

# Programación Distribuida y Tiempo Real

## Práctica 1

1) Teniendo en cuenta al menos los ejemplos dados (puede usar 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?

2) Desarrolle experimentos que se ejecuten de manera automática en una máquina virtual (mv) con Vagrant donde

- a.- Se muestre que no necesariamente siempre se leen/escriben todos los datos involucrados en las comunicaciones con una llamada read/write con sockets. Sugerencia: puede modificar los programas (C o Java o ambos) para que la cantidad de datos que se comunican sea de  $10^3$ ,  $10^4$ ,  $10^5$  y  $10^6$  bytes y contengan bytes asignados directamente en el programa (pueden no leer de teclado ni mostrar en pantalla cada uno de los datos del buffer), explicando el resultado en cada caso. **Importante:** notar el uso de “attempts” en “...attempts to read up to count bytes from file descriptor fd...” así como el valor de retorno de la función read (del man read).

- b.- Agregue a la modificación anterior una verificación de llegada correcta de los datos que se envían (cantidad y contenido del buffer), de forma tal que se asegure que todos los datos enviados llegan correctamente, independientemente de la cantidad de datos involucrados.

3) Tiempos y tamaños de mensajes

- a.- Como en el caso anterior, desarrolle y documente experimentos en mv para evaluar obtener los tiempos de comunicaciones para tamaños de mensajes de  $10^1$ ,  $10^2$ ,  $10^3$ ,  $10^4$ ,  $10^5$  y  $10^6$  bytes.

- b.- Grafique el promedio y la desviación estándar para cada uno de los tamaños del inciso anterior.

- c.- Provea una explicación si los tiempos no son proporcionales a los tamaños (ej: el tiempo para  $10^2$  bytes no es diez veces mayor que el tiempo para  $10^1$  bytes).

4) ¿Por qué en C se puede usar la misma variable tanto para leer de teclado como para enviar por un socket? ¿Esto sería relevante para las aplicaciones c/s?

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) Defina qué es un servidor con estado (stateful server) y qué es un servidor sin estado (stateless server).

**Entrega de la práctica** (individual o en grupos de dos estudiantes como máximo):

- Se debe entregar un informe de lo realizado para cada ejercicio. Debe tener un formato bien definido identificando materia, trabajo práctico y autor/es. Se debe entregar en formato electrónico con tipo de archivo .pdf, en tamaño de hoja A4.

- Para cada programa modificado o generado para resolver los ejercicios, debe explicarse el cambio o la implementación realizada. Si bien el programa fuente puede estar comentado, el cambio o la implementación realizada debe explicarse en el texto del informe (no es aceptable “ver código fuente” en el informe).

- Se debe entregar en formato electrónico tanto el informe como todo el código fuente usado/desarrollado.