



(<https://poesiabinaria.net/2015/10/programas-que-me-han-salvado-la-vida-este-2015/>)

Estás aquí : Poesía Binaria (<https://poesiabinaria.net/>)

Cliente TCP no bloqueante en C, en el que podemos enviar y recibir en cualquier momento

13 enero, 2014

([/#facebook](#)) ([/#twitter](#))
(https://www.addtoany.com/add_to/google_plus?linkurl=https%3A%2F%2Fpoesiabinaria.net/2014/01/cliente-tcp-no-bloqueante-en-c/&linkname=Cliente%20TCP%20no%20bloqueante%20en%20C%20en%20cualquier-momento%2F&title=Cliente%20TCP%20no%20bloqueante%20en%20C%20en%20cualquier-momento%2F)
 ([/#pinterest](#)) ([/#buffer](#)) ([/#flipboard](#)) ([/#whatsapp](#))
([/#telegram](#)) ([/#meneame](#)) ([/#linkedin](#)) ([/#pocket](#))
(<https://www.addtoany.com/share?url=https%3A%2F%2Fpoesiabinaria.net/2014/01/cliente-tcp-no-bloqueante-en-c/&title=Cliente%20TCP%20no%20bloqueante%20en%20C%20en%20cualquier-momento%2F&title=Cliente%20TCP%20no%20bloqueante%20en%20C%20en%20cualquier-momento%2F>)



(<https://poesiabinaria.net/wp-content/uploads/2014/01/routers.jpg>)

Hace tiempo veíamos un artículo de un servidor que fuera capaz de aceptar múltiples conexiones (<https://poesiabinaria.net/2011/02/creando-un-servidor-que-acepte-multiples-clientes-simultaneos-en-c/>), ahora, toca le toca el turno a la implementación del cliente.

En principio, vamos a implementar un cliente sencillo, en el que conectaremos a un servidor y nos permitirá enviar texto que escribamos por teclado. No podremos recibir nada del servidor, pero nos servirá como primera aproximación:

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <string.h>
4 #include <sys/types.h>
```

```

5  #include <sys/socket.h>
6  #include <netinet/in.h>
7  #include <netdb.h>
8  #include <arpa/inet.h>
9  #include <errno.h>
10 #include <unistd.h>

```

Vamos directamente al `main()`, ya que `usage()` sólo muestra texto y `panic()` provoca la salida cuando se produce un error. Lo primero que hacemos es determinar el servidor y el puerto en función de los argumentos de entrada, creamos el socket y establecemos la dirección del cliente. Como el puerto es 0, el cliente cogerá un puerto libre cualquiera, es más, no nos importa cuál sea, con `bind()`, asignamos dicha dirección al socket.

Luego, creamos la dirección del servidor, como éste puede venir por su hostname, y tal vez debamos resolverlo utilizamos `getbyhostname()`, cuando creamos la variable de dirección conectamos y empezamos el intercambio de mensajes. En este caso utilizamos `fgets()` para pedir información al usuario por teclado y con `send()` enviamos la información especificando qué enviamos (`mens`) y cuánto ocupa (`strlen(mens)+1`, tenemos que incluir el terminador `\0` al final de la cadena).

Nota: Es posible que `fgets()` envíe el retorno de carro producido al pulsar intro cuando termina la cadena. Podemos utilizar una función `trim()` (<https://poesiabinaria.net/2010/03/trim-un-gran-amigo-php-c-c/>), o filtrar con:

```

1  if (mens[strlen (http://www.opengroup.org/onlinepubs/009695399/functions/strlen.html)(mens)-1]!='\n')
2      mens[strlen (http://www.opengroup.org/onlinepubs/009695399/functions/strlen.html)(mens)-1] = '\0';

```

Para poder recibir un mensaje del servidor, bastará con poner:

```

1  size_t recvsize;
2  recvsize = recv(socketfd, mens, MENS_MAX_LEN, 0);
3  if (recvsize== -1)
4      panic("Cannot receive information");
5  mens[recvsize] = '\0';

```

La recepción es muy parecida al envío, sólo que esta vez no sabemos cuánto ocupa el mensaje, por lo que damos un tamaño máximo (si se sobrepasa este tamaño, el resto del mensaje se recibirá en sucesivas llamadas a `recv()`).

Como este mensaje puede no tener un terminador (debe haberse enviado el terminador expresamente), como último carácter en el buffer imponemos este terminador.

Hasta aquí todo es sencillo, el gran problema viene cuando tenemos que establecer turnos fijos para envío y recepción, es más, el cliente podrá estar en modo recepción o modo envío. Pero si estamos acostumbrados a utilizar telnet, o incluso cualquier programa de envío de mensajes o chat, podemos ver que los mensajes pueden salir o entrar en cualquier momento, no tenemos por qué esperar a que alguien escriba un mensaje para recibir todo lo que tenemos pendiente.

Ahora completaremos este ejemplo con una llamada a `select()`, esta función detectará cuándo hay datos en alguno de los descriptores monitorizados y podremos elegir si leer de la entrada estándar o del socket, pero tiene una funcionalidad más que nos resultará muy interesante, dispone de timeout, por lo que si transcurrido un tiempo no ha venido nada por ninguno de los descriptores, la ejecución puede continuar, no estaremos bloqueando el programa de forma indefinida hasta que venga un dato.

```

1  #include <stdlib.h>
2  #include <stdio.h>
3  #include <string.h>
4  #include <sys/types.h>
5  #include <sys/select.h>
6  #include <sys/socket.h>
7  #include <netinet/in.h>
8  #include <netdb.h>
9  #include <arpa/inet.h>
10 #include <errno.h>

```

Foto: Julien Gong Min (<http://www.flickr.com/photos/61368956@N00/5309867377/in/photolist-96dszx-cNwVsA-8rrfyX-bQCKnX-bQCK2i-bBJ47m-8tXQwS-871iPX-8tUJsH-8tUJP2-8tXRuy-8tXS7A-b2ccbR-8tUK4H-8tUKst-8tXPSN-8tXRFb-8tXSes-8tULkx-8tXRQY-8tUKar-8tXQVE-6yusnz-ahLNpc-bBJ4kb-bQCKuM-bBJ4au-bBJ4ss-bQCKhR-bQCJN6-bQCJXr-bQCK7t-bQCJRX-bBJ3Vm-bBJ4j5-bBJ4vy-bBJ4c9-bQCJW4-bQCJQp-bQCKfa-bBJ41G-bQCKaD-6XZWZG-8tXRY3-8tXQf3-5EZUSJ-5EZUz1-aoYNcG-9Wv5U-8tUJAp-8tXR6s>) (Flickr) CC-by

También podría interesarte....

[Facebook](#) ([/#facebook](#)) [Twitter](#) ([/#twitter](#))
(https://www.addtoany.com/add_to/google_plus?linkurl=https%3A%2F%2Fpoesiabinaria.net/recibir-en-cualquier-momento%2F&linkname=Cliente%20TCP%20no%20bloqueante%20en%20C%20en%20el%20que%20podemos%20enviar%20y%20recibir%20en%20cualquier%20momento%2F)
[Pinterest](#) ([/#pinterest](#)) [Buffer](#) ([/#buffer](#)) [Flipboard](#) ([/#flipboard](#)) [Whatsapp](#) ([/#whatsapp](#))
[Telegram](#) ([/#telegram](#)) [Meneame](#) ([/#meneame](#)) [LinkedIn](#) ([/#linkedin](#)) [Pocket](#) ([/#pocket](#))
(<https://www.addtoany.com/share?url=https%3A%2F%2Fpoesiabinaria.net/recibir-en-cualquier-momento%2F&title=Cliente%20TCP%20no%20bloqueante%20en%20C%20en%20el%20que%20podemos%20enviar%20y%20recibir%20en%20cualquier%20momento%2F>)

Posted in : [C/C++](https://poesiabinaria.net/category/codigoc/) (<https://poesiabinaria.net/category/codigoc/>) , [Comunicación](https://poesiabinaria.net/category/comunicacion/) (<https://poesiabinaria.net/category/comunicacion/>)

Tags : [comunicaciones](https://poesiabinaria.net/tag/comunicaciones/) (<https://poesiabinaria.net/tag/comunicaciones/>) , [datos](https://poesiabinaria.net/tag/datos/) (<https://poesiabinaria.net/tag/datos/>) , [descriptor](https://poesiabinaria.net/tag/descriptor/) (<https://poesiabinaria.net/tag/descriptor/>) , [entrada](https://poesiabinaria.net/tag/entrada/) (<https://poesiabinaria.net/tag/entrada/>) , [envio](https://poesiabinaria.net/tag/envio/) (<https://poesiabinaria.net/tag/envio/>) , [estándar](https://poesiabinaria.net/tag/estandar/) (<https://poesiabinaria.net/tag/estandar/>) , [ip](https://poesiabinaria.net/tag/ip/) (<https://poesiabinaria.net/tag/ip/>) , [lectura](https://poesiabinaria.net/tag/lectura/) (<https://poesiabinaria.net/tag/lectura/>) , [recepción](https://poesiabinaria.net/tag/recepcion/) (<https://poesiabinaria.net/tag/recepcion/>) , [simultaneos](https://poesiabinaria.net/tag/simultaneos/) (<https://poesiabinaria.net/tag/simultaneos/>) , [socket](https://poesiabinaria.net/tag/socket/) (<https://poesiabinaria.net/tag/socket/>) , [tcp](https://poesiabinaria.net/tag/tcp/) (<https://poesiabinaria.net/tag/tcp/>)

URL for this post : <https://poesiabinaria.net/2014/01/cliente-tcp-no-bloqueante-en-c-en-el-que-podemos-enviar-y-recibir-en-cualquier-momento/>
(<https://poesiabinaria.net/2014/01/cliente-tcp-no-bloqueante-en-c-en-el-que-podemos-enviar-y-recibir-en-cualquier-momento/>)

Only 1 comment left

[Ir a comentario](#)

Pingback: [BlogESfera.com](http://www.blogesfera.com/cliente-tcp-no-bloqueante-en-c-en-el-que-podemos-enviar-y-recibir-en-cualquier-momento_1868820/) (http://www.blogesfera.com/cliente-tcp-no-bloqueante-en-c-en-el-que-podemos-enviar-y-recibir-en-cualquier-momento_1868820/) / 15 enero, 2014 at 11:28

Leave a Reply

Comment

Name (required)

Email (required)

Website

☐ He leído y acepto la [Política de privacidad](https://poesiabinaria.net/politica-de-privacidad/) (<https://poesiabinaria.net/politica-de-privacidad/>) *

Post Comment

Otras cosas

[Sobre el blog](#) ([/poesiabinaria/informacion/](https://poesiabinaria.net/informacion/))

[Estadísticas](#) ([/poesiabinaria/estadisticas/](https://poesiabinaria.net/estadisticas/))

[Sobre mí](#) ([/poesiabinaria/sobre-mi/](https://poesiabinaria.net/sobre-mi/))

¡Atención! Este sitio utiliza cookies propias y de terceros para mejorar la experiencia de usuario. Si continúas la navegación en el sitio, asumo que estás de acuerdo con esto, aunque siempre puedes [Contacto](#) ([/poesiabinaria/contacto/](https://poesiabinaria.net/contacto/)) [desactivar las cookies en tu navegador](#) [Aceptar](#) [Leer más](#) (<https://poesiabinaria.net/politica-de-cookies/>)

Nube de etiquetas

aplicaciones (/poesiabinaria.net/tag/aplicaciones/) archivos (/poesiabinaria.net/tag/archivos/) array (/poesiabinaria.net/tag/array/) base (/poesiabinaria.net/tag/base/) Bash (/poesiabinaria.net/tag/bash/) c (/poesiabinaria.net/tag/c/) codigo (/poesiabinaria.net/tag/codigo/) comandos (/poesiabinaria.net/tag/comandos/) Comunicación (/poesiabinaria.net/tag/comunicacion/) concurso (/poesiabinaria.net/tag/concurso/) conexión (/poesiabinaria.net/tag/conexion/) configuración (/poesiabinaria.net/tag/configuracion/) consola (/poesiabinaria.net/tag/consola/) correo (/poesiabinaria.net/tag/correo/) datos (/poesiabinaria.net/tag/datos/) Emacs (/poesiabinaria.net/tag/emacs/) facebook (/poesiabinaria.net/tag/facebook/) gnu (/poesiabinaria.net/tag/gnu/) google (/poesiabinaria.net/tag/google/) imagen (/poesiabinaria.net/tag/imagen/) ip (/poesiabinaria.net/tag/ip/) libre (/poesiabinaria.net/tag/libre/) Linux (/poesiabinaria.net/tag/linux/) memoria (/poesiabinaria.net/tag/memoria/) MySQL (/poesiabinaria.net/tag/mysql/) PHP (/poesiabinaria.net/tag/php/) procesos (/poesiabinaria.net/tag/procesos/) programacion (/poesiabinaria.net/tag/programacion/) programas (/poesiabinaria.net/tag/programas/) red (/poesiabinaria.net/tag/red/) script (/poesiabinaria.net/tag/script/) seguridad (/poesiabinaria.net/tag/seguridad/) servicios (/poesiabinaria.net/tag/servicios/) servidor (/poesiabinaria.net/tag/servidor/) sistema (/poesiabinaria.net/tag/sistema/) software (/poesiabinaria.net/tag/software/) ssh (/poesiabinaria.net/tag/ssh/) terminal (/poesiabinaria.net/tag/terminal/) texto (/poesiabinaria.net/tag/texto/) tiempo (/poesiabinaria.net/tag/tiempo/) tuenti (/poesiabinaria.net/tag/tuenti/) usuario (/poesiabinaria.net/tag/usuario/) video (/poesiabinaria.net/tag/video/) web (/poesiabinaria.net/tag/web/) windows (/poesiabinaria.net/tag/windows/)

Últimos comentarios

- BlogESfera.com (http://www.blogesfera.com/cliente-tcp-no-bloqueante-en-c-en-el-que-podemos-enviar-y-recibir-en-cualquier-momento_1868820) en Cómo configurar auto escalado asociado al balanceador de carga de Amazon Web Services (/poesiabinaria.net/2016/12/como-configurar-auto-escalado-asociado-al-balanceador-de-carga-de-amazon-web-services/#comment-83323)
- BlogESfera.com (http://www.blogesfera.com/cliente-tcp-no-bloqueante-en-c-en-el-que-podemos-enviar-y-recibir-en-cualquier-momento_1868820) en Foto: Chuleta de atajos de teclado de Emacs (/poesiabinaria.net/2017/08/foto-chuleta-atajos-teclado-emacs/#comment-83322)
- BlogESfera.com (http://www.blogesfera.com/cliente-tcp-no-bloqueante-en-c-en-el-que-podemos-enviar-y-recibir-en-cualquier-momento_1868820) en Crear múltiples direcciones de correo desde el CLI de Plesk (/poesiabinaria.net/2020/03/crear-multiples-direcciones-correo-desde-cli-plesk/#comment-83321)
- BlogESfera.com (http://www.blogesfera.com/cliente-tcp-no-bloqueante-en-c-en-el-que-podemos-enviar-y-recibir-en-cualquier-momento_1868820) en Singletons en C++ y alguna nota sobre thread safety (I) (/poesiabinaria.net/2014/04/singleton-in-cpp/#comment-83320)
- BlogESfera.com (http://www.blogesfera.com/cliente-tcp-no-bloqueante-en-c-en-el-que-podemos-enviar-y-recibir-en-cualquier-momento_1868820) en Singletons en C++ y alguna nota sobre thread safety (I) (/poesiabinaria.net/2014/04/singleton-in-cpp/#comment-83319)

Lo más comentado

- Aventuras devolviendo una licencia de Windows que venía de serie con un portátil (/poesiabinaria.net/2012/08/aventuras-devolviendo-una-licencia-de-windows-que-venia-de-serie-con-un-portatil/) (47 Comments (/poesiabinaria.net/2012/08/aventuras-devolviendo-una-licencia-de-windows-que-venia-de-serie-con-un-portatil/#comments))
- Crear mapas HTML5 interactivos con RaphaelJS (/poesiabinaria.net/2014/10/crear-mapas-html5-interactivos-con-raphaeljs/) (44 Comments (/poesiabinaria.net/2014/10/crear-mapas-html5-interactivos-con-raphaeljs/#comments))
- Escribiendo en muros desde nuestra aplicación de Facebook (/poesiabinaria.net/2012/01/escribiendo-en-muros-desde-nuestra-aplicacion-de-facebook/) (31 Comments (/poesiabinaria.net/2012/01/escribiendo-en-muros-desde-nuestra-aplicacion-de-facebook/#comments))
- Listar archivos dentro de un directorio o carpeta en C (/poesiabinaria.net/2011/09/listar-archivos-dentro-de-un-directorio-o-carpeta-en-c/) (28 Comments (/poesiabinaria.net/2011/09/listar-archivos-dentro-de-un-directorio-o-carpeta-en-c/#comments))
- Andanzas en Facebook: Publicar en el muro personal fácilmente (/poesiabinaria.net/2009/08/andanzas-en-facebook-publicar-en-el-muro-personal-facilmente/) (27 Comments (/poesiabinaria.net/2009/08/andanzas-en-facebook-publicar-en-el-muro-personal-facilmente/#comments))