Lab. Programación Concurrente

7° Programación Valenzuela/Varela

Programación concurrente

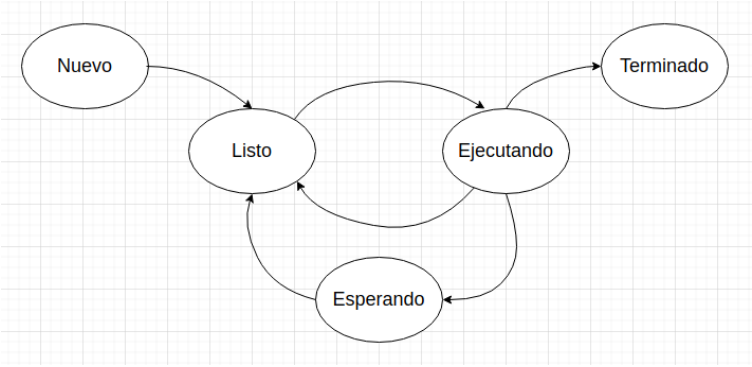
Repasando una y otra vez

¿Qué vimos hasta ahora?

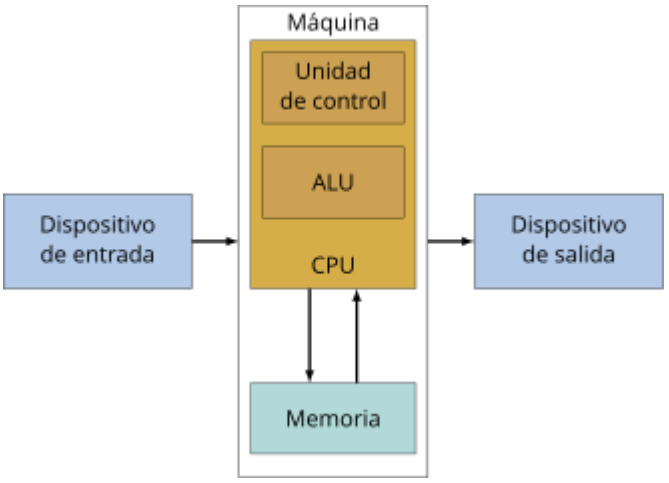
Para esta altura de la materia deberíamos tener en claro qué es un proceso, qué es un hilo (thread), la diferencia entre ambos, qué es una traza y sobre todo poder explicar con nuestras palabras qué es un programa concurrente.

Un concepto que es fundamental para entender la programación concurrente es nuestro caprichoso amigo el scheduler, sin él nada sería posible ya que es el que va repartiendo aleatoriamente tiempo de ejecución a cada uno de nuestros threads (hilos)

Recordemos también que el scheduler conceptual que nosotros tenemos en cuenta para trabajar le va a asignar SIEMPRE tiempo de ejecución a un thread que esté esperando a ser ejecutado. Y que los procesos pueden tener varios estados diferentes según su estado de ejecución. A continuación el gráfico que más que una explicación una imágen que vale más que mil palabras…



Vamos a volver un paso atrás, y recordar el esquema de una computadora según Von Newman



Hasta ahora vimos básicamente todo lo que pasa en la parte marrón del esquema. Pero ¿Qué pasa cuando queremos usar la memoria en programas concurrentes?

¿Existe algún problema o podremos usarla sin ningún tipo de cuidados?

Obviamente no habría dos docentes si esto fuese tan simple…

Lab. Programación Concurrente

7° Programación Valenzuela/Varela

Memoria compartida

Lo primero que vamos a explicar es que no todo es lo que parece cuando escribimos una instrucción en un programa, y que muchas veces damos cosas por obvias y evidentes cuando realmente hay muchas cosas que ignoramos por simplicidad.

El mejor y más fácil ejemplo es la siguiente instrucción:
(supongamos que inicialmente x = 0)

x = x + 1

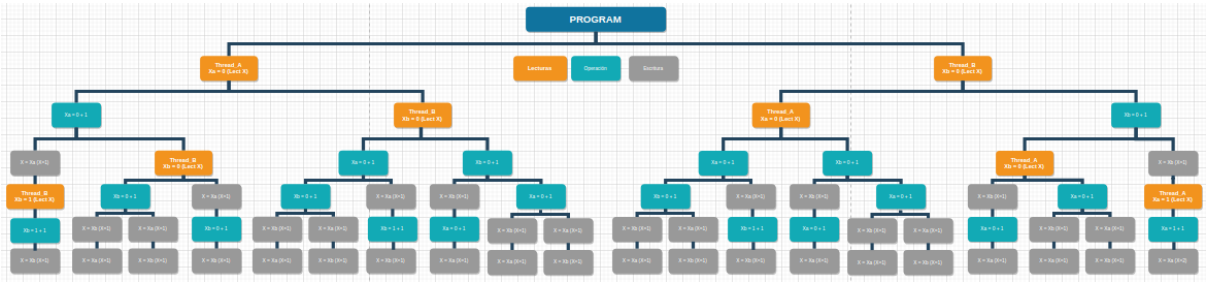
¿Cuántos tiempos de procesamiento piensan que tendrá esa simple instrucción?
Muches seguro pensarán en UN solo tiempo de procesamiento.
Algune podrá decir DOS, uno para realizar la operación suma y otro para guardar el resultado.
Pero, casi seguro, pocos dirán que son **TRES tiempos de procesamiento**:

- Leer el valor de la variable en memoria (Lectura)
- Realizar la operación (Operación)
- Guardar el resultado en memoria (Escritura)

Que esto pase en 3 tiempos es un problema… un gran problema. Porque como ya dijimos el scheduler es caprichoso y puede sacarnos el tiempo de procesamiento en el medio de una de esas 3 operaciones y darle tiempo a otro thread dejando nuestra *instrucción de una línea sin terminar*.
Veámoslo con un ejemplo:

```
1   Global x = 0
2
3   thread A {              thread B {
4       x = x +1                x = x + 1
5   }                       }
6
7
```

En este link van a encontrar el gráfico de trazas del ejercicio, les pego acá una captura pero a sabiendas de que no se van a ver todos los detalles. Por otro lado el link es privado así que para verlo van a tener que solicitar permiso lo cual marca que leyeron esto y cuenta como un presente… si somos rebuscados, pero los queremos…



Como veníamos diciendo antes, los grafos son muy engorrosos de leer y para peor no se ve toda la información que queremos de la forma que nos gustaría. Así que les vamos a mostrar una nueva forma de mostrar trazas.
Ya no queremos ver TODAS las trazas, solamente queremos ver las que tengan algo interesante para mostrar

Ejemplo de este ejercicio. ¿Cuántos valores puede tomar X al finalizar la ejecución?
Respuesta: X puede tomar los valores 1 y 2

X = 2

thread y linea	valor de var Global	valor de var Local	lect / op / esc	instruccion
Tread A.1	X = 0	Xa = 0	Lectura	X = X + 1
Tread A.1	X = 0	Xa = 1	Operación	X = X + 1
Tread A.1	X = 1	Xa = 1	Escritura	X = X + 1
Tread B.1	X = 1	Xb = 1	Lectura	X = X + 1
Tread B.1	X = 1	Xb = 2	Operación	X = X + 1
Tread B.1	X = 2	Xb = 2	Escritura	X = X + 1

X = 1

thread y linea	valor de var Global	valor de var Local	lect / op / esc	instruccion
Tread A.1	X = 0	Xa = 0	Lectura	X = X + 1
Tread A.1	X = 0	Xa = 1	Operación	X = X + 1
Tread B.1	X = 0	Xb = 0	Lectura	X = X + 1
Tread A.1	X = 1	Xa = 1	Escritura	X = X + 1
Tread B.1	X = 1	Xb = 1	Operación	X = X + 1
Tread B.1	X = 1	Xb = 1	Escritura	X = X + 1

Estas tablas son las que vamos a pedir que nos hagan en los ejercicios para mostrar trazas específicas.