

Técnicas de Programación Concurrente I

Introducción

Ing. Pablo A. Deymonnaz
pdeymon@fi.uba.ar

Facultad de Ingeniería
Universidad de Buenos Aires



1. Introducción

Definiciones

2. Modelos de Concurrencia

3. Threads

- ▶ **Programa:** conjunto de datos, asignaciones e instrucciones de control de flujo que compilan a instrucciones de máquina, las cuales se ejecutan secuencialmente en un procesador y acceden a datos almacenados en memoria principal o memorias secundarias.
- ▶ **Programa concurrente:** conjunto de programas secuenciales que pueden ejecutarse en paralelo.
- ▶ **Proceso:** cada uno de los programas secuenciales que conforman el programa concurrente.
- ▶ **Sistema paralelo:** sistema compuesto por varios programas que se ejecutan simultáneamente en procesadores distintos.

- ▶ **Multitasking:** ejecución de múltiples procesos concurrentemente en un cierto periodo de tiempo. El scheduler, parte del kernel del sistema operativo, se encarga de coordinar el acceso a los procesadores.
- ▶ **Multithreading:** construcción provista por algunos lenguajes de programación que permite la ejecución concurrente de threads dentro del mismo programa.

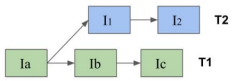
Necesidad de sincronizar y comunicar procesos diferentes.

- ▶ **Sincronización:** coordinación temporal entre distintos procesos
- ▶ **Comunicación:** datos que necesitan compartir los procesos para cumplir la función del programa

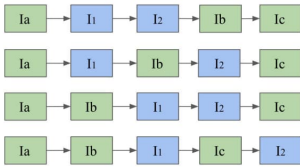
Definiciones

- ▶ **Programa concurrente:** consiste en un conjunto finito de procesos secuenciales.
- ▶ **Procesos:** están compuestos por un conjunto finito de instrucciones atómicas.
- ▶ **Ejecución del programa concurrente:** resulta al ejecutar una secuencia de instrucciones atómicas que se obtiene de intercalar arbitrariamente las instrucciones atómicas de los procesos que lo componen.

Ejecución del programa concurrente



Orden causal



Algunas ejecuciones posibles

Una instrucción atómica puede:

- ▶ Ejecutarse de principio a fin sin interrupciones.
- ▶ No ejecutarse.

1. Introducción

2. Modelos de Concurrencia

Modelos

3. Threads

- ▶ Estado mutable compartido
- ▶ Paralelismo fork-join
- ▶ Canales / mensajes
- ▶ Programación asincrónica
- ▶ Actores

Serializar el acceso al estado compartido

- ▶ Los procesos se ejecutarán al mismo tiempo, pero habrá ciertos conjuntos de procedimientos tales que solo una ejecución de un procedimiento en cada se permite que suceda a la vez.
- ▶ Si se está ejecutando algún procedimiento en el conjunto, entonces cualquier otro proceso que intente ejecutar cualquier procedimiento en el conjunto será obligado a esperar hasta que la primera ejecución haya terminado
- ▶ Serializar para controlar el acceso a las variables compartidas
- ▶ Podemos marcar regiones de código que no pueden superponerse en la ejecución al mismo tiempo.

1. Introducción

2. Modelos de Concurrencia

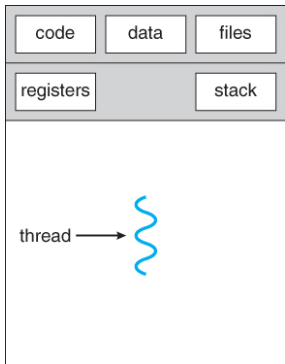
3. Threads

Threads Introducción

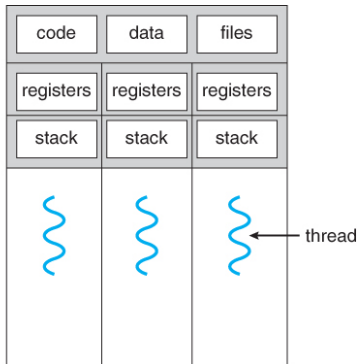
Threads Introducción

Los Threads comparten los recursos del proceso, entre ellos, el espacio de memoria.

Cada thread mantiene su propia información de estado (stack, PC, registros).



single-threaded process



multithreaded process

- ▶ **Principles of Concurrent and Distributed Programming**, M. Ben-Ari, Segunda edición (capítulos 1 y 2)
- ▶ **Operating System Concepts**, Ninth Edition, Abraham Silberschatz, Greg Gagne, and Peter Baer Galvin, Cap. 4.