

1er Proyecto: Cliente-Servidor con sockets

Descripción general

Para mejorar el sistema de distribución de gasolina PDVSA a pedido a los estudiantes de Redes de Computación que implementen un prototipo en lenguaje C y usando sockets.

El sistema tendrá bombas de gasolina (clientes) y centros de distribución de combustible (servidores). Las bombas pedirán a los centros de distribución una gandola de combustible para reponer sus inventarios.

Bombas de gasolina (Clientes):

Para simplificar en el prototipo inicial las bombas dispensaran solo un tipo de combustible (gasolina 95). Para cada una de ellas se tendrá su capacidad máxima de combustible, el inventario inicial y el consumo promedio por minuto en litros.

Centros de distribución (Servidores):

Igualmente que en el caso de las bombas se considerara que solo tienen un tipo de combustible. Todas las gandas disponibles tienen una capacidad de 38.000 litros para el despacho de la gasolina a las bombas.

Para cada centro se tendrá su capacidad máxima de combustible, el inventario inicial, tiempo de respuesta en minutos (tiempo que tarda la gandola en llegar a la gasolinera) y la tasa suministro de la refinería al centro de distribución en litros por minuto.

Funcionamiento del sistema

El prototipo debe simular el funcionamiento del sistema con los parámetros dados por 8 horas ($8 \times 60 = 480$ minutos), para que el tiempo de ejecución no sea excesivo se simulará cada minuto por 0,10 segundos (*Sleep(100)* o *usleep(100*1000)*).

- 1) Poner en funcionamiento los clientes y los servidores
- 2) Cuando una bomba detecte que dispondrá de capacidad libre suficiente (38.000 litros) según el tiempo de respuesta mínimo del los centros de distribución. Se pedirá a este centro (el que tenga tiempo mínimo), en el caso que este no responda o no tenga disponibilidad se hará en orden creciente según tiempo de respuesta.

- 3) Reportar en un fichero (log_nombrecliente.txt o log_nombreservidor.txt) los eventos relevantes de la simulación. Tanto para los clientes como los servidores.

Eventos relevantes (Cliente = C y Servidor = S):

-Estado inicial (C y S): Inventario

-Tanque full (C y S): Tiempo

-Tanque vacío (C y S): Tiempo

-Petición (C): Tiempo, nombre del centro de suministro, resultado de la petición

-Suministro (S): Tiempo, nombre de la bomba que pide, resultado de la petición, inventario

-Llegada de la gandola (C): Tiempo, inventario

Consideraciones:

- Durante el funcionamiento puede ser que los centros de distribución no estén siempre operativos. No responden al "connect" o responden con "No disponible".
- En el caso que un centro de distribución no cuente con inventarios suficientes para despachar una gandola debe notificarlo a la bomba, esta debe hacer un pedido al próximo centro con menor tiempo de respuesta.
- Suponga que tanto las bombas como los centros de distribución trabajan de forma continua.

Implementación Cliente-Servidor

Cliente

El cliente se invoca con el comando

>bomba -n nombreBomba -cp capacidadMaxima -i inventario -c consumo -fc FicheroCentros

Donde:

-n indica el nombre de la bomba (cadena de caracteres sin espacios)

-cp capacidad máxima en litros (entero entre 38.000 y 380.000)

-i inventario actual (entero entre 0 y capacidad máxima)

-c consumo promedio en litros por minuto de operación (entero entre 0 y 1.000)

-fc nombre del fichero que contiene en cada línea el nombre de un centro de distribución así como las direcciones (DNS) de la máquina donde corre su servicio y el número de puerto en el que corre el servicio en esa máquina, separados entre sí por un "&".

Ejemplo de archivo centro de distribución:

El_Picoteo&korba ldc.usb.ve&23562

La_Pala&lala ldc.usb.ve&15630

Nota: Tamaño máximo de 40 caracteres tanto para el nombre del centro de distribución como para la dirección de la máquina.

Para evitar conflictos, se usarán los últimos cinco dígitos del carnet de los integrantes del equipo como número de puerto. Si su últimos cinco dígitos son superiores a 65535, simplemente substituyan el primero por 1 o 2. En algunas máquinas trabajar con puertos superiores a treinta mil, aun siendo validos dan problemas, en ese caso también substituyan el primer número por 1 o 2.

Servidor

El servidor se invoca con el comando

>centro -n nombreCentro -cp capacidadMaxima -i inventario -t tiempo -s suministro -p puerto

Donde:

-n indica el nombre del centro de distribución (cadena de caracteres sin espacios)

-cp capacidad máxima en litros (entero entre 38.000 y 3.800.000)

-i inventario actual (entero entre 0 y capacidad máxima)

-t tiempo de respuesta en minutos (entero entre 0 y 180)

-s suministro promedio en litros por minuto (entero entre 0 y 10.000)

-p puerto por el cual escuchara peticiones

Evaluación

- Este proyecto corresponde a 15% de la nota del curso. El proyecto que se entregue debe funcionar correctamente. Si no corre no será evaluado.
- Habrá un punto de control, sin nota pero obligatorio. Aquellos grupos que no entreguen este avance del proyecto no les será corregido el proyecto final.
- Las especificaciones de la entrega:
 - Deberán entregar por aula virtual un tar.gz con el código y un makefile. Los profesores descomprimirán su contenido un directorio vacío para cada grupo del LDC y ejecutarán el make. El proyecto no debe requerir ninguna otra configuración adicional para su funcionamiento.
 - También deberán entregar un **LEEME.txt** con las particularidades de su proyecto. Indicando por ejemplo: Casos en los que falla, extensiones que hayan hecho al mismo y cualquier otro detalle relevante que consideren de interés. **Para más detalles sobre su contenido vean el documento “Especificación de Entrega de Proyectos” disponible en aula virtual.**
 - El código debe estar correctamente documentado. Si los profesores no pueden comprender que hace su código serán penalizados. Deben seguir las buenas prácticas de programación en C disponibles en aula virtual.
 - Por último deberán entregar en el casillero del departamento de computación del **profesor Clotet** la declaración de honestidad. Para facilitar las cosas pueden obtenerla directamente de google drive:

https://docs.google.com/document/d/1ioDoq_YFreFJ2Xi-ZJjIVsOzHinjQ38kMTqU_RJyQc8/edit?pli=1

- Para su evaluación el proyecto será probado con un conjunto de datos que serán establecidos en el momento de la corrección, no necesariamente los que fueron suministrados como ejemplo.
- Otros detalles de la entrega, así como archivos de prueba, serán publicados oportunamente.
- El programa debe correr en las máquinas del LDC. Específicamente en la sala 221-222 (Debian 6 ver 2.6.32). Todos los proyectos deben funcionar en las computadoras de esa sala. Disponen de más de 20 equipos, suficientes para que todos los estudiantes, donde pueden acceder de manera presencial o por ssh. Para asegurarse que están trabajando con los equipos de esta sala en vez de usar los nombres usen los alias cuando usen ssh: m\$s-222 ldc.usb.ve, donde \$ debe ser substituido por el número de equipo, del 1 al 20.

Observaciones

- La implementación se hará con sockets, usando TCP. **Pueden usar diferentes conexiones durante la ejecución, pero solo una por petición, salvo error en la misma.**

- **Recuerde que los servidores deben soportar obligatoriamente la concurrencia. Pueden usar cualquier cualquiera de las formas vistas en materias anteriores, les recomiendo revisar Operativos 1. Estamos en Redes 1 y vamos a evaluar principalmente las conexiones, cualquier implementación es válida siempre que la hagan de la forma correcta, documentada y justifiquen sus decisiones en el LEEME.txt**
- Los formatos, nombres de los comandos, etc., deberán ser estrictamente los especificados aquí.
- Todos los parámetros estarán siempre presentes, pero pueden estar en diferente orden que el indicado.
- Por simplicidad de los proyectos se pide que no se implemente ningún tipo de interfaz gráfica, que se emplee la línea de comandos para invocar a los programas y la salida estándar para cualquier información que se desee reportar.
- El no seguir las instrucciones será penalizado.
- Para efectos de desarrollo puede utilizarse una máquina para correr todos los componentes (clientes y servidores). Una vez funcione en una máquina, debe probarlo en al menos dos máquina para verificar que funcione bien de forma remota. La corrección se hará usando varias máquinas.
- Se recomienda el uso de gdb para debug.
- Los grupos serán de 2 estudiantes, pudiendo ser de diferentes secciones.

Proyecto 1 de Redes de Computadoras

Enero - Marzo 2013

Declaramos que entendemos que la honestidad es uno de los valores fundamentales de la Universidad Simón Bolívar y que el plagio o la copia en cualquier evaluación constituyen una falta de probidad en el ejercicio de nuestras obligaciones como alumnos. Sabemos que esta falta puede ser severamente sancionada según el Reglamento de Sanciones y Procedimientos Disciplinarios. Asimismo declaramos que el trabajo contenido en esta entrega ha sido realizado solamente por los suscritos miembros del grupo.

Carnet	Nombre y Apellido	Firma

Disponible en: https://docs.google.com/document/d/1ioDoq_YFreFJ2Xi-ZJjIVsOzHinjq38kMTqU_RJyQc8/edit?pli=1