

Sistemas Operativos

ITBA

Cuatrimestre 2 - 2011

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 organizar una red de distribución de materiales mediante una vía aérea a lo largo de diferentes lugares. Varias flotillas de aviones deberán transportar diversos medicamentos a través de múltiples países, cubriendo las necesidades particulares de cada uno.

Se definirá una zona con múltiples destinos, y varias rutas de vuelo entre ellos (no necesariamente todos los destinos poseen una ruta de vuelo directa). Cada destino poseerá una lista de medicamentos que necesita y las cantidades que requiere de cada uno. De una o varias de estas ciudades saldrán varios aviones de una o varias empresas farmacéuticas, con el objetivo de distribuir las medicaciones de la manera necesaria.

La complejidad adicional es que mientras que todos los aviones saben en todo momento qué medicamento se requiere en cada lugar (y si ya se satisfizo la necesidad de cada uno en particular), los aviones de cada empresa sólo pueden saber el plan de recorrido de los otros aéreos de la misma farmacéutica, desconociendo por completo los de las otras (si las hubiese). Por tal motivo, es posible que dos aviones de distintas empresas vayan al mismo destino a cubrir la misma necesidad por desconocimiento.

La simulación tendrá varios archivos de configuración: El mapa y las empresas.

El mapa tendrá la siguiente información:

- Lista de ciudades.
- Tiempo que se tarda en recorrer desde ciudad A a ciudad B (para todas aquellas ciudades que tengan una ruta directa de vuelo). Se asume que todos los aviones van a la misma velocidad siempre, para simplificar el modelo.
- Lista de medicaciones que requiere cada ciudad y en qué cantidad (para simplificar, asumimos que todas las medicaciones se miden en unidades).

Por otro lado, cada empresa tendrá un archivo de configuración con la siguiente información

- Lista de aviones.
- Lista de medicaciones disponibles como cargo de cada avión y cantidad.
- Punto de partida de cada avión.

El objetivo del trabajo práctico será realizar un simulador que dados los archivos de configuración necesarios pueda resolver la distribución de medicamentos en las diferentes ciudades.

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.

2. Los archivos de configuración tendrán el siguiente formato:

Mapa

- Una línea con la cantidad de ciudades seguida de una línea en blanco
- Por cada ciudad, una línea con el nombre de la ciudad, seguida de una línea por cada medicamento que requiere esa ciudad y la cantidad, separados por un espacio, seguido, finalmente, por una línea en blanco.
- Por cada ciudad, una línea por cada conexión de esa ciudad con otras ciudades que contiene: el nombre de la ciudad origen, el nombre de la ciudad destino y la distancia, separados por espacios. Al final de cada lista de ciudades, seguirá una línea en blanco. (Nota: Las vías de tránsito entre 2 ciudades no son dirigidas, si existe la vía que conecta A y B, y tarda N, se asume que hay una vía que va de B a A y tarda N también.

Empresa Farmacéutica:

- Una línea con la cantidad de aviones seguida de una línea en blanco.
- Por cada avión, una línea con el nombre de la ciudad en la que inicia el avión, seguido por una línea por cada medicamento que posee ese avión con el nombre y la cantidad del mismo separados por un espacio. Al final de cada lista, se agrega una línea en blanco.

3. El programa debe permitir la visualización de la simulación en un tablero en la pantalla, iterando turno a turno.

4. Cada miembro de cada tipo de unidad debe tener la misma programación, es decir, si bien lo que hacen dependerá de su posición y su cargamento y pueden elegir moverse a un lugar al azar, debe haber una única función con la programación para todos los aviones.

5. Se asume que el tiempo de descargar un avión es instantáneo e inmediatamente parte para la próxima ciudad.

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 12 de Septiembre a las 16:00.