PROYECTO FINAL

Objetivos

General

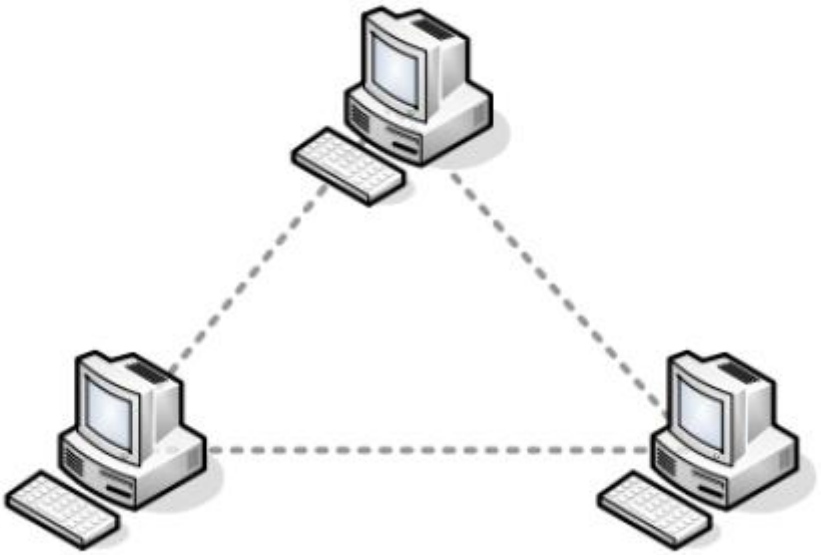
- Que el estudiante ponga en práctica los conocimientos del curso, así como su espíritu investigativo que le ayude a obtener los conocimientos que le puedan servir en su vida de profesional y desarrollar la capacidad para desarrollar y tomar decisiones en el diseño de algoritmos.

Específicos

- Manejo de semáforos y colas de mensajes del sistema operativo (Linux).
- Manejo concurrencia de procesos, y algoritmos de planificación de recursos.

Descripción:

El proyecto consiste en la creación de un protocolo de comunicación. Para la creación de un protocolo de red se deben solucionar varios problemas de concurrencia, el objetivo es que el estudiante ponga en práctica todos los conocimientos adquiridos en el curso para resolver los problemas que pueda encontrar para establecer la comunicación entre varias computadoras, enviar y recibir paquetes entre sí.



El proyecto consiste en comunicar N computadoras (N = 3) por medio del puerto serial, para lo cual se debe de implementar un protocolo de comunicación entre las computadoras. Se enviaran y recibirán paquetes conteniendo información de archivos de texto. Cada paquete tiene un tamaño máximo de 1 Kb. El paquete es el mensaje que envía a una computadora a otra con el objetivo de intercambiar archivos. En general cualquier computadora puede manejar o tener acceso a los archivos o recursos (impresora) de las demás computadoras. Como lo es normal en los protocolos de comunicación si el archivo a compartir tiene un tamaño mayor al de la capacidad del paquete, este archivo debe ser segmentado y enviado en una serie de paquetes.

La transmisión de la información se realiza por medio del puerto serial, la información se transmite por una única línea en forma consecutiva. En este tipo, de transmisión los bits que forman un byte o palabra, se transmiten sucesivamente uno detrás del otro. La información le llega al receptor bit a bit. Además la comunicación entre cada computadora es simplex, la información se transmite en una sola dirección. Los enlaces están dedicados a la transmisión en un solo sentido. En esta comunicación están perfectamente definidas las funciones del emisor y el receptor, y la transmisión de datos siempre se efectúa en una dirección: emisor - receptor. En este tipo de comunicación se dice que hay un único canal físico y un único canal lógico unidireccional.

El proyecto debe ser desarrollado en grupo no mayores de 3 personas. Queda a discreción del estudiante el diseño de los paquetes, por lo tanto 2 o más grupos no pueden tener el mismo formato del protocolo.

Funcionamiento:

- El envío y la recepción de paquetes, así como su manipulación, deben de ser sincronizados para su correcto funcionamiento.
- Deben existir Colas de espera para los paquetes que recibe o envía cada computadora.
- Cada computadora podrá enviar paquetes a cualquier computadora de la red, de igual forma todas las computadoras podrán recibir paquetes de cualquier computadora.
- Para facilitar el traslado de información se manejarán únicamente archivos de texto plano.
- En cada computadora debe existir una consola o terminal. Una consola consiste en una interfaz grafica con el Usuario donde ingrese el comando con sus parámetros correspondientes, además debe de existir un área de la consola para el receptor y emisor en la cual solo se mostrara la información que pasa por esos procesos, se muestra el paquete y los datos más

importantes, la información mostrada debe de ser entendible para cualquier usuario.

Comandos:

1. **Send:** Permite enviar un mensaje de texto a un usuario desde la consola hacia una computadora X. Ejemplo:

   #send “mensaje” to PC1

   El tamaño máximo del mensaje es de 500 caracteres.

2. **Copy:** Permite copiar un archivo de una computadora X hacia una computadora Y.

   #copy PC1 archivo_origen.txt to PC2 archivo_destino.txt

   Si el archivo de destino ya existe se debe agregar el parámetro –r para indicar que se debe de sobrescribir de lo contrario no se reemplazará el archivo de destino.

3. **Type:** Permite mostrar en consola el contenido de un archivo en una computadora X.

   #type PC1 archivo_origen.txt to PC2

4. **Print:** Permite Imprimir un documento de una computadora X en una computadora Y.

   #print PC1 archivo_origen.txt to PC2

5. **Dir:** Permite visualizar en consola el directorio de una computadora X de la red.

   #dir PC1 directorio to PC2

6. **add_rem:** Permite agregar texto remotamente a un archivo en la computadora Y, desde la computadora X.

   #add_rem PC_Destino archivo_ nativo.txt “texto a agregar”

7. **remove_rem:** Permite eliminar un archivo remotamente de una computadora Y, desde una computadora X.

   #remove_rem PC_Destino archivo_nativo.txt

Procesos del Sistema:

para garantizar el envío y la recepción de paquetes, así como su manipulación, en cada computadora se debe de ejecutar cada uno de los siguientes procesos

1. **Receptor:** Recibe todos los paquetes y realiza la acción que corresponda:
   - Enviar el paquete al proceso ejecutor
   - Enviar el paquete al proceso emisor en caso que el paquete no haya alcanzado la computadora destino.
2. **Emisor:** Proceso encargado de enviar el paquete a la maquina destino. Se utilizará una cola con capacidad máxima de 10 Kb.
3. **Editor:** Es la consola donde se ingresan los comandos explicados anteriormente, es la interfaz de texto entre el sistema y el usuario.
4. **Ejecutor:** Es el encargado de realizar la acción que indique el comando. Por ejemplo si un paquete llega a una computadora y según el protocolo este debe de imprimir un archivo, este proceso será el encargado de enviar el archivo a la impresora predeterminada.
5. **Otros:** Aquellos procesos que considere necesarios para garantizar el funcionamiento del sistema, deberá de explicarlos en los respectivos manuales.

Consideraciones a tomar:

- Para interpretar los comandos no se hace diferencia entre mayúsculas o minúsculas.
- Para las opciones copy, type y print, si el archivo de origen no existe se debe mostrar un mensaje en la consola mostrando el error.
- Para la opción de add_rem, se puede agregar texto desde las otras dos terminales, por lo que deberá de garantizar el acceso.
- Para la opción de remove_rem, si el archivo está siendo utilizado se deberá de negar la eliminación del mismo.
- Todos los algoritmos que se utilizan en este proyecto deben ser documentados con su debida ejemplificación de funcionamiento.
- La conexión entre las computadoras deberá de realizarse por el puerto serial.

Documentación:

- Manual Técnico.
- Manual de Usuario.

Entregables [solamente digital]:

- Código Fuente, Ejecutables.
- Documentación

Notas importantes:

- El proyecto debe ser realizado en grupos no mayores de 3 personas.
- Copias de proyecto tiene nota de 0, se verificará código sospechoso, y el correspondiente reporte ante la Escuela de Ciencias y Sistemas quién se encargará de aplicar lo establecido en el reglamento.
- La calificación es presencial, y el día que se les indique, de no presentarse solamente se calificarán manuales. Cualquier ausencia debe justificarse con anticipación para tomar las consideraciones necesarias. Es obligatorio que todos los integrantes del grupo estén presentes durante la calificación.
- Puede utilizar el lenguaje de programación java o C.
- Todos los Semáforos, Colas de mensajes y procesos deben de poder visualizarse en consola, para garantizar el manejo de la exclusión mutua.
- Podrá trabajar sobre cualquier sistema operativo, tomando en cuenta que al menos un SO debe de ser de código abierto.

Fecha de entrega

28 de Mayo de 2011