * *more*)

"Atiende" el comando PONG

Parámetros

command	El comando recibido
more	Puntero a estructura conn_data auxiliar

Devuelve

Ningún valor definido, la función controlan el error de manera interna

2.3.2.14. int privmsg (char * *command,* void * *more*)

Atiende el comando PRIVMSG.

Parámetros

command	El comando recibido
more	Puntero a estructura conn_data auxiliar

Devuelve

Ningún valor definido, la función controlan el error de manera interna

2.3.2.15. int quit (char * *command,* void * *more*)

Atiende el comando QUIT.

Parámetros

command	El comando recibido
more	Puntero a estructura conn_data auxiliar

Devuelve

Ningún valor definido, la función controlan el error de manera interna

2.3.2.16. int time_c (char * *command,* void * *more*)

Atiende el comando TIME.

Parámetros

command	El comando recibido
more	Puntero a estructura conn_data auxiliar

Devuelve

Ningún valor definido, la función controlan el error de manera interna

2.3.2.17. int topic (char * *command,* void * *more*)

Atiende el comando TOPIC.

Parámetros

command	El comando recibido
more	Puntero a estructura conn_data auxiliar

Devuelve

Ningún valor definido, la función controlan el error de manera interna

2.3.2.18. int user (char * *command,* void * *more*)

Atiende el comando USER.

Parámetros

command	El comando recibido
more	Puntero a estructura conn_data auxiliar

Devuelve

Ningún valor definido, la función controlan el error de manera interna

2.3.2.19. int version (char * *command,* void * *more*)

Atiende el comando VERSION.

Parámetros

command	El comando recibido
more	Puntero a estructura conn_data auxiliar

Devuelve

Ningún valor definido, la función controlan el error de manera interna

2.3.2.20. int whois (char * *command,* void * *more*)

Atiende el comando WHOIS.

Parámetros

command	El comando recibido
more	Puntero a estructura conn_data auxiliar

Devuelve

Ningún valor definido, la función controlan el error de manera interna

2.4. Referencia del Archivo srclib/G-2301-01-P1-server.c

Libreria de manejo de un servidor de sockets.

```
#include <stdlib.h>
#include <stdio.h>
#include <pthread.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <sys/select.h>
#include <sys/time.h>
#include <unistd.h>
#include <strings.h>
#include <string.h>
#include <signal.h>
#include <arpa/inet.h>
#include <G-2301-01-P1-tcp.h>
#include "G-2301-01-P1-server.h"
```

Funciones

- int connection_add (int socketd)

  Añade una conexion al conjunto del select.
- int connection_rmv (int socketd)

  Borra una conexion del conjunto del select.
- int is_connected (int socketd)

  Indica si hay una conexion abierta.
- int is_readable (int socketd)

  Indica si hay una conexion que requiere atencion.
- int connection_block (int socketd)

  Bloquea una conexion.

- int connection_unblock (int socketd)
  Desbloquea una conexion.
- int connection_isblocked (int socketd)
  Comprueba si una conexion esta bloqueada.
- char * get_ip_from_connection (int socketd)
  Devuelve la ip asociada a un socked.
- void sigint_handler (int sig)
  Manejador de la señal.
- int server_launch (uint16_t port, void *(*handler)(void *), void *more)
  Lanza un servidor que atiende cada mensaje recibido con la funcion handler.
- int server_stop ()
  Para el servidor TCP.
- int set_do_on_disconnect (void(*handler)(void *))
  Para el servidor TCP.

Variables

- pthread_mutex_t **mutex_conn**
- fd_set **connections**
- fd_set **readable**
- fd_set **blocked**
- int **loop**
- char **ips** [FD_SETSIZE][16]
- void(* **handler_disconnection**)(void *) = NULL

2.4.1. Descripción detallada

Libreria de manejo de un servidor de sockets.

Autor

Sergio Fuentes sergio.fuentesd@estudiante.uam.es
Daniel Perdices daniel.perdices@estudiante.uam.es

Fecha

2016/02/01

2.4.2. Documentación de las funciones

2.4.2.1. int connection_add (int *socketd*)

Añade una conexion al conjunto del select.

Parámetros

socketd	Descriptor del socket

Devuelve

SERVOK

2.4.2.2. int connection_block (int *socketd*)

Bloquea una conexion.

Parámetros

socketd	Descriptor del socket

Devuelve

SERVOK

2.4.2.3. int connection_isblocked (int *socketd*)

Comprueba si una conexion esta bloqueada.

Parámetros

socketd	Descriptor del socket

Devuelve

0 si no esta bloqueada, otro valor si esta bloqueada

2.4.2.4. int connection_rmv (int *socketd*)

Borra una conexion del conjunto del select.

Parámetros

socketd	Descriptor del socket

Devuelve

SERVOK

2.4.2.5. int connection_unblock (int *socketd*)

Desbloquea una conexion.

Parámetros

socketd	Descriptor del socket

Devuelve

SERVOK

2.4.2.6. char* get_ip_from_connection (int *socketd*)

Devuelve la ip asociada a un socked.

Parámetros

socketd	Descriptor del socket

Devuelve

La ip del socket en decimal

2.4.2.7. int is_connected (int *socketd*)

Indica si hay una conexion abierta.

Parámetros

socketd	Descriptor del socket

Devuelve

0 si esta cerrada, cualquier valor en otro caso

2.4.2.8. int is_readable (int *socketd*)

Indica si hay una conexion que requiere atencion.

Parámetros

socketd	Descriptor del socket

Devuelve

0 si esta cerrada, cualquier valor en otro caso

2.4.2.9. int server_launch (uint16_t *port,* void ∗(∗)(void ∗) *handler,* void ∗ *more*)

Lanza un servidor que atiende cada mensaje recibido con la funcion handler.

Parámetros

port	Puerto en el que abrir el servidor
handler	Rutina de atencion de los mensajes
more	Parametros adicionales que se necesiten en la rutina de atencion

Devuelve

SERVOK en caso de que el servidor termine correctamente un numero negativo en caso de error

2.4.2.10. int server_stop ( )

Para el servidor TCP.

Devuelve

SERVOK si el servidor se para, SERVERR_NRUN si no hay servidor

2.4.2.11. int set_do_on_disconnect (void(∗)(void ∗) *handler*)

Para el servidor TCP.

Devuelve

SERVOK si el servidor se para, SERVERR_NRUN si no hay servidor

2.4.2.12. void sigint_handler (int *sig*)

Manejador de la señal.

Parámetros

sig	señal recibido

Devuelve

0 si no esta bloqueada, otro valor si esta bloqueada

2.5. Referencia del Archivo srclib/G-2301-01-P1-tcp.c

Libreria de manejo de sockets TCP.

```
#include "G-2301-01-P1-tcp.h"
#include <stdio.h>
#include <stdlib.h>
#include <netdb.h>
#include <pthread.h>
#include <strings.h>
#include <string.h>
#include <unistd.h>
#include <syslog.h>
```

Funciones

- int server_tcpsocket_open (uint16_t port, int ∗socketd)
  Abre un socket TCP para servidor.
- int client_tcpsocket_open (uint16_t port, int ∗socketd, char ∗hostname)
  Abre un socket TCP para cliente.
- int tcpsocket_accept (int socketd, tcpsocket_args ∗args)
  Acepta una conexion a socket TCP.
- int tcpsocket_snd (int socketd, void ∗data, size_t len)
  Envia los datos a traves del socket TCP.
- int tcpsocket_rcv (int socketd, void ∗data, size_t max, size_t ∗len)
  Recibe a traves del socket TCP.
- void tcpsocket_close (int socketd)
  Cierra un socket TCP.

2.5.1. Descripción detallada

Libreria de manejo de sockets TCP.

Autor

Sergio Fuentes sergio.fuentesd@estudiante.uam.es
Daniel Perdices daniel.perdices@estudiante.uam.es

Fecha

2016/02/01

2.5.2. Documentación de las funciones

2.5.2.1. int client_tcpsocket_open (uint16_t *port,* int ∗ *socketd,* char ∗ *hostname*)

Abre un socket TCP para cliente.

Parámetros

port	Puerto desde el que se desea escuchar
socketd	Puntero al descriptor del socket

Devuelve

TCPOK si todo fue correcto TCPERR_ARGS/SOCKET/BIND/LISTEN en caso de error con estas funciones

2.5.2.2. int server_tcpsocket_open (uint16_t *port,* int * *socketd*)

Abre un socket TCP para servidor.

Parámetros

port	Puerto desde el que se desea escuchar
socketd	Puntero al descriptor del socket

Devuelve

TCPOK si todo funciono correctamente TCPERR_ARGS/SOCKET/BIND/LISTEN en caso de error con estas funciones

2.5.2.3. int tcpsocket_accept (int *socketd,* tcpsocket_args * *args*)

Acepta una conexion a socket TCP.

Parámetros

socketd	Descriptor del socket escuchado
args	estructura de datos de la conexion

Devuelve

TCPOK si todo fue correcto TCPERR_ARGS/ACCEPT/PTHREAD en caso de error con estas funciones

2.5.2.4. void tcpsocket_close (int *socketd*)

Cierra un socket TCP.

Parámetros

socketd	descriptor del socket

2.5.2.5. int tcpsocket_rcv (int *socketd,* void * *data,* size_t *max,* size_t * *len*)

Recibe a traves del socket TCP.

Parámetros

socketd	Descriptor del socket escuchado
data	puntero a donde se almacenan los datos

max	tamaño de la zona de memoria
len	puntero a la variable de longitud recibida

Devuelve

TCPOK si todo fue correcto TCPERR_ARGS/RECV en caso de error con estas funciones

2.5.2.6. int tcpsocket_snd (int *socketd,* void * *data,* size_t *len*)

Envia los datos a traves del socket TCP.

Parámetros

socketd	Descriptor del socket escuchado
data	puntero a los datos que se han de enviar
len	tamano de los datos a enviar

Devuelve

TCPOK si todo fue correcto TCPERR_ARGS/SEND en caso de error con estas funciones

2.6. Referencia del Archivo srclib/G-2301-01-P1-tools.c

Libreria de utilidades.

```
#include "G-2301-01-P1-tcp.h"
#include <stdio.h>
#include <stdlib.h>
#include <netdb.h>
#include <pthread.h>
#include <strings.h>
#include <string.h>
#include <unistd.h>
#include <syslog.h>
```

'defines'

- #define **strlen** _strlen
- #define **strcmp** _strcmp

Funciones

- size_t _strlen (const char *s)

  Funcion strlen de strings.h.
- int _strcmp (const char *a, const char *b)

  Funcion strcmp de strings.h.

2.6.1. Descripción detallada

Libreria de utilidades.

Autor

Sergio Fuentes sergio.fuentesd@estudiante.uam.es
Daniel Perdices daniel.perdices@estudiante.uam.es

Fecha

2016/02/01

2.6.2. Documentación de las funciones

2.6.2.1. int _strcmp (const char * a, const char * b)

Funcion strcmp de strings.h.

Parámetros

a	Puntero a la cadena de caracteres
b	Puntero a la cadena de caracteres

Devuelve

la diferencia entre ambas cadenas

2.6.2.2. size_t _strlen (const char * s)

Funcion strlen de strings.h.

Parámetros

s	Puntero a la cadena de caracteres

Devuelve

la longitud

2.7. Referencia del Archivo srclib/G-2301-01-P1-udp.c

Libreria de manejo de sockets UDP.

```
#include <G-2301-01-P1-udp.h>
```

Funciones

- int server_udpsocket_open (uint16_t port, int *socketd)

  Abre un socket UDP para servidor.
- int client_udpsocket_open (uint16_t port, int *socketd)

  Abre un socket UDP para cliente.
- int udpsocket_snd (int socketd, void *data, size_t len, udpsocket_side *dst)

  Envia los datos a traves del socket UDP.
- int udpsocket_rcv (int socketd, void *data, size_t max, size_t *len, udpsocket_side *src)

  Recibe a traves del socket UDP.

2.7.1. Descripción detallada

Libreria de manejo de sockets UDP.

Autor

Sergio Fuentes sergio.fuentesd@estudiante.uam.es
Daniel Perdices daniel.perdices@estudiante.uam.es

Fecha

2016/02/01

2.7.2. Documentación de las funciones

2.7.2.1. int client_udpsocket_open (uint16_t *port,* int * *socketd*)

Abre un socket UDP para cliente.

Parámetros

port	Puerto desde el que se desea escuchar
socketd	Puntero al descriptor del socket

Devuelve

UDPOK si todo fue correcto UDPERR_ARGS/SOCKET/BIND/LISTEN en caso de error con estas funciones

2.7.2.2. int server_udpsocket_open (uint16_t *port,* int * *socketd*)

Abre un socket UDP para servidor.

Parámetros

port	Puerto desde el que se desea escuchar
socketd	Puntero al descriptor del socket

Devuelve

UDPOK si todo funciono correctamente UDPERR_ARGS/SOCKET/BIND en caso de error con estas funciones

2.7.2.3. int udpsocket_rcv (int *socketd,* void * *data,* size_t *max,* size_t * *len,* udpsocket_side * *src*)

Recibe a traves del socket UDP.

Parámetros

socketd	Descriptor del socket escuchado
data	puntero a donde se almacenan los datos
max	tamaño de la zona de memoria
len	puntero a la variable de longitud recibida

Devuelve

UDPOK si todo fue correcto UDPERR_ARGS/RECV en caso de error con estas funciones

2.7.2.4. int udpsocket_snd (int *socketd,* void * *data,* size_t *len,* udpsocket_side * *dst*)

Envia los datos a traves del socket UDP.

Parámetros

socketd	Descriptor del socket escuchado
data	puntero a los datos que se han de enviar
len	tamano de los datos a enviar

Devuelve

UDPOK si todo fue correcto UDPERR_ARGS/SEND en caso de error con estas funciones

Índice alfabético