

**Instituto Politécnico Nacional  
Escuela Superior de Cómputo**

Aplicaciones para comunicaciones de red

“Práctica 3”

Profesor: Rangel González Josué

Alumnos:

-Frausto Hernández Omar  
-Guzman Pérez Oscar  
-Castro Cruces Jorge Eduardo

Grupo: 3CV18

## INTRODUCCIÓN

Para esta práctica debemos de implementar una aplicación consistente en un cliente y un servidor, en el cual el cliente debe enviar archivos pdf, mp4, jpg mediante sockets al servidor el cual debe ser capaz de recibir y guardar cada uno de estos.

### Socket

Socket designa un concepto abstracto por el cual dos procesos (posiblemente situados en computadoras distintas) pueden intercambiar cualquier flujo de datos, generalmente de manera fiable y ordenada.

El término socket es también usado como el nombre de una interfaz de programación de aplicaciones (API) para la familia de protocolos de Internet TCP/IP, provista usualmente por el sistema operativo.

Los sockets de Internet constituyen el mecanismo para la entrega de paquetes de datos provenientes de la tarjeta de red a los procesos o hilos apropiados. Un socket queda definido por un par de direcciones IP local y remota, un protocolo de transporte y un par de números de puerto local y remoto.

Típicamente, un zócalo se rige por un flujo específico de eventos para que funcione. Para un modelo cliente-servidor orientado a la conexión, el socket en el proceso del servidor espera la petición de un cliente. Para hacer esto, el servidor necesita principalmente establecer una dirección que los clientes puedan usar para encontrar y conectarse al servidor. Cuando se establece una conexión con éxito, el servidor esperará a que los clientes soliciten un servicio. El intercambio de datos cliente-servidor tendrá lugar si el cliente se conecta al servidor a través del socket. El servidor responderá a la solicitud del cliente y enviará una respuesta.

La mayoría de las veces, las URL y sus conexiones se utilizan para acceder a Internet. Los programas requerirán un simple enlace de comunicación entre el cliente del programa y el servidor. Esta función está asociada a un socket que ayuda a vincular los extremos del cliente y del servidor del programa. Si un cliente comienza a establecer comunicación con el servidor, se realizará una conexión fiable entre el servidor y el cliente a través del canal de comunicación TCP. Con este tipo de comunicación, tanto el cliente como el servidor pueden leer o escribir en sockets que están conectados a un canal de comunicación específico.

## **Sincronización**

Hablamos de la sincronización de datos como el proceso de alineación entre los datos provenientes de diversas fuentes, y su continua armonización en el tiempo. Vale decir, la coordinación de procesos que se ejecutan simultáneamente, a fin de obtener un orden de ejecución correcto y evitar errores del sistema.

Esta acción permite obtener datos precisos y fidedignos de todos los sistemas, sean éstos operativos o transaccionales. Así, cada organización podrá crear una lógica propia para la sincronización de sus datos en tiempo real, si se requiere, obteniendo una mayor precisión, coherencia de la información y una mejora sustancial del rendimiento.

## **Memoria compartida**

La memoria compartida es la memoria física que se asigna a la agrupación de memoria compartida y se comparte entre varias particiones lógicas. El término agrupación de memoria compartida es una colección definida de bloques de memoria física que el hipervisor gestiona como una agrupación de memoria individual. Las particiones lógicas que configura para que utilicen memoria compartida (en adelante denominadas particiones de memoria compartida) comparten la memoria en la agrupación con otras particiones de memoria compartida.

## **Buffer de datos**

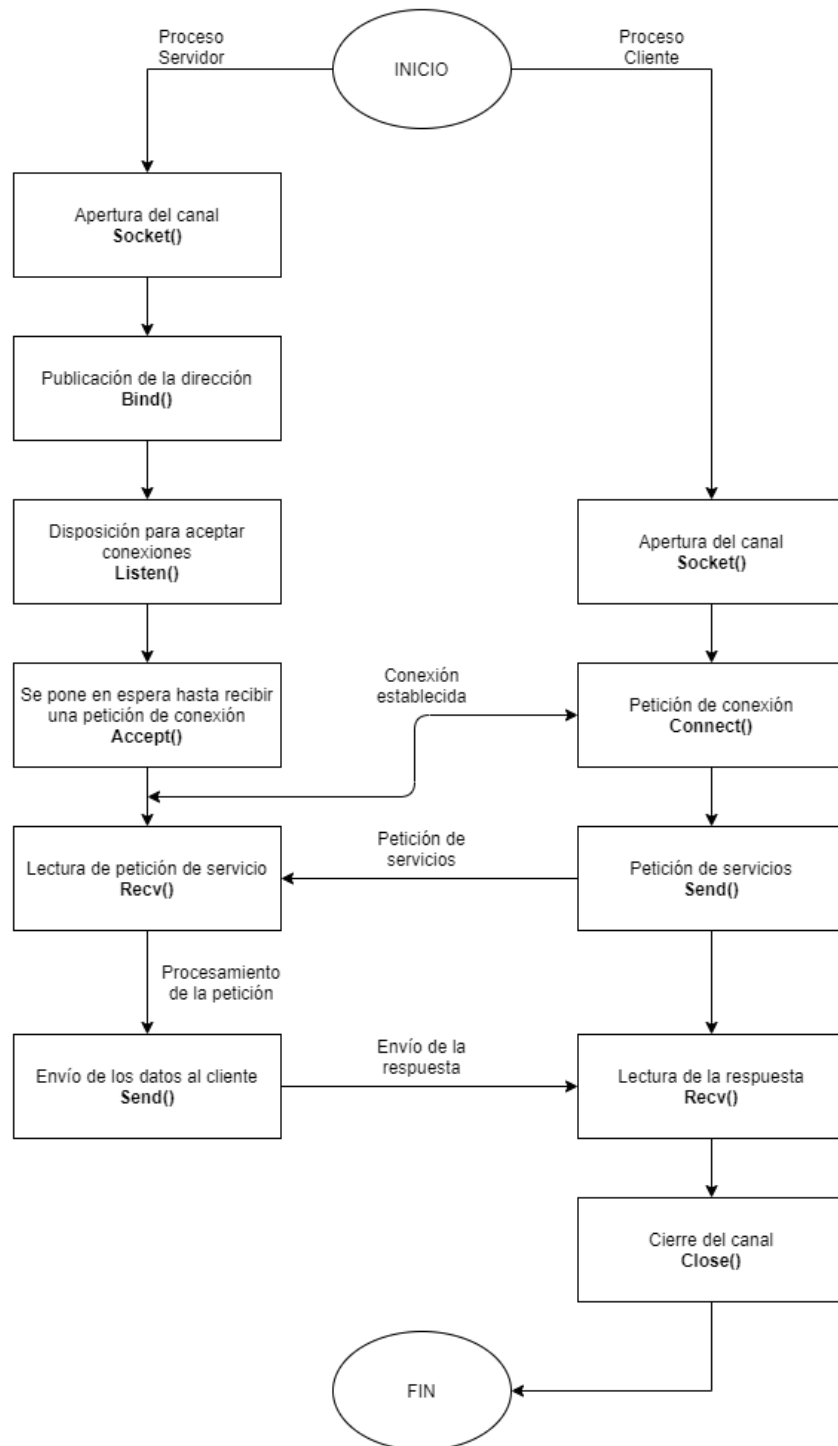
En computación, un buffer de datos es un espacio de memoria usado de forma temporal que se suele usar para mover datos de un lugar a otro, usualmente se utilizan en dispositivos de entrada y salida, procesos y redes, en general suelen ser utilizados cuando se está procesando información de forma continua como podría ser el caso de un sensor, un micrófono, o una cámara, o simplemente si no podemos procesar la información de una forma directa por su tamaño.

## **Arquitectura Cliente-Servidor**

En computación la arquitectura cliente-servidor es una forma de comunicar dos procesos o equipos a través de la red en donde el cliente realiza peticiones a un servidor y el servidor está a la espera de clientes para procesar sus peticiones, una de las ventajas es que un solo servidor puede atender a múltiples clientes a la vez; de la misma forma un sólo cliente puede hacer peticiones a más de un servidor a la vez para consumir diferentes servicios o acceder a diferente información.

# FUNCIONAMIENTO DEL PROGRAMA

- Diagrama de flujo



- **Algoritmo**

1. Abrir un canal para el servidor y otro para el cliente, mediante un `socket()`.
2. Publicar la dirección del socket del servidor, mediante `bind()`.
3. Aceptar las conexiones al servidor, mediante `listen()`.
4. Poner en estado de espera de solicitudes de conexión al servidor, mediante `accept()`.
5. Pedir conexión del cliente al servidor, mediante `connect()`.
6. Pedir servicio del cliente al servidor, mediante `send()`.
7. Leer la petición de servicio del cliente, mediante `recv()`.
8. Procesar la petición del cliente.
9. Contestar la petición del cliente, mediante `send()`.
10. Leer la respuesta del servidor, mediante `recv()`.
11. Cerrar el canal de comunicación, mediante `close()`.

- **Pruebas de ejecución**

Proceso Servidor esperando solicitudes:

```
Microsoft Windows [Versión 10.0.19042.867]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\georg>cd C:\Users\georg\Desktop\ESCOM\7mo. Semestre\Redes2\Practicas\3\APCR-main

C:\Users\georg\Desktop\ESCOM\7mo. Semestre\Redes2\Practicas\3\APCR-main>python FileServer.py
El servidor TCP está disponible y en espera de solicitudes
```

Proceso Cliente solicitando y recibiendo todos los archivos existentes en la carpeta indicada:

```
Microsoft Windows [Versión 10.0.19042.867]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\georg>cd C:\Users\georg\Desktop\ESCOM\7mo. Semestre\Redes2\Practicas\3\APCR-main

C:\Users\georg\Desktop\ESCOM\7mo. Semestre\Redes2\Practicas\3\APCR-main>python FileClient.py
Enviando archivo EduardoCastro.c...
el servidor confirmo que esta listo: Listo
El Servidor recibirá el archivo
Esperando confirmación del servidor
El servidor dice: exito al guardar
terminando
Enviando archivo Examen3.png...
el servidor confirmo que esta listo: Listo
El Servidor recibirá el archivo
Esperando confirmación del servidor
El servidor dice: exito al guardar
terminando
Enviando archivo Examen3.py...
el servidor confirmo que esta listo: Listo
El Servidor recibirá el archivo
Esperando confirmación del servidor
El servidor dice: exito al guardar
terminando
Enviando archivo multiples_secciones_criticas.c...
el servidor confirmo que esta listo: Listo
El Servidor recibirá el archivo
Esperando confirmación del servidor
El servidor dice: exito al guardar
terminando
Enviando archivo prod_cons.c...
el servidor confirmo que esta listo: Listo
El Servidor recibirá el archivo
Esperando confirmación del servidor
El servidor dice: exito al guardar
terminando
Enviando archivo Proyecto.c...
el servidor confirmo que esta listo: Listo
El Servidor recibirá el archivo
Esperando confirmación del servidor
El servidor dice: exito al guardar
terminando
Enviando archivo resultado.txt...
el servidor confirmo que esta listo: Listo
El Servidor recibirá el archivo
Esperando confirmación del servidor
El servidor dice: exito al guardar
terminando
Enviando archivo semaforo.c...
el servidor confirmo que esta listo: Listo

El Servidor recibirá el archivo
Esperando confirmación del servidor
El servidor dice: exito al guardar
terminando
Enviando archivo Semaforos.pdf...
el servidor confirmo que esta listo: Listo
El Servidor recibirá el archivo
Esperando confirmación del servidor
El servidor dice: exito al guardar
terminando

C:\Users\georg\Desktop\ESCOM\7mo. Semestre\Redes2\Practicas\3\APCR-main>
```

Proceso Servidor Enviando archivos solicitados al cliente:

```
Se guardo exitosamente el archivo, listo para recibir nuevo
cliente ('127.0.0.1', 56814)
Salvando Examen3.py
1024 de 3842
2048 de 3842
3072 de 3842
4096 de 3842
Se guardo exitosamente el archivo, listo para recibir nuevo
cliente ('127.0.0.1', 56814)
Salvando multiples_secciones_criticas.c
1024 de 2365
2048 de 2365
3072 de 2365
Se guardo exitosamente el archivo, listo para recibir nuevo
cliente ('127.0.0.1', 56814)
Salvando prod_cons.c
1024 de 2496
2048 de 2496
3072 de 2496
Se guardo exitosamente el archivo, listo para recibir nuevo
cliente ('127.0.0.1', 56814)
Salvando Proyecto.c
1024 de 9395
2048 de 9395
3072 de 9395
4096 de 9395
5120 de 9395
6144 de 9395
7168 de 9395
8192 de 9395
9216 de 9395
10240 de 9395
Se guardo exitosamente el archivo, listo para recibir nuevo
cliente ('127.0.0.1', 56814)
Salvando resultado.txt
1024 de 25148
2048 de 25148
3072 de 25148
4096 de 25148
5120 de 25148
```

Archivos recibidos en la carpeta:

Nombre	Fecha de modificación	Tipo	Tamaño
13-04-21-11-47-48-AM_EduardoCastro.c	13/04/2021 11:47 a. m.	Archivo C	5 KB
13-04-21-11-47-48-AM_Examen3.png	13/04/2021 11:47 a. m.	Archivo PNG	81 KB
13-04-21-11-47-48-AM_Examen3.py	13/04/2021 11:47 a. m.	Python File	4 KB
13-04-21-11-47-48-AM_multiples_seccio...	13/04/2021 11:47 a. m.	Archivo C	3 KB
13-04-21-11-47-48-AM_prod_cons.c	13/04/2021 11:47 a. m.	Archivo C	3 KB
13-04-21-11-47-48-AM_Proyecto.c	13/04/2021 11:47 a. m.	Archivo C	10 KB
13-04-21-11-47-48-AM_resultado.txt	13/04/2021 11:47 a. m.	Documento de te...	25 KB
13-04-21-11-47-48-AM_semaforo.c	13/04/2021 11:47 a. m.	Archivo C	2 KB
13-04-21-11-47-48-AM_Semaforos.pdf	13/04/2021 11:47 a. m.	Chrome HTML Do...	330 KB
FileClient.py	12/04/2021 08:47 p. m.	Python File	2 KB
FileServer.py	11/04/2021 08:29 p. m.	Python File	2 KB

## CONCLUSIONES

- **Oscar Guzmán Pérez**

De esta práctica concluyó que distribuir la información por medio de la red en presenta diferentes ventajas y que los sockets ofrecen un número muy grande de posibilidades para aplicarlos, en este caso el enviar archivos de un cliente a un servidor abre la posibilidad de liberar almacenamiento en el cliente lo cual es una gran ventaja por la cantidad de recursos de las que los equipos suelen disponer, aunque en este caso la aplicación o el recurso beneficiado por parte del cliente es el almacenamiento, no es el único recurso del que podemos sacar mayor ventaja de esta manera puesto que también podemos utilizar esto para procesar cálculos o otras tareas pesadas en el servidor y enviar sólo el resultado al cliente, de esta forma aun si el dispositivo cliente posee menos recursos se podrán realizar más tareas con él, o simplemente reducir el tiempo que le tomaría procesarlas de forma local.

- **Castro Cruces Jorge Eduardo**

En conclusión, podemos decir que se logró en su totalidad los objetivos de esta práctica, los cuales son crear un canal de comunicación entre un cliente y un servidor, y poder transmitir n cantidad de archivos de distinto formato: mp3, pdf, png, jpg, etc.

- **Frausto Hernández Omar**

Para el desarrollo de esta práctica fue necesario la implementación de sockets tal como lo pedía nuestro profesor, se pudo desarrollar en su totalidad dicha práctica y aprendimos bastante a manejar sockets