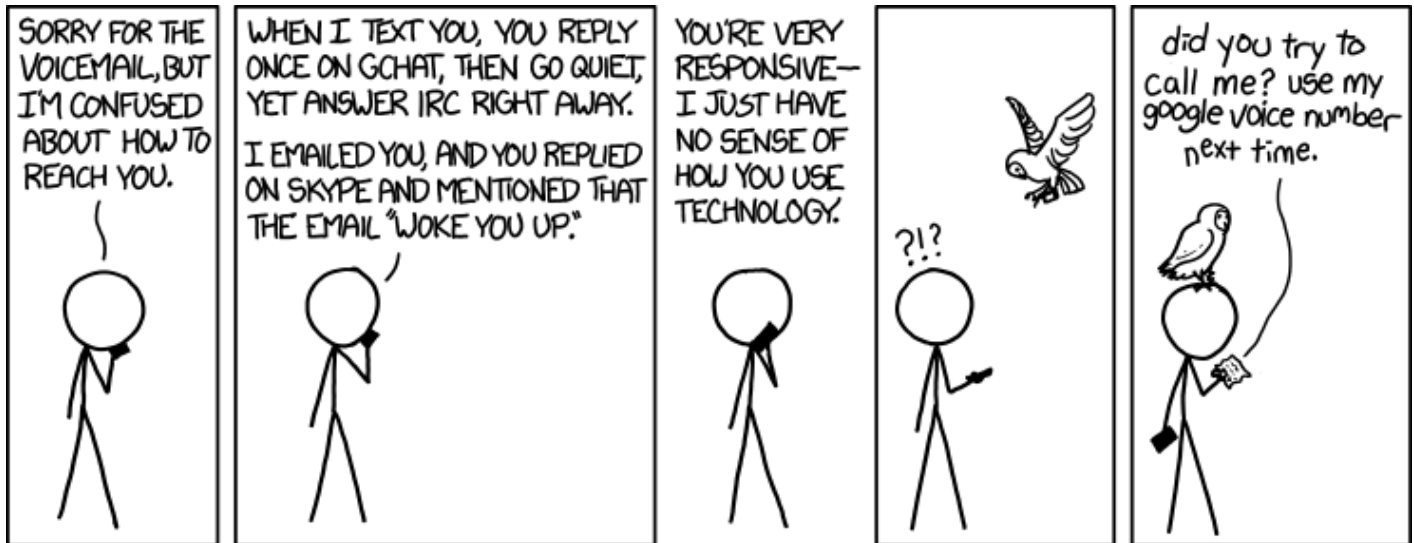


## Tarea 2: Chat *full-duplex* con *Sockets*



### 1. Actividad

Se requiere que desarrolle un programa de *chat* para el sistema operativo GNU/Linux que permita comunicación *full-duplex* entre dos participantes. El programa debe ser implementado en el lenguaje C++ usando la biblioteca de interfaces gráficas de usuario FLTK. Junto a este enunciado se incluyen dos archivos llamados `chat.cpp` y `chat.h`, los cuales ya tienen implementada la interfaz. El flujo de acciones que debe seguir el programa para establecer una sesión de chat es el siguiente:

1. Un usuario debe iniciar el programa (A) sin argumentos de línea de comandos.
2. El programa A debe lanzar un hilo de ejecución adicional el cual debe abrir un *socket* en modo de escucha. El hilo puede ser creado con la biblioteca *Pthreads* o con un objeto `std::thread` de C++11. Este hilo debe entonces esperar a que se acepte una conexión y debe rechazar cualquier conexión pendiente una vez haya sido aceptada la primera (revisar el parámetro *backlog* de la función `listen()`).
3. El otro usuario debe entonces iniciar otra instancia del programa (B) pasando como argumento de línea de comandos la dirección IP del primer usuario.
4. El programa B debe crear un *socket* de cliente y conectarse con el *socket* de servidor establecido por el programa A anteriormente.
5. Al igual que el programa A, el programa B debe también abrir un *socket* de servidor en un hilo de ejecución adicional y colocarlo en modo de escucha.
6. Cuando el programa A acepte la conexión del programa B, deberá tomar la dirección IP del programa B del parámetro `client` de la función `accept()` y establecer una conexión con el *socket* de servidor del programa A usando un *socket* de cliente.

Una vez ejecutado el flujo de programa anterior, ambos programas tendrán dos conexiones abiertas y dos hilos de ejecución. La idea en este punto es que en cada programa el *socket* de cliente conectado con el programa opuesto debe ser usado para enviar el texto del usuario al otro programa; y la conexión establecida en el hilo de ejecución secundario será usada para recibir texto del programa opuesto.

El texto enviado y recibido debe ser mostrado en el componente `m.Output` de la interfaz gráfica usando el siguiente formato:

1. Todo texto producido por el usuario debe ir precedido por la cadena “Yo: ”. Esta funcionalidad ya se encuentra en el código provisto.
2. Todo texto recibido del otro *chat* debe ir precedido de la cadena “Otro: ”.

El acceso al componente `m.Output` debe ser protegido apropiadamente mediante un semáforo de *Pthreads* o un objeto *std::mutex* de C++11.

## 2. Consideraciones

1. Fecha de entrega: viernes 7 de febrero de 2020.
2. La tarea puede ser realizada en grupos de a lo sumo dos (2) personas.
3. El reporte a entregar debe estar identificado con el nombre, apellido y cédula de identidad de todos los integrantes del grupo.
4. Las copias serán penalizadas con la nota mínima para todos los involucrados.
5. No habrá prorrogas, sin excepción.