

# Sistemas Operativos

ITBA

Cuatrimestre 1 - 2012

## Filesystems, IPCs y Servidores Concurrentes

### Objetivo

El objetivo de este trabajo es familiarizarse con el uso de sistemas cliente-servidor concurrentes, implementando el servidor mediante la creación de procesos hijos utilizando `fork()` y mediante la creación de threads. Al mismo tiempo, ejercitar el uso de los distintos tipos de primitivas de sincronización y comunicación de procesos (IPC) y manejar con autoridad el filesystem de Linux desde el lado usuario.

### Enunciado

La misión en este trabajo práctico es crear una aplicación cliente servidor de una liga de fantasía. Los clientes deberán poder registrarse a la aplicación y crear o unirse a ligas, creando en ellas equipos. Luego de que una liga esté llena, los clientes podrán acceder al draft de esa liga y, por turnos, seleccionar los jugadores para sus equipos. Una vez que todos los jugadores hayan sido drafteados, la liga se convertirá en activa. Sobre una liga activa, los clientes podrán tradear jugadores entre sí y contratar jugadores de la lista de agentes libres (los que hayan sobrado del draft) reemplazando a uno de sus jugadores a cambio. Finalmente, el administrador deberá poder cargar partidos, que asignarán puntos a los jugadores y a aquellos equipos que dispongan de ellos al momento de la carga del partido.

La simulación tendrá un archivo de configuración con la lista de jugadores a draftear y varios con partidos a cargar

El servidor deberá soportar los siguientes comandos:

- `register [user, password]`: Crea una cuenta para acceder a la liga.
- `login [user, password]`: Accede a su cuenta.
- `list [leagues—teams—trades]`: Accede a la lista de ligas, equipos o trades.

- league show [leagueId] Muestra la tabla de posiciones de una liga y los puntajes individuales de los jugadores drafteados por esa liga
- team show [teamId]: Muestra información sobre el equipo, la liga a la que pertenece, los jugadores que drafteó, el puntaje individual de cada uno y su puntaje total.
- trade show [tradeId]: Muestra información sobre el trade, de que equipo a que equipo es, y qué jugadores interactúan en el intercambio.
- trade [teamId]: Genera un trade con ese equipo (solo en caso de tener un equipo en la misma liga que el equipo con quien se tradea)
- trade withdraw [tradeId]: Cancela un trade creado por uno mismo
- trade accept [tradeId]: Aceptar un trade del cual el usuario es destinatario.
- trade negotiate [tradeId]: Cancela un trade del cual se es destinatario y crea uno nuevo como contraoferta.

Los partidos se cargarán copiando el archivo correspondiente a la carpeta de partidos. El servidor deberá estar monitoreando el directorio constantemente para detectar nuevos partidos.

### Actividades

1. Implemente la simulación utilizando procesos y threads y haga cuatro versiones del sistema, usando las siguientes primitivas de IPC:

- Pipes o FIFOs.
- Colas de mensajes - System V o POSIX.
- Memoria compartida - System V o mmap(). Semáforos - System V o POSIX.
- Sockets - TCP o de dominio Unix.

### **Criterios de Implementación**

Los programas correrán en ambiente Linux. En su realización se requiere:

Buena modularización  
Buen estilo de programación  
Documentación adecuada  
Makefile para la construcción.

Estos factores se toman en cuenta para la evaluación. Los trabajos podrán hacerse en grupos de no más de cuatro integrantes.

### **Material a entregar**

Cada grupo deberá entregar todos los fuentes del trabajo debidamente comentados, junto a un makefile para compilarlos en un ambiente linux. Además deberán entregar un informe impreso detallando las decisiones respecto a los items que quedaron a elección del grupo, problemas presentados y solución de los mismos durante la realización del trabajo

### **Consideraciones**

Todo punto no explícito en este documento podrá ser interpretado a conveniencia del alumno, siempre dentro de los márgenes del sentido común. Ante la duda, consultar a los docentes o enviar un mail a la cátedra.

### **Fecha de entrega**

La fecha de entrega del trabajo práctico es el Lunes 9 de Abril a las 16:00.