Filesystems, IPCs y Servidores Concurrentes
"Smiling"

Magni Nicolás, Purita Nicolás, Zemin Luciano R.

April 3, 2010

Contents

0.1 Introducción

0.2 Objetivos

0.2.1 Objetivos del Trabajo

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

0.2.2 Objetivos Personales

0.3 Enunciado

Se desea implementar un servicio de resolución general de problemas. La idea consiste en tener un servidor dedicado a resolver dos tipos de problemas: un problema paralelizable y un problema que soporte pipelining. Este servidor estará escuchando dos directorios, en donde esperará a que alguien ingrese archivos de entrada con información sobre los problemas. Cada directorio recibirá archivos de entrada de uno de los dos tipos de problema. El servidor deberá consumir todos los archivos de entrada en esos directorios y procesar la información, distribuyendola en threads y procesos que procederán a obtener las solución parciales de cada uno de los archivos de entrada, para luego reunirlas en la solución final, que será guardada en un archivo nuevo en la carpeta de soluciones correspondiente. La cátedra no obliga a implementar ningún problema ni algoritmo en particular. Los alumnos están en condiciones de elegir los problemas que deseen.

0.4 Desarollo

En esta sección, se detallara todo lo relacionado al desarrollo del trabajo practico.

0.4.1 Paralleling

La computación paralela es una técnica de programación en la que muchas instrucciones se ejecutan simultáneamente. Se basa en el principio de que los problemas grandes se pueden dividir en partes más pequeñas que pueden resolverse de forma concurrente. Existen varios tipos de computación paralela: paralelismo a nivel de bit, paralelismo a nivel de instrucción, paralelismo de datos y paralelismo de tareas. Durante muchos años, la computación paralela se ha aplicado en la computación de altas prestaciones, pero el interés en ella ha aumentado en los últimos anios debido a las restricciónes físicas que impiden el escalado en frecuencia. La computación paralela se ha convertido en el paradigma dominante en la aquitectura de computadores, principalmente en los procesadores multinúcleo. Sin embargo, recientemente, el consumo de energía

de los ordenadores paralelos se ha convertido en una preocupación.

Paralelismo de tareas es un paradigma de la programación concurrente que consiste en asignar distintas tareas a cada uno de los procesadores de un sistema de cómputo. En consecuencia, cada procesador efectuará su propia secuencia de operaciones.

Algoritmo

0.4.2 Consideraciones Tomadas

0.4.3 Problemas Encontrados

0.4.4 Pipelining

Algoritmo

0.4.5 Consideraciones Tomadas

0.4.6 Problemas Encontrados

0.5 Librerias adicionales

0.6 Conclusión