



Universidad Simón Bolívar
Departamento de Computación y TI
Redes de Computadoras I (CI-4835)
Trimestre Abril-Julio 2016

Asignación N° 3

Objetivo General: Proveer al estudiante de un trabajo práctico de diseño y construcción de una aplicación final, que use el paradigma de programación Cliente – Servidor, para que este comprenda, en general, el funcionamiento simple de aplicaciones y servicios en redes.

Objetivos Específicos: Al finalizar esta actividad el estudiante deberá estar en capacidad de:

- Comprender el uso y la programación de la Interfaz de Aplicaciones (API) Sockets de Berkeley.
- Aplicar los conceptos dados en clases para desarrollar el diseño e implementación de un protocolo de comunicación básico.

Planteamiento del Problema

El Centro Comercial **Moriah** dispone de un estacionamiento de 200 puestos con tres puertas. Cada puerta permite la entrada y salida de vehículos y está dotada de un botón pulsador y un expendedor de tickets. Además dispone de un letrero luminoso donde se puede escribir:

- Estacionamiento *con puestos*
- Estacionamiento sin puestos o

Se quiere diseñar un sistema informático basado en el paradigma Cliente/Servidor que permita automatizar y controlar dicho estacionamiento. Para ello se le ha contactado a usted como estudiante del curso de Redes de Computadores 1, a fin de que proponga e implemente una aplicación de red sencilla que incluya un computador central (CC) que va a controlar el sistema completo junto a los equipos de las puertas que permiten el acceso o salida de vehículos, tal y como se muestra en el Figura 1.

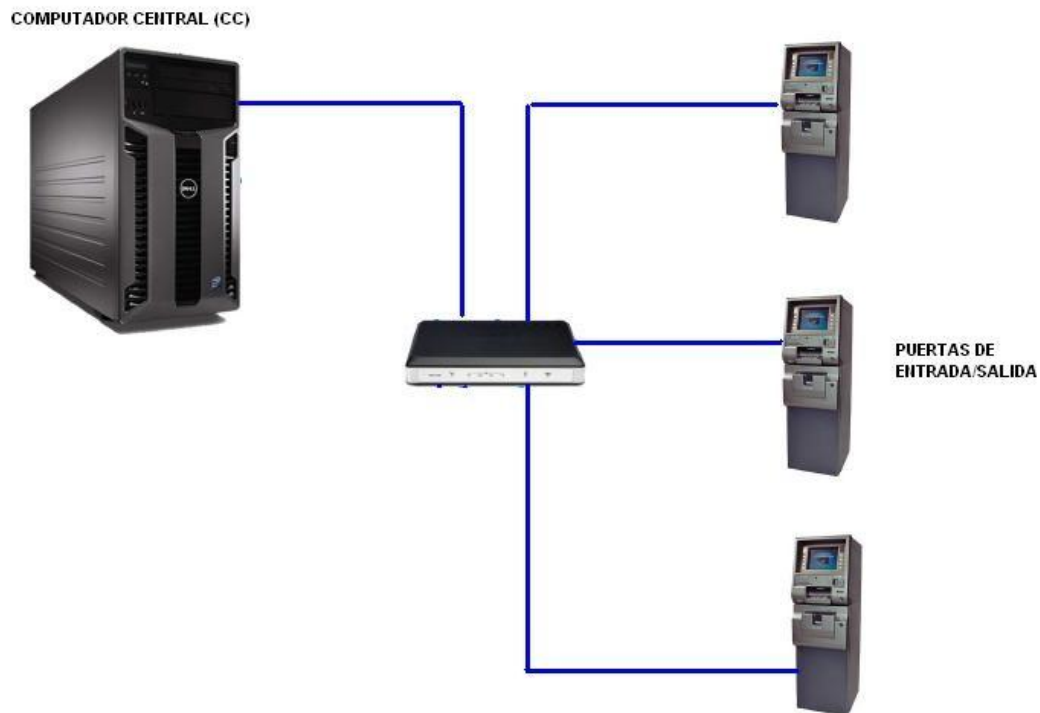


Figura 1. Red del estacionamiento del Centro Comercial Moriah

El funcionamiento del Sistema de Estacionamiento Moriah (SEM) es el siguiente:

- Cuando un vehículo llega a una puerta el conductor presiona el pulsador. Si el estacionamiento tiene puestos disponibles, el equipo de la puerta envía al computador central un mensaje que indica que un vehículo quiere entrar. Si el estacionamiento está sin puestos, el CC envía un mensaje informando que el vehículo no puede pasar (de esta forma el computador de la puerta no imprime el ticket de entrada).
- Cuando el CC recibe un mensaje de entrada incrementa el contador de puestos ocupados y responde al computador de la puerta adecuada con un mensaje indicando la hora y fecha de entrada, un código (un número entero) que identifica al vehículo. Con la fecha, hora y código, el computador de la puerta imprime el ticket.
- Cuando un conductor desea salir del estacionamiento se dirige hacia alguna de las puertas e introduce el ticket de salida. El CC realiza el cálculo del importe a pagar y le indica al conductor el precio a cancelar para que se proceda a la retirada del vehículo. Cada vez que un vehículo sale, el CC debe decrementar el contador de puestos. Si el estacionamiento pasa a estar con puestos *libres* se informa al conductor siguiente de dicho evento.

- Considere que la primera hora de estacionamiento cuesta 80 Bs y la fracción (1 hora) cuesta 30 Bs.
- Al cierre del día el Computador Central debe tener un registro de todas las operaciones realizadas (entradas/salidas) indicando fecha, hora, código del vehículo, el evento (entrada o salida del vehículo), monto pagado.
- Considere que pueden haber solicitudes del servicio simultáneos.
- El estacionamiento funciona 24 horas al día.

Sintaxis de la invocación de comandos:

La sintaxis para la invocación de la ejecución del módulo que ofrece el servicio del CC deberá ajustarse a la siguiente indicación:

sem_svr -l <puerto_sem_svr> -i <bitácora_entrada> - o <bitácora_salida>

Dónde:

<puerto_sem_svr> Es el número de puerto local en que el computador central ofrecerá el servicio

<bitácora_entrada> Es el nombre y dirección relativa o absoluta de un archivo de texto que almacena las operaciones de entrada al estacionamiento.

<bitácora_salida> Es el nombre y dirección relativa o absoluta de un archivo de texto que almacena las operaciones de salida del estacionamiento.

La sintaxis del módulo que ejecutan los equipos de las puertas:

sem_cli -d <nombre_módulo_atención> -p <puerto_sem_svr> - c <op> -i <identificación_vehículo>

Donde:

<nombre_módulo_atención>: Es el nombre de dominio o la dirección IP (version 4) del equipo en donde se deberá ejecutar el módulo de atención centralizada.

<puerto_sem_svr>: Es el número de puerto remoto en que el módulo de servicio atenderá la comunicación solicitada.

<op>: Indica si el vehículo va a entrar o salir del estacionamiento, puede tener dos valores **e** ó **s**

<identificación_vehículo> Es un número serial que identifica únivocamente un vehículo

Diseño del protocolo de comunicación:

Para la definición del protocolo se deben tener en cuenta los siguientes aspectos y como parte de la documentación se deberá describir en forma clara, cada uno de ellos:

1. ¿Qué tipo de sockets decidió emplear?. Justifique su respuesta.
2. Identifique todos los mensajes del sistema, indicando: el formato del mismo, su tamaño en bytes, quién genera el mensaje y quién lo recibe y procesa.
3. Realice el diseño completo del protocolo de comunicación que construya y describa como opera el mismo, debe usar UDP como capa de transporte. Incluya el punto de vista de los cambios de estado de las entidades que se comunican. Se sugiere usar diagramas de máquinas de estados finitos y facilitar la explicación de como es la actividad del protocolo.
4. Describa aspectos del proyecto que funcionan según el enunciado y cuales no. Cualquier requerimiento no desarrollado o que contenga fallas, deberá ser señalado claramente.
5. La implementación es en lenguaje C.
6. Opcional: Construya un aplicación confiable, que garantice la entrega, evite información duplicada y pérdida de información. Todos estos controles deben estar en capa de aplicación.

Esta sección debe imprimirse y será entregada el lunes de semana 12 a los profesores del laboratorio, o en su defecto, puede dejarlo en el casillero ubicado en el departamento de computación y TI, de 8:30 a.m. a 3:45 p.m. Sea organizado en el documento que será entregado, incluya portada, contenido, descripción del problema a resolver, entre otros; se evaluará la presentación y estructura.

Consideraciones Adicionales:

- Durante el funcionamiento del sistema puede ser que el servidor no esté siempre operativo. En ese caso el cliente hará la petición hasta 3 veces, si el servidor no le responde, entonces enviará un mensaje a la pantalla indicando que el tiempo de repuesta se agotó.
- El servidor deberá operar en forma continua.

- Todos los parámetros estarán siempre presentes, pero pueden ser invocados en un orden diferente al que se indicó en la sintaxis. Aún así, el sistema deberá operar correctamente.
- Todo lo que usted asuma justifíquelo adecuadamente.

Plataforma donde será evaluado su trabajo:

- Los proyectos serán evaluados en los equipos del Laboratorio de Computación (LDC).
- Para las pruebas de su proyecto utilice como número de puertos uno de 5 cifras que comience por 2 y que las siguientes 4 cifras coincidan con las últimas cuatro cifras del número de carnet de alguno de los integrantes del equipo. Si se requiere más de un puerto, repita el procedimiento pero con el número de carnet de otro integrante.

La entrega de la aplicación debe hacerse de manera electrónica a través del Aula Virtual, en la fecha y hora indicada: hasta el lunes de semana 12 a las 11:55 p.m. El acceso para subir los mini proyectos será cerrado luego de la hora establecida, por lo tanto se recomienda no esperar hasta la última hora para subir los archivos.

La entrega será con un archivo comprimido que incluya toda la documentación y los programas y se identificará de la siguiente manera: Apellido1_Apellido2_X.tar.gz

(ver <http://ldc.usb.ve/~figueira/Cursos/ci3825/taller/material/TGZ.html> de cómo hacer un archivo tar.gz), donde X será reemplazado por el término Sockets y Apellido1 y Apellido2 corresponden a los apellidos de los integrantes del grupo (por ejemplo, Mora_Azuaje_Sockets.tar.gz) con los siguientes archivos:

- Archivos fuente. Por ejemplo, si se trata de un proyecto en lenguaje C, debe incluir todos los ".c" y ".h".
- Makefile
- Un archivo LEEME.txt, que explique el contenido del "tar.gz":

Tenga cuidado de no incluir ejecutables, archivos objeto (.o) o cualquier otro archivo generado automáticamente a partir de los fuentes.

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 máquinas para verificar que funcione bien de forma remota. La corrección se hará usando varias máquinas.

Código:

El código deberá incluir un documento del tipo "Leeme.txt" con al menos los siguientes elementos:

- nombres y apellidos de los integrantes del grupo
- números de carnet
- qué archivos lo componen y qué tiene cada archivo
- qué hace el programa
- cómo se ejecuta
- que condiciones particulares tiene, por ejemplo la entrada XX tiene máximo 50 caracteres" o "el máximo número de registros YY es 20"

Por su parte, el código deberá estar debidamente documentado, siguiendo los estándares de documentación de Javadoc con el programa y los encabezados de las funciones implementadas. Adicionalmente, todos sus programas deben seguir las buenas prácticas de estilo de programación en C y todas las llamadas al sistema deben ser correctamente manejadas.

Condiciones de la entrega.

La entrega se deberá hacer dentro del lapso indicado. Cualquier demora podrá dar la potestad al docente de no aceptar el trabajo. Los informes deberán ser impresos además y se harán llegar según indique el profesor de turno. Cualquier falla de versión inapropiada, falta de algún elemento u otra situación anómala, podrá dar potestad al docente de colocar cero (0) como calificación. Recuerde, el docente no está obligado a ajustar o adaptar su sistema. Usted debe preparar todo para facilitar la ejecución de su Proyecto sin esfuerzos. No piense que tendrá otra oportunidad para arreglar cualquier falla.

Los equipos deben ser de dos (2) estudiantes, en caso de ser necesario sus integrantes deberán asegurarse de poder explicar, a cabalidad, la distribución del trabajo de cada miembro integrante. Además, cada miembro del equipo debe estar en capacidad de comprender y desarrollar cualquier parte del trabajo, incluso aunque no le haya sido asignada originalmente. Si estas condiciones no se cumplen, el evaluador del proyecto podrá reprobar al integrante que no cumpla con las mismas.

Cualquier caso de plagio será severamente castigado, no será evaluado ninguno de los proyectos involucrados y serán aplicadas las sanciones correspondientes establecidas en los reglamentos de la universidad.