



IRC

Internet Relay Chat

Resumen: El objetivo de este proyecto es hacerte escribir tu propio servidor IRC. Para hacerlo, probaremos tu trabajo con un cliente IRC real. Internet se rige por una serie de estándares y protocolos que permiten una interacción completa entre ordenadores. Está bien saberlo.

Índice general

I.	Instrucciones generales	2
II.	Introducción	3
III.	Parte obligatoria	4
IV.	Parte extra	7

Capítulo I

Instrucciones generales

- Tu programa no puede crashear bajo ninguna circunstancia (incluso si se queda sin memoria), y no debe cerrar inesperadamente. Si esto sucede, tu proyecto se considerará no funcional y recibirá un 0 durante la evaluación.
- Si el subject lo requiere, debes entregar un **Makefile** que compilará tus fuentes al output requerido. Tu **Makefile** no debe hacer relink.
- Tu **Makefile** debe contener al menos las reglas `$(NAME)`, `all`, `clean`, `fclean` y `re`.
- Te recomendamos crear tus propios programas de prueba, aunque estos **no serán entregados ni evaluados**. Te dará la opción de probar fácilmente tu trabajo y el de tus compañeros. Encontrarás estas pruebas especialmente útiles durante tu defensa. Sí, durante tu defensa, eres libre de utilizar tus pruebas o las del compañero que estás evaluando.
- Entrega tu trabajo en el repositorio de Git asignado. Solo el trabajo en el repositorio de Git será evaluado. Si Deepthought se asigna a evaluar tu trabajo, lo hará tras las evaluaciones de tus compañeros. Si un error sucede en cualquier sección de la evaluación de Deepthought, esta terminará.

Capítulo II

Introducción

Un IRC o Internet Relay Chat es un protocolo de comunicación textural de Internet. Es una comunicación instantánea principalmente en la forma de discusiones en grupos a través de canales, pero también se puede utilizar como comunicación de uno a uno.

Los clientes IRC se conectan al servidor IRC para acceder a un canal específico. Los servidores IRC se conectan entre ellos para proveer de una red global con canales únicos.

Capítulo III

Parte obligatoria

Nombre de programa	ircserv
Archivos a entregar	
Makefile	Sí
Argumentos	
Funciones autorizadas	Todo en C++ 98. socket, setsockopt, getsockname, getprotobyname, gethostbyname, getaddrinfo, freeaddrinfo, bind, connect, listen, accept, htons, htonl, ntohs, ntohl, inet_addr, inet_ntoa, send, recv, signal, lseek, fstat,fcntl, poll (o equivalente)
Se permite usar libft	
Descripción	Escribe un servidor IRC en C++ 98

- Debes escribir un servidor IRC en C++ 98.
- Si necesitas más funciones de C, puedes utilizarlas pero busca siempre utilizar su versión de C++.
- El estándar de C++ debe ser C++ 98. Tu proyecto debe cumplirlo.
- No se permiten librerías externas, ni Boost, etc.
- Utiliza siempre la versión más “C++” posible (por ejemplo, utiliza `<cstring>` por encima de `<string.h>`).
- En el subject y en la evaluación mencionaremos poll pero puedes utilizar el equivalente que quieras, como select, kqueue y epoll.
- La comunicación entre cliente y servidor debe hacerse a través de TCP/IP(v4) o (v6).
- No necesitas programar un cliente.

- No necesitas gestionar comunicaciones servidor a servidor.
- Tu ejecutable se utilizará de la siguiente manera:

```
./ircserv [host:port_network:password_network] <port> <password>
```

- `host` es el nombre del host al que el IRC debe conectarse para unirse a una red ya existente.
 - `port_network` es el puerto del servidor al que el IRC debe conectarse en `host`.
 - `password_network` es la contraseña utilizada para conectarse a `host`.
 - `port` es el puerto del que tu servidor aceptará conexiones entrantes.
 - `password` es la contraseña utilizada por cualquier cliente IRC que quiera conectarse a tu servidor.
 - Si `host`, `port_network` y `password_network` no se dan, debes crear una nueva red IRC.
- El servidor debe ser capaz de gestionar múltiples clientes a la vez y nunca colgarse. Utilizar forks no está permitido, todas las operaciones de tipo I/O deben ser no bloqueantes y utilizar solo un poll (o equivalente) para todo (read, write, pero también listen, ...)



Te hemos dejado usar `fcntl` porque MacOS X no implementa `write` del mismo modo en que otros sistemas operativos Unix lo permiten. Debes utilizar un fd no bloqueante para tener un resultado similar a otros sistemas operativos.



Dado que estás utilizando fd no bloqueantes, podrías utilizar las funciones `read/recv` o `write/send` sin poll (o equivalente) y tu servidor sería no bloqueante. Esto consumiría más recursos del sistema.

Así que de nuevo, intentar utilizar `read/recv` o `write/send` en cualquier fd sin pasar por poll (o equivalente) te dará una nota 0 y significará el fin de tu evaluación.



Solo puedes utilizar `fcntl` de la siguiente forma: `fcntl(fd, F_SETFL, O_NONBLOCK);`.

Otras flags están PROHIBIDAS.

- Por supuesto, se espera de ti un código limpio. Verifica absolutamente todos los errores, y los casos más extremos (datos parciales recibidos, bajo ancho de banda, etc).

- Para verificar que tu servidor utiliza todo lo que envías, una prueba inicial se puede hacer con `nc` (utiliza CTRL+D para enviar partes del comando):

```
\$> nc 127.0.0.1 6667
com^Dman^Dd
\$>
```

Esto te permitirá enviar primero las letras `com`, `man`, `d\n`. Debes primero agragar los paquetes para reconstruir el comando `command` y poder gestionarlo.

- Muchos clientes IRC existen, debes elegir uno como referencia y utilizarás este durante tu defensa.
- Para ser compatible con este cliente deberás implementar al menos parte del RFC oficial del IRC.
- Utilizar el cliente en tu servidor debe ser como utilizar cualquier servidor IRC, aunque para estar seguros aquí tienes una lista de lo mínimo que se espera:
 - Debes poder conectar el cliente de referencia a tu servidor sin producir ningún error.
 - Debes ser capaz de identificarte, establecer un nick, un usuario, entrar a un canal y enviar y recibir mensajes privados utilizando este cliente.
 - Todos los mensajes de un cliente en un canal se envían a todos los demás clientes del canal.
 - Debes tener operadores y usuarios normales.
 - Algunos comandos y acciones específicas de operadores.

Capítulo IV

Parte extra

Aquí tienes los bonus que puedes añadir a tu IRC para acercarlo a un IRC real. Por supuesto, si la parte obligatoria no es perfecta, ni se te ocurra pensar en los bonus.

- Transferencia de archivos.
- Un bot.