

Proyecto 2: Mini-Twitter

Objetivo del Proyecto

Después de realizar este proyecto, los estudiantes estarán en capacidad de utilizar herramientas para la comunicación síncrona y asíncrona entre procesos de sistemas basados en Unix, especialmente los pipes y las señales.

Descripción General

Todos hemos oído hablar o pertenecemos la red social twitter (<https://twitter.com/>). Twitter es un programa que permite el envío de mensajes de texto cortos entre múltiples usuarios. También se pueden enviar otros archivos multimedia (imágenes, audio, etc.) Cada usuario en la red social tiene un conjunto de seguidores y, a su vez éstos pueden seguir a otros usuarios. Cada vez que un usuario envía un **tweet**, este mensaje se envía a todos sus seguidores.

En este proyecto implementaremos un **twitter** muy sencillo, cuyos usuarios se conectarán en un mismo computador y se enviarán mensajes de texto de como máximo 200 caracteres. El proyecto a implementar será un modelo “cliente-servidor”, donde los clientes serán la interfaz a los usuarios que envía los tweets y el proceso servidor se encarga del envío de tweets entre usuarios y otras operaciones que permite la red social. **Todos, clientes y el servidor, serán procesos ejecutándose en el mismo computador y comunicándose a través de pipes.** A continuación, explicamos las funciones de estos procesos, comenzando por el servidor al que llamaremos el Gestor de Tweets.

1. El Gestor de Tweets

Es el programa encargado de gestionar el envío de **tweets** entre usuarios y, por lo tanto, de almacenar los datos correspondientes; también almacena las relaciones existentes entre ellos. **Una vez que se ha iniciado el Gestor** los usuarios pueden conectarse. Si un usuario A envía un tweet en un momento determinado, sus seguidores conectados deben recibirlo inmediatamente. El gestor almacenará los tweets para los seguidores que no están conectados.

El proceso **Gestor** se invocará de la siguiente forma desde el *shell*:

\$ gestor -r <relaciones> -p <pipeNom>

Donde:

relaciones: es un archivo de texto que describe las relaciones actuales entre los usuarios del sistema. El sistema tendrá como máximo 10 usuarios registrados, por lo que en el archivo de entrada se representará una matriz de 10X10 números. En cada línea se almacena información sobre las relaciones de un usuario en particular con el resto de los usuarios. Las relaciones entre usuarios se expresan con números binarios, es decir, un cero (0) en una determinada posición significa que no se sigue a un determinado usuario y un uno (1) significa que sí se es seguidor. Aunque hay 10 usuarios inscritos en el sistema, no necesariamente todos estarán conectados al mismo tiempo, ni han establecido todas las relaciones deseadas con el resto de los usuarios. El archivo Relaciones es, por lo tanto, un grafo dirigido inicial de relaciones que puede cambiar. Por ejemplo, para el grafo de relaciones de la Figura 1, el archivo de entrada sería el siguiente:

1	0	1	1	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0
3	1	0	0	0	1	1	1	0	0	0
4	1	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	1	0	0	0
7	1	0	1	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0

Nota 1: la primera columna que está a la izquierda, no es parte del archivo, se coloca aquí en la figura únicamente para dejar claro que la primera fila corresponde a las relaciones del usuario 1, la segunda al usuario 2, etc. Este archivo de ejemplo se colocará en Uvirtual en la misma carpeta donde se coloque el proyecto. **Es solo un ejemplo, otro archivo de entrada puede tener distintas relaciones.**

Nota 2: La gráfica de la figura 1 ilustra las relaciones de la matriz anterior. Una flecha del nodo i al nodo j, indica que i **es seguidor de j**.

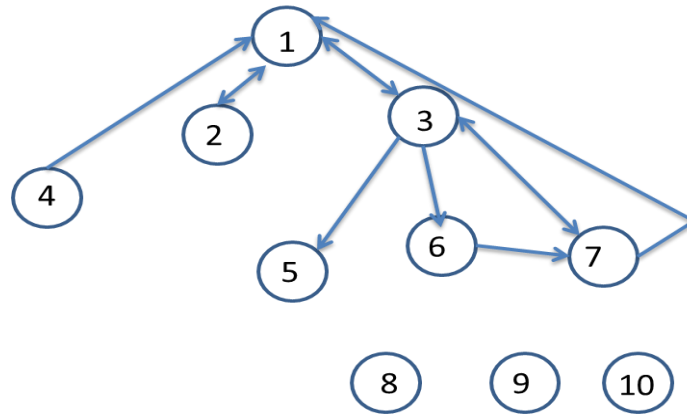


Figura 1: Relaciones entre los 10 usuarios

pipeNom: es el nombre del pipe nominal por donde los usuarios (representados por procesos **Cliente**) se conectarán inicialmente al proceso Gestor. Es el pipe azul en la figura 2 y es por donde los clientes enviarán las peticiones.

Cada operación que realice el Gestor para sus clientes debe comunicarse adecuadamente por la consola, de forma que se pueda seguir lo que el Gestor está haciendo y qué comunicaciones está recibiendo de sus clientes.

2. Los Clientes

Son los procesos que mostrarán la interfaz de la red social a los usuarios. Cada proceso cliente se invocará desde un terminal con los siguientes argumentos:

\$ Cliente -i <Id> -p <pipeNom>

Donde:

Id: identifica al usuario que se quiere conectar en el sistema y será sencillamente un número del 1 al 10. El sistema debe validar que el usuario exista y que no se encuentre ya conectado.

pipeNom: es el nombre del pipe nominal utilizado para conectarse al gestor (pipe azul en la figura 2). Por medio de este pipe los clientes se comunicarán con el **Gestor** para realizar varias operaciones.

Para explicar la semántica de las operaciones, vamos a suponer que el usuario que las solicita es el usuario A.

1. **Follow:** le permite al usuario A seguir a un usuario determinado B. El sistema debe preguntar el identificador del usuario al que se desea seguir (B). Si el identificador no es válido o ya A está siguiendo a ese usuario, el sistema retornará un error. Cuando se comienza a seguir a un determinado usuario B, **no se recibirán los tweets anteriores, a esta operación, que el usuario B haya enviado.**
2. **Unfollow:** le permite al usuario A dejar de seguir a otro usuario B. El sistema debe preguntar el identificador del usuario al que se desea dejar de seguir (B). Si el identificador no es válido

o el usuario que solicita la operación (A) no es seguidor del usuario que está solicitando “desconectar” (B), el sistema retornará un error.

3. **Tweet:** le permite a un usuario enviar un mensaje a sus seguidores. El mensaje será un mensaje en texto plano de corta longitud, con un máximo de 200 caracteres. Los seguidores conectados recibirán el tweet de forma inmediata. Los seguidores no conectados recibirán el tweet (o los tweets) al conectarse.
4. **Desconexión:** Esta opción le permite al usuario desconectarse temporalmente del sistema.

Funcionamiento del Sistema

El **Gestor** es el primer programa que se debe crear e invocar. Éste crea e inicializa las estructuras de datos que permiten almacenar los usuarios, sus relaciones y los tweets. También crea el primer pipe nominal por donde recibirá la información y solicitudes de los usuarios que se van conectando (Ver figura 2a). El Gestor nunca deja de funcionar a menos que se elimine de forma explícita desde el terminal al usar Ctrl+C.

La función de un Gestor es atender las solicitudes de los procesos **Cliente: registro de usuarios, follow, unfollow, enviar un tweet y desconexión**. Veamos lo que hace el Gestor ante estas solicitudes de los clientes:

- **Registro:** Al momento de conectarse un usuario, el proceso **Cliente** enviará el identificador del usuario al **Gestor** usando el pipe. El Gestor, luego de mostrar en su pantalla cual fue la solicitud, le devolverá al Cliente el resultado de la operación (si fue exitosa o no), lo cual dependerá fundamentalmente de si un usuario con el mismo identificador ya se encuentra conectado en el sistema. El Gestor también devolverá los tweets enviados para ese usuario (en caso de que tuviere).
Si Ud. desea crear otro pipe para la comunicación exclusiva entre el Gestor y el Cliente (Figura 2b), puede aprovechar la operación de registro para crear e intercambiar la información para este nuevo pipe.
- Con las operaciones **follow y unfollow** el protocolo es similar, el Cliente envía la información sobre el usuario que desea realizar la operación y sobre el usuario que se desea seguir o dejar de seguir. El Gestor muestra por su pantalla la solicitud del cliente, y le envía al cliente el resultado de la operación (exitosa o no) y actualiza sus estructuras de datos.
- Cuando el Gestor **recibe un tweet** de parte de un determinado **usuario A** deberá realizar las siguientes operaciones:
 - o Reportar por su pantalla cuál es la solicitud del cliente.
 - o Determinar cuáles son los seguidores de A y enviar inmediatamente el tweet a todos los seguidores de A que estén conectados. El tweet aparecerá por la pantalla del proceso Cliente, interrumpiendo las operaciones que esté realizando. Aunque la salida por pantalla se vea interrumpida, el Cliente debe seguir atendiendo las solicitudes del usuario. El Gestor almacenará el tweet para los seguidores de A que estén desconectados.

Los Clientes: al comenzar, deben registrarse con el Gestor. Luego, el proceso Cliente ejecutará un ciclo en el cual le mostrará y ejecutará las distintas operaciones que solicite el usuario. Este programa termina cuando el usuario indique la opción de desconectarse. Noten que el proceso Cliente está pendiente de las opciones del usuario y no de los nuevos tweets que van llegando al sistema. Para poder hacer esto puede hacer uso de señales o hacer implementaciones con varios procesos o hilos en el Cliente.

Consideraciones de Diseño e Implementación

- Tenga en cuenta que el Gestor y los procesos Cliente son independientes. Se comunicarán inicialmente por un único pipe nominal, pero Uds. deciden si requieren de otros pipes tal y como se observa en la Figura 2b.
- El proceso **Gestor** y cada proceso **Cliente** se inician de la forma indicada y con los parámetros señalados, aunque estos pueden venir en cualquier orden. Una vez que el **Cliente** esté en funcionamiento, podrá mostrar un menú a los usuarios donde se soliciten las operaciones descritas; las operaciones pueden ser expresadas también en comandos con sus respectivos argumentos. Deben validar que cada tipo de proceso se levanta con el número de argumentos adecuado.
- El **Gestor** debe ir informando por pantalla sobre las operaciones que está realizando.
- El proyecto lo pueden realizar en C.
- Deben respetar estrictamente el formato del archivo de entrada y la sintaxis de los comandos especificadas en este enunciado.
- Un usuario determinado puede estar conectado un tiempo, desconectarse y luego volverse a conectar.

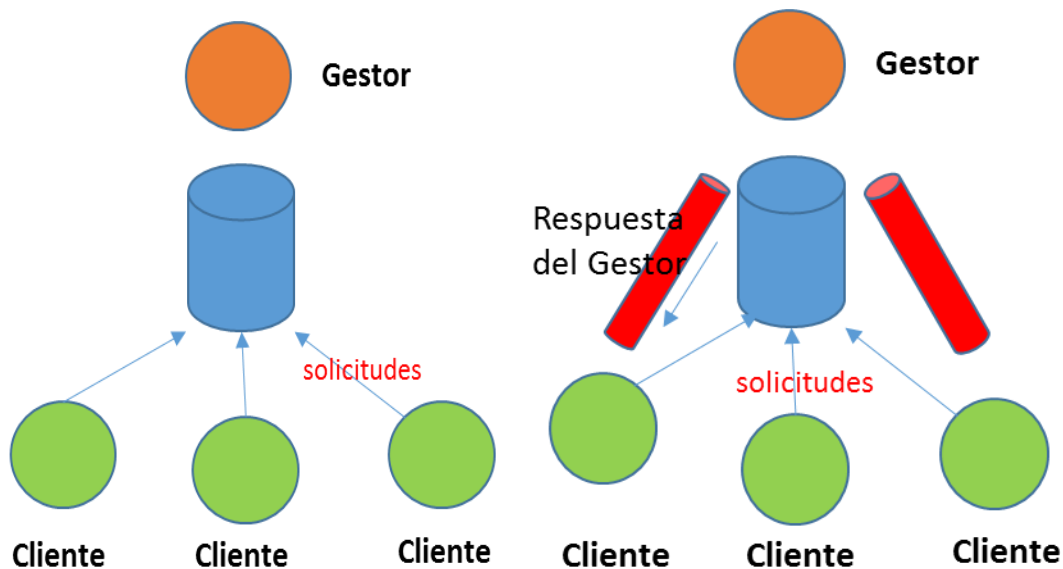


Figura 2a: Pipe inicial por donde todos los clientes envían sus peticiones al gestor.

Figura 2b: Pipes de comunicación entre el Gestor y cada uno de sus clientes (rojos). Por estos pipes el Gestor envía respuestas

Ejemplo de operación del proyecto

1. Se levanta el proceso Gestor, con el archivo que representa las relaciones de la figura 1.
2. Luego, se conectan los usuarios **4, 3 y 8** al sistema. No reciben ningún tweet al iniciar su conexión.
3. Se conecta el usuario 1.
4. El usuario 1 envía el siguiente tweet: *"buenos días queridos seguidores"*. Los usuarios 4 y 3 recibirán y mostrarán el tweet por pantalla.
5. El usuario 10 se conecta.
6. El usuario 4 manda un tweet. No tiene seguidores conectados.
7. El usuario 10 decide seguir al usuario 1 y al usuario 4
8. El usuario 1 envía un nuevo tweet: *"Hoy estoy trabajando"*. Lo reciben los usuarios 3,4 y 10.
9. Se conecta el usuario 7 y recibe los tweets enviados por el usuario 1.

Sobre la Entrega

- El proyecto lo deben realizar en grupos sólo de **dos o de tres estudiantes**.
- Lo deben entregar y sustentar el martes de la semana 17 (**12 de noviembre de 2019**). La entrega, que se realizará por UVirtual (antes de las 11 am del día **12 de noviembre**). Antes de hacer la sustentación se revisará que los estudiantes hayan subido el código a UVirtual.
- La entrega consiste de los códigos fuente y el archivo makefile para hacer la compilación de todo el proyecto.
- El código debe seguir los lineamientos de la Guía de Programación de C. Esta vez debería usar archivos de encabezado (.h). Debe validar llamadas al sistema y todas las entradas del usuario.
- El día de la sustentación un integrante del equipo realizará una modificación al proyecto. El integrante será escogido por los profesores.

Deben seguir todas las especificaciones que este enunciado. Si tiene alguna duda diríjase a la Prof. Mariela Curiel.

Suerte!