Pontificia Universidad Javeriana
Departamento de Ingeniería de Sistemas
Sistemas Operativos, SIU 4085.
julio-noviembre 2018

Proyecto 2: Chattering

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 Linux, especialmente los pipes y las señales.

Descripción General

La mayoría de nosotros usa WhatsApp para comunicarnos rápidamente con conocidos, amigos, familiares, etc.

En este proyecto implementaremos un chat, que al igual que WhatsApp, nos permitirá enviar mensajes directos a una persona o a un grupo de personas. Los mensajes escritos serán observados de forma inmediata por todos los destinatarios. Nuestro sistema se llamará **Chattering**

Implementación

El sistema Chattering está conformado por una arquitectura Cliente/Servidor. *La **arquitectura cliente-servidor** es un modelo de diseño de software en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta (Wikipedia)*

Cada usuario que desee interactuar con otros usuarios a través del Chattering, deberá invocar un proceso *talker* (el cliente). Ese proceso le permitirá enviar y recibir mensajes, crear grupos, listar los usuarios y grupos en el sistema, etc. Así como sucede en las redes sociales, los usuarios se conectan por un determinado periodo de tiempo y luego cierran o terminan el proceso. Chattering estará formado también por un proceso servidor (el Manager), que registrará la información de clientes y grupos en el sistema y será intermediario para la comunicación entre los diferentes talkers.

Todos los procesos se comunicarán a través de pipe nominales. Al crear cada tipo de proceso, recibirá como parámetro el nombre de un único pipe que servirá para las comunicaciones en el sentido cliente->servidor. Posteriormente se podrán crear otros pipes en la dirección servidor->cliente (Ver figura 1).

Talkers

Son los procesos que van a representar a los usuarios registrados en Chattering. Cada proceso usuario se ejecutará desde una consola con los siguientes argumentos:

$ talker ID pipeNom

Donde

ID: identifica al usuario que se quiere conectar en el sistema y será sencillamente un número del 1 al N (N es el máximo número de usuarios en el sistema y será un parámetro de creación del Manager). El sistema debe validar que el número de usuario sea válido (ID <= N), y si existe, que no esté ya conectado en el sistema.

pipeNom: es el nombre del pipe nominal que se utilizará para comunicarse con el proceso Manager. Todos los talkers se crean con el mismo nombre de pipe.

Una vez iniciado el proceso talker, éste ofrecerá un prompt al usuario para que solicite alguna de las siguientes operaciones:

- **List**: lista los usuarios actualmente conectados en el sistema. Para obtener esta lista se debe preguntar al Manager.
- **List friends**: lista los "amigos" del talker con identificador ID (este es el identificador del usuario que invoca la operación). Más adelante se define qué son los "amigos".
- **List GID**: lista los integrantes del grupo con identificador GID. Esta operación falla si GID no es un identificador válido, es decir si no ha sido creado un grupo con este identificador. La información de los grupos se encuentra en el Manager.
- **Rel IDi**: Permite establecer una **relación entre el talker que la invoca (con identificador ID) con el talker con identificador IDi** . Cuando un talker solicita esta operación, el Manager debe registrarla y enviar un mensaje asíncrono al talker con identificador IDi. Para efectos del proyecto se supone que el receptor del mensaje siempre acepta la relación. La operación falla, si el talker IDi no se encuentra conectado en el sistema, en este caso el proceso que invoca la llamada, debe recibir un mensaje indicando que el usuario solicitado no está conectado al sistema en ese momento. Una vez establecida esta relación, ambos usuarios serán "amigos" y podrán enviarse mensajes (en ambas direcciones)

1. **Group ID1, ID2, ID3…IDN: Esta operación permite crear un grupo con usuarios "amigos".** El grupo creado debe registrarse en el Manager. Todos los miembros del grupo deben ser amigos y estar conectados actualmente o haber estados conectados en forma previa; de lo contrario la operación devuelve error. La operación devuelve un identificador del grupo creado a quien la invoca y envía este identificador de forma asíncrona a todos los miembros del nuevo grupo. Este identificador debe ser único en el sistema. Una vez que se establece un grupo, cualquiera de los usuarios puede enviar un mensaje a sus integrantes. <u>Pueden suponer que ningún talker va solicitar la creación de un grupo ya existente.</u>
2. **Sent msg IDi:** Permite enviar un mensaje corto (máximo 100 caracteres) al "amigo" con identificador = IDi. Los amigos deben ver el mensaje de forma inmediata**.** El sistema da un error si IDi no es amigo de ID o, si siendo amigo, está desconectado. Este mensaje se debe enviar a través del Manager. Los mensajes se envían a los destinatarios, no se almacenan en el Manager.

3. **Sent msg GroupID:** permite enviar un mensaje a todos los miembros del grupo con identificador GroupID. El programa da error sólo si el grupo no existe. Es posible que algunos miembros del grupo estén desconectados. Los miembros desconectados pierden el mensaje, este no se almacena en ninguna parte.
4. **Salir.** Con esta opción un talker se desconecta del sistema**.** El usuario puede volver a conectarse con el mismo ID.

Note que mientras el talker se ejecuta atendiendo las peticiones del usuario, puede llegar un mensaje del Manager. Los tipos de mensajes que pueden llegar son: solicitud de establecimiento de una relación, formación de un grupo, mensaje que envía un amigo o mensaje que se envía a un grupo. Este mensaje se muestra por la cónsola y puede interrumpir la interacción actual del usuario y el talker por un momento; una vez que se recibe el mensaje, las operaciones deben continuar. **La atención de los mensajes asíncronos en el talker se implementa a través de señales (signals).**

El Manager

Es el proceso encargado de gestionar las relaciones entre usuarios, los grupos y el envío de **mensajes** entre usuarios, ya sea en forma directa, entre un usuario y otro, o de un usuario a un grupo. El proceso **Manager** se invocará de la siguiente forma desde el *shell:*

$ ***Manager N pipeNom***

Donde:

N: es el número máximo de usuarios que estarán conectados simultáneamente en el sistema. Puede suponer que 100 será el máximo N con el que se invocará este programa.

pipeNom: es el nombre del pipe nominal por donde los procesos talkers se conectarán inicialmente al proceso Manager.

El Manager debe imprimir por la cónsola todas las operaciones que va realizando para los talkers.

Funcionamiento del sistema y comunicaciones

Al comenzar su ejecución, todo talker debe registrarse con el **Manager**. Una vez registrado, el talker ejecutará un ciclo, en el cual le mostrará el prompt al usuario, para que éste solicite las distintas operaciones ya explicadas. Este programa termina cuando el usuario indique la opción de salir. El Manager debe estar al tanto de todos los usuarios conectados, las relaciones existentes y los grupos creados. El pipe ***pipeNom***, que reciben los dos tipos de procesos al ser inicializados, servirá de comunicación entre los talkers y el Manager. Un solo pipe es suficiente en este sentido. Se recomienda crear también un pipe entre el Manager y cada uno de los talker, como lo muestra la figura 1, para los mensajes que envíe el Manager al talker. Uds. pueden crear otros pipes entre los procesos si su diseño lo requiere. Las decisiones de diseño deben ser justificadas por cualquier

integrante del grupo cuando se evalúe el proyecto. Use dibujos para dar mayor claridad a los conceptos. El envío de mensajes entre procesos talkers (de forma directa o grupal) debe hacerse a través del Manager, como lo muestra la figura 2.

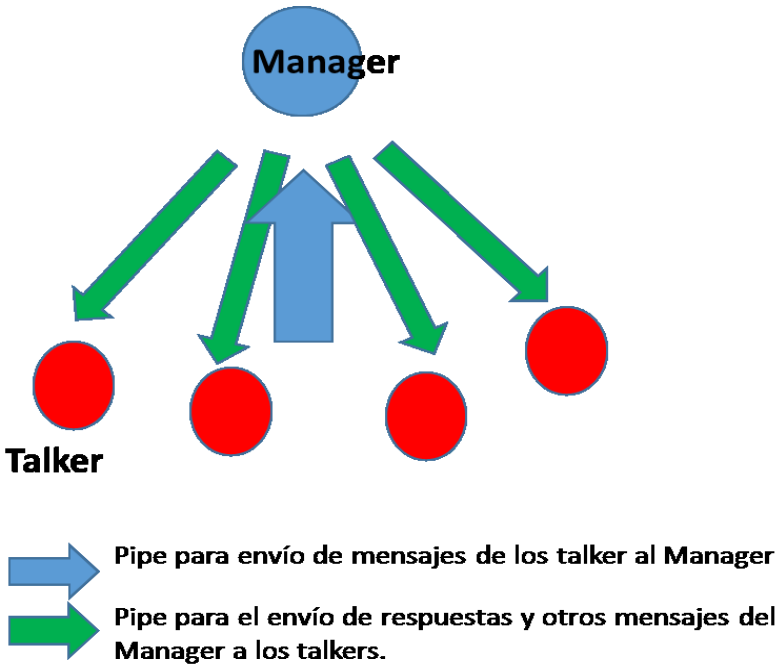

Figura 1: Pipes para las comunicaciones.

Finalmente, noten que un proceso talker está pendiente de las opciones del usuario y no de los mensajes que van llegando del Manager. La recepción asíncrona de los mensajes que envía el Manager a los talkers se puede implementar con una señal (*signal*) del proceso Manager al talker, avisando que tiene un nuevo mensaje para él.

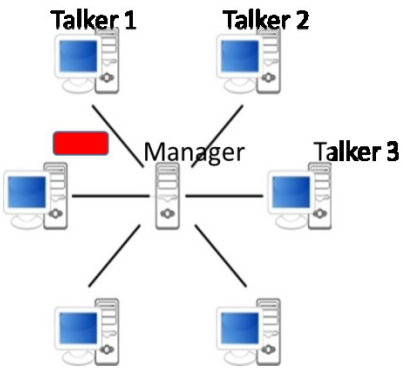

Figura 2a: Envío de un mensaje del talker1 al manager. El manager reenviará el mensaje a los talker 2 y 3

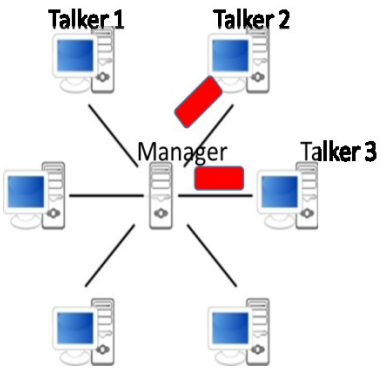

Figura 2b: re-envío del mensaje del talker 1 a los talker2 y talker 3.

Ejemplo de funcionamiento

A continuación presentamos un ejemplo de ejecución de Chattering. En la primera columna mostramos lo que se está mostrando en el resto de las columnas: ejecución de programas, solicitud de comandos o salida de los programas. **En rojo se destacan salidas que son producto de un mensaje asíncrono**. El orden de las operaciones que realiza el Manager dependerán del orden de llegada de las peticiones. Los mensajes asíncronos pueden salir de cualquier manera en la cónsola de los talkers.

Comandos	**$Manager 10 pipecom**	**$Talker 1 pipecom**	**$Talker 2 pipecom**	**$Talker 5 pipecom**
Salidas	Manager iniciado y el sistema podrá tener como máximo 10 usuarios.	Se inicializa un talker con ID = 1	Se inicializa un talker con ID = 2	Se inicializa un talker con ID = 5
Comandos		Rel 4	Rel 5	
Salidas	Talker 1 desea establecer relación con Talker 4. Talker 4 no está conectado. Se devuelve error. Talker 2 desea establecer relación con talker 5. Relación establecida en el Manager.			
Salidas		Error: el Talker 4, no existe en el sistema	Se establece una relación entre los Talkers 2 y 5	Se establece una relación entre los Talkers 2 y 5
Comandos		List	Rel 1	Rel 1
Salida	Talker 1 solicita los usuarios conectados. Talker 2 quiere establecer una relación con el Talker 1. Relación establecida en el Manager. Talker 5 quiere establecer una relación con el Talker 1. Relación establecida en el Manager.			

		Los usuarios actualmente en el sistema son 1,2,5 Se establece una relación entre el Talker 2 y el 1 Se establece una relación entre el Talker 1 y el 5	Se establece una relación entre el Talker 2 y el 1	Se establece una relación entre el Talker 1 y el 5
			Group 1, 5	Sent “Hola como estas” 1
	Talker 2 solicita creación de un grupo con amigos 1 y 5. Se crea un grupo con identificador G1 Talker 5 envía un mensaje al Talker 1			
		Talker 5 envía: “Hola como estas” Talker forma parte del grupo G1	Se gorma el grupo G1 con integrantes 1,2 y 5.	Talker forma parte del grupo G1
			Sent “hey” G1	
	Talker 2 envía “hey” a G1			
		“hey” viene de G1	“hey” viene de G1	“hey” viene de G1

Consideraciones de Diseño e Implementación

- Tenga en cuenta que el Manager y los procesos Talker son independientes. Se comunicarán **a través de pipes nominales y señales. Sólo se ejecuta un Manager, pero pueden existir varios Talker.**

- El proceso **Manager** y cada proceso **Talker** se inician de la forma indicada y con los parámetros señalados. Una vez que el **Talker** esté en funcionamiento, permitirá a los usuarios realizar las operaciones descritas.
- El **Manager** debe ir informando por pantalla sobre las operaciones que está realizando.
- El proyecto lo deben realizar en lenguaje C.

Sobre la Entrega

- El proyecto lo deben realizar en grupos de **como máximo dos estudiantes**.
- Lo deben entregar y sustentar el jueves de la semana 17 (**15 de noviembre de 2018**). La entrega, se realizará por UVirtual (antes de las 2 de la tarde del día 15 de noviembre, el día 16 para la sección que ve clases los viernes). Antes de hacer la sustentación se revisará que los estudiantes hayan subido el código a UVirtual. El proyecto debe colocarse en el aula virtual de forma comprimida usando el comando desde la consola de linux tar czvf (http://ecapy.com/comprimir-y-descomprimir-tgz-tar-gz-y-zip-por-linea-de-comandos-en-linux/index.html), los archivos empleados estarán dentro de un único directorio que tendrá como nombre proyectoII, más cuatro letras que corresponderán a los nombres y apellidos de los integrantes de grupo, no deben existir más directorios o serán penalizados.
- La entrega consiste de los códigos fuente, el makefile y durante la evaluación todos los integrantes del grupo deben poder responder los siguientes aspectos:
    - Problema planteado.
    - Estructuras de datos más importantes para almacenar las relaciones entre usuarios e información de los grupos. Deben describir estructuras de datos y presentarlas de forma gráfica.
    - Algoritmos que considere importantes.
    - Cómo es la comunicación entre procesos: cuantos pipes y qué mensajes se utilizaron entre los Talker y el Manager. Formato de los mensajes y para qué se usan. ¿Se implementó algún protocolo sencillo? Cómo se implementó la comunicación asíncrona.
- 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 las entradas del usuario.
- El día de la sustentación deberán realizar, individualmente, una modificación al proyecto.
- **El proyecto debe poder ejecutarse en las computadoras de la Universidad, de no ser así se considerará que el proyecto no funciona**.
- **Recuerden**: Para este proyecto no se otorga prórroga porque se necesita al menos una semana para calificarlo y la semana siguiente se deben entregar las notas finales.

Deben seguir todas las especificaciones que este enunciado. Si tiene alguna duda diríjase al Prof. Ricardo González.

Suerte!