

Sebastián Cárdenas
13638211

pregunta 1: ¿Que ventajas y desventajas podr ía tener la implementación de esta tarea con threads en vez de el uso de procesos separados?

En general

*las ventajas de los FORK() son:

- Son seguros ya que cada proceso tiene su propio espacio de ejecución, además de que no pueden modificar las variables de los otros procesos.
- Al ser independientes, si se cae algún proceso este no afecta al resto

*las desventajas:

- Cada proceso requiere su propio espacio de memoria por lo que debe manejar cada vez que se realiza un fork
- La comunicación entre procesos es complicada ya que no se puede compartir memoria entre ellos y se deben utilizar métodos externos

*las ventajas de los Threads son:

- Todos los threads comparten el mismo espacio de memoria, por ende son más livianos ya que ocupan menos memoria que los fork
- Comunicación rápida entre procesos

*las desventajas:

- son más inseguros dado que todos los threads de un proceso comparten la misma memoria por ende es fácil modificar variables o datos.

Dado lo anterior si lo aplicamos en este proyecto; las ventajas de usar thread que que se ocupe menos memoria para alcanzar el objetivo, además de hacer el programa más escalable a inputs de tamaños muy grandes ya que al usar fork se podría limitar por uso de memoria. Las desventajas sería la disminución de seguridad entre procesos debido a esta memoria compartida, ya que tendría una mayor probabilidad de generar un cambio no deseado.

pregunta 2: Ejecute su programa de tal manera que haya 1, 2, 4, 8 y 16 simulaciones ejecutando simultáneamente. Para cada ejecución mida su tiempo de ejecución usando time2. Explique el comportamiento de los tiempos real, user, y sys.

Primero aclararemos algunos puntos.

Def tiempos;

- real : Es el tiempo real invertido en ejecutar el proceso de principio a fin, como si lo midiera un humano con un cronómetro.
- usuario : Es el tiempo acumulado gastado por todas las CPU durante el cálculo.
- sys : Es el tiempo acumulado que gastan todas las CPU durante las tareas relacionadas con el sistema, como la asignación de memoria.

En procesadores multi-core puede darse el caso en que $\text{sys} + \text{user} > \text{real}$ esto se debe a que los procesadores pueden funcionar en paralelo por ende el tiempo real disminuye.

simulación con tiempo $t = 20$

Cantidad de procesos simultáneos; 1

real 0m0,004s
user 0m0,002s
sys 0m0,002s

Cantidad de procesos simultáneos; 2

real 0m0,005s
user 0m0,006s
sys 0m0,001s

Cantidad de procesos simultáneos; 4

real 0m0,006s
user 0m0,005s
sys 0m0,008s

Cantidad de procesos simultáneos; 6

real 0m0,006s
user 0m0,006s
sys 0m0,006s

Cantidad de procesos simultáneos; 12

real 0m0,007s
user 0m0,0013s
sys 0m0,009s

Se puede observar que no existe un gran incremento de los tiempos a medida de que se van generando más procesos simultáneos esto es debido a la generación de los procesos en paralelo si bien no ha sido un gran incremento, a medida de que se generen más procesos se verá limitada por la cantidad de cores que posee el computador además de que el wait implementado en el programa genera incrementos en los tiempos de ejecución dentro de cada proceso por lo tanto es lógico que haya pequeños incrementos al momento de duplicar o triplicar la cantidad de procesos en paralelo. pero sigue siendo más eficiente que hacer el programa en forma secuencial.