1. Introducción

Dra. Elizabeth Pérez Cortés
UAM-Iztapalapa
Depto. de Ing. Eléctrica

1. Introducción

Elizabeth Pérez Cortés

1.1 Conceptos básicos

Considere el problema de multiplicar dos matrices A y B de 2x2 para obtener la matriz C. Si escribimos un programa para calcular C, resultará en algo como lo siguiente:

1. $t1 = A_{11} * B_{11}$
2. $t2 = A_{12} * B_{21}$
3. $C_{11} = t1 + t2$
4. $t3 = A_{11} * B_{12}$
5. $t4 = A_{12} * B_{22}$
6. $C_{12} = t3 + t4$
7. $t5 = A_{21} * B_{11}$
8. $t6 = A_{22} * B_{21}$
9. $C_{21} = t5 + t6$
10. $t7 = A_{21} * B_{12}$
11. $t8 = A_{22} * B_{22}$
12. $C_{22} = t7 + t8$
13. FIN

Este es un algoritmo secuencial pues especifica el siguiente orden total entre las instrucciones que componen el algoritmo:

$1 \rightarrow 2$, $2 \rightarrow 3$, $3 \rightarrow 4$, $4 \rightarrow 5$, $5 \rightarrow 6$, $6 \rightarrow 7$, $7 \rightarrow 8$, $8 \rightarrow 9$, $9 \rightarrow 10$, $10 \rightarrow 11$, $11 \rightarrow 12$ y $12 \rightarrow 13$.

Donde $a \rightarrow b$ denota que a debe ocurrir antes que b y si $a \rightarrow b$ y $b \rightarrow c$, entonces $a \rightarrow c$

Elizabeth Pérez Cortés

1.1 Conceptos básicos

Si tomamos las instrucciones de nuestro algoritmo y nos preguntamos ¿Qué pares deben estar ordenados?

1. $t1 = A_{11} * B_{11}$
2. $t2 = A_{12} * B_{21}$
3. $C_{11} = t1 + t2$
4. $t3 = A_{11} * B_{12}$
5. $t4 = A_{12} * B_{22}$
6. $C_{12} = t3 + t4$
7. $t5 = A_{21} * B_{11}$
8. $t6 = A_{22} * B_{21}$
9. $C_{21} = t5 + t6$
10. $t7 = A_{21} * B_{12}$
11. $t8 = A_{22} * B_{22}$
12. $C_{22} = t7 + t8$
13. FIN

Notaremos que son solo los siguientes:

$1 \rightarrow 3,\ 2 \rightarrow 3$

$4 \rightarrow 6,\ 5 \rightarrow 6$

$7 \rightarrow 9,\ 8 \rightarrow 9$

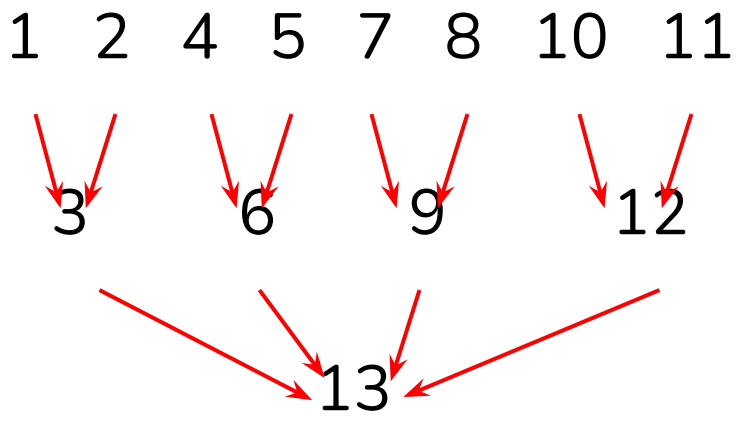
$10 \rightarrow 12,\ 11 \rightarrow 12$

$3 \rightarrow 13, 6 \rightarrow 13, 9 \rightarrow 13, 12 \rightarrow 13$

Elizabeth Pérez Cortés

1.1 Conceptos básicos

Lo cual también se puede representar así:



Las instrucciones entre las que no hay una relación de orden (por ejemplo las que están en la misma fila) se pueden ejecutar simultáneamente y para indicar eso debemos escribir un nuevo algoritmo para este problema. Uno en donde se exprese esa posibilidad.

Elizabeth Pérez Cortés

1.1 Conceptos básicos

Definiciones

Según la RAE, **concurrencia** significa la acción o efecto de **concurrir**, que, a su vez, significa: “Dicho de diferentes personas, sucesos o cosas: **Juntarse en un mismo lugar o tiempo**”.

Elizabeth Pérez Cortés

1.1 Conceptos básicos

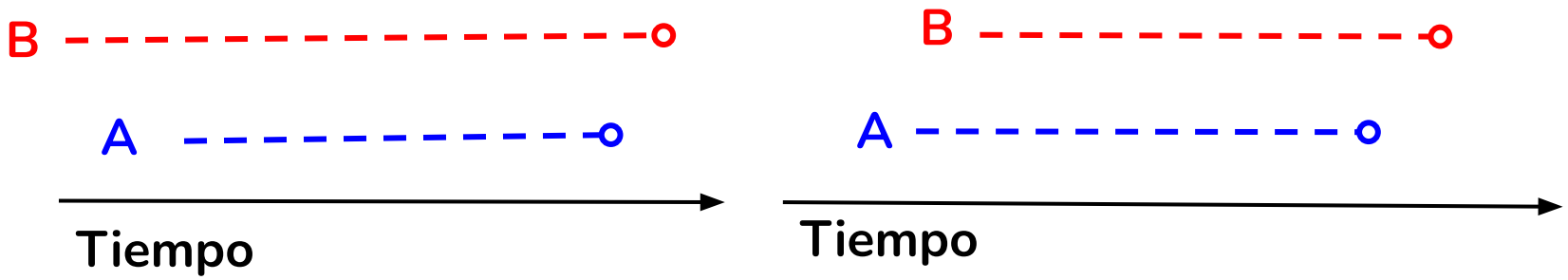
Definiciones

- Un **algoritmo** contiene las instrucciones que deben llevarse a cabo para resolver un problema.
- Un **algoritmo es secuencial** si es una secuencia de instrucciones, es decir, si define un orden total entre las instrucciones.
- Un **algoritmo es concurrente** si contiene instrucciones que **pueden** ser ejecutadas simultáneamente. En otras palabras, si define un orden parcial entre las instrucciones a ejecutar para resolver el problema correspondiente.

Elizabeth Pérez Cortés

1.1 Conceptos básicos

Se dice que la ejecución de dos programas secuenciales A y B es concurrente o se traslapa **si la primera instrucción de A ocurre después de la primera instrucción de B y antes de la última o viceversa**.



Elizabeth Pérez Cortés

1.1 Conceptos básicos

Un **algoritmo concurrente** está compuesto por un conjunto de algoritmos secuenciales que **pueden** ejecutarse simultáneamente (Ben-Ari). Es decir, cuya ejecución se puede traslapar. Cada algoritmo secuencial está a cargo de una **tarea**.

Por ejemplo para calcular la multiplicación de matrices, se debe calcular cada una de las entradas de la matriz resultado así que se tendrá un algoritmo secuencial que lleve a cabo la tarea de calcular la entrada.

Elizabeth Pérez Cortés

1.1 Conceptos básicos

Si usamos el operador || para indicar “simultáneamente”, el algoritmo concurrente quedaría como sigue:

```
{
{t1 = A11 *B11; t2= A12*B21; C11 = t1 + t2} ||
{t3 = A11 *B12; t4 = A12*B22; C12 =   t3+t4} ||
{t5 = A21 *B11; t6 = A22*B21; C21 = t5 + t6} ||
{t7 = A21 *B12; t8 = A22*B22; C22 =  t7 + t8}
} FIN
```

Elizabeth Pérez Cortés

1.1 Conceptos básicos

Note que los algoritmos secuenciales aún están imponiendo orden entre algunas instrucciones que no lo necesitan. Una versión que explote toda la simultaneidad que el problema permite, sería la siguiente:

{
{{t1 = A_{11} *B_{11}|| t2= A_{12}*B_{21}} C_{11} = t1 + t2} ||
{{t3 = A_{11} *B_{12}|| t4 = A_{12}*B_{22}} C_{12} = t3+t4} ||
{{t5 = A_{21} *B_{11}|| t6 = A_{22}*B_{21}} C_{21} = t5 + t6} ||
{{t7 = A_{21} *B_{12}|| t8 = A_{22}*B_{22}} C_{22} = t7 + t8}
} FIN

Elizabeth Pérez Cortés

1.1 Conceptos básicos

Cuando uno de estos algoritmos se expresa usando algún lenguaje de programación, entonces se tiene un **programa concurrente**.

Un programa concurrente al ejecutarse da lugar a **varios flujos de ejecución**, cada uno de ellos es llevado a cabo por una *entidad activa* dentro de una plataforma de cómputo.

Elizabeth Pérez Cortés

1.1 Conceptos básicos

Programación concurrente: Disciplina de las ciencias de la computación encargada del estudio de los problemas inherentes a la concurrencia y sus soluciones.

Elizabeth Pérez Cortés

1.1 Conceptos básicos: relaciones entre las entidades activas

Las entidades activas que se ejecutan concurrentemente pueden ser clasificadas con base en la percepción que tienen de la existencia de otra, como:

- Ajenas: Las entidades activas no trabajan juntas. Este es el caso de los procesos en un sistema multiprogramado. Aunque no trabajan juntas, las entidades usan los recursos de la plataforma (como el procesador, impresoras, discos, etc) y, en consecuencia, compiten por ellos.
- Indirectamente conscientes de la existencia de otra: Estas entidades activas están relacionadas porque comparten algún recurso (por ejemplo: un archivo) y en ese sentido deben cooperar para usarlo adecuadamente.
- Directamente conscientes de la existencia de otra: Estas entidades activas están relacionadas porque los algoritmos que ejecutan fueron diseñados para trabajar juntos

Se requieren herramientas de comunicación y sincronización para el correcto uso de los recursos.

Elizabeth Pérez Cortés

1.1 Conceptos básicos: Comunicación y sincronización entre tareas

- **Comunicación**: intercambio de datos entre las tareas
- **Primitivas de comunicación**: conjunto de herramientas provistas por las capas de soporte para permitir la comunicación de los programas concurrentes.
- Tipos principales:
  - **Basados en pizarrón**: las entidades se comunican escribiendo y leyendo datos
  - **Basados en mensajes**: las entidades envían y reciben los datos

Elizabeth Pérez Cortés

1.1 Conceptos básicos: Comunicación y sincronización entre tareas

- **Sincronización: Control del orden de ejecución de las instrucciones en una ejecución concurrente.**

- **Primitivas de sincronización: conjunto de herramientas provistas por las capas de soporte para sincronizar los programas concurrentes.**

Elizabeth Pérez Cortés

1.1 Conceptos básicos: Modos de acceso a los recursos compartidos.

La manera más simple de gestionar el uso de los recursos es conceder su uso exclusivo a una entidad activa a la vez.

Para implementarla, se ubican las secciones de código en donde se usan los recursos (llamadas secciones críticas) y se utilizan mecanismos de sincronización que garanticen la exclusión mutua, es decir, que solo una entidad activa esté ejecutándose en su sección crítica en cualquier instante de tiempo.

En ocasiones, un recurso compartido puede ser usado por varias entidades simultáneamente sin que esto tenga efectos nocivos ni sobre el recurso ni sobre la ejecución, en estos casos se dice que se permite el acceso compartido y se usan mecanismos de sincronización que lo permitan.

Elizabeth Pérez Cortés

1.1 Conceptos básicos: Modos de acceso a los recursos compartidos.

La exclusión mutua suele implementarse de acuerdo al siguiente protocolo.

Adquirir el derecho de acceso

sección crítica

Liberar el derecho de acceso

Es decir, en el programa/algoritmo se incluyen las primitivas de sincronización para adquirir el derecho de acceso a los recursos antes de la sección crítica y, al final, se incluyen las primitivas de sincronización para liberar el derecho de acceso a los recursos y permitir que otra entidad los utilice.

Elizabeth Pérez Cortés

1.1 Conceptos básicos: Modos de acceso a los recursos compartidos.

Asumiendo que la ejecución de la sección crítica consume un tiempo finito, las primitivas de sincronización para garantizar el uso exclusivo de los recursos deben asegurar:

1. La exclusión mutua.
2. Que ninguna entidad activa esperará indefinidamente a que se le asigne el derecho de acceso a ejecutar su sección crítica. Se dice que ninguna entidad sufrirá de inanición.
3. Que si ninguna entidad activa está ejecutando su sección crítica, porque se han detenido o están ejecutándose fuera de esta, cualquier entidad activa que solicite el derecho de acceso a su sección crítica debe recibirlo de inmediato sin tardar. En otras palabras, deben garantizar el progreso.

Elizabeth Pérez Cortés