

# Programación Distribuida y Tiempo Real

## Práctica 1

1) Teniendo en cuenta la comunicación con sockets (puede usar tanto los ejemplos provistos como también otras fuentes de información, que se sugiere referenciar de manera explícita):

- a.- Identifique similitudes y diferencias entre los sockets en C y en Java.
- b.- ¿Por qué puede decirse que los ejemplos no son representativos del modelo c/s? Nota: corroborar con la clase donde se explica el modelo C/S.
- c.- ¿Qué cambio/s deberían hacerse para que “cliente” provisto funcione como “servidor” y el “servidor” provisto funcione como “cliente”? Nota: corroborar con la clase donde se explica el modelo C/S.

2) Desarrolle experimentos para quede claro que:

- a.- Aunque un proceso “servidor” programado en C haya obtenido un socket y hecho un bind() no va a haber ningún otro proceso que pueda hacer una conexión con él a menos que se haya hecho el listen() ¿Por qué este mismo experimento no podría hacerse programando en Java?
- b.- Un proceso “cliente” puede tener una conexión con el proceso “servidor” aunque el “servidor” no haya ejecutado la operación accept().

3) Modifique el código (programa C o Java o ambos) para que la cantidad de datos que se comunican sea de  $10^1$ ,  $10^2$ ,  $10^3$ ,  $10^4$ ,  $10^5$  y  $10^6$  y contengan bytes asignados directamente en el programa (sin leer de teclado ni mostrar en pantalla los datos del buffer). Tenga en cuenta que el valor de retorno de la llamada a read(...) tanto en C como en Java retornan la cantidad de bytes efectivamente leída y puede ser menor a la que se le indica/pedida como parámetro. Explique cómo verifica el correcto funcionamiento de lo desarrollado. El envío debe realizarse en una única llamada a la función correspondiente a menos que el valor de retorno indique que hay datos pendientes de envío.

4) Con el programa que desarrolló en el ejercicio anterior para comunicar  $10^1$ ,  $10^2$ ,  $10^3$ ,  $10^4$ ,  $10^5$  y  $10^6$  bytes desarrolle un experimento que muestre el tiempo que toma en el cliente

a) La función write(...) para cada cantidad de datos. Tomando C como ejemplo:

```
t0 = ...
write(...)
t1 = ...
tiempo = t1 - t0;
```

b) La función `read(...)`. Tomando C como ejemplo:

```
t0 = ...  
n = read(sockfd, buffer, 255);  
t1 = ...  
tiempo = t1 - t0;
```

c) Grafique y explique los resultados obtenidos. En particular, identifique si las diferencias de tiempos son proporcionales a las cantidades de datos, por ej: ¿`write(...)` con 1000 bytes toma 10 veces más tiempo que con 100 bytes? Identifique si el tiempo de la función `read()` se mantiene constante, dado que involucra siempre la misma cantidad de datos.

5) ¿Podría implementar un servidor de archivos remotos utilizando sockets? Describa brevemente la interfaz y los detalles que considere más importantes del diseño. No es necesario implementar.

6) Explique y justifique brevemente ventajas y desventajas de un servidor con estado respecto de un servidor sin estados. Considere las respuestas de diferentes asistentes de IA y analice

a) Similitudes y diferencias entre las respuestas de los diferentes asistentes de IA.

b) Elabore su propio análisis.