

Tarea 2 Sistemas Operativos

19 de abril de 2019

Pregunta 1

Antes de responder, quiero entregar las siguientes afirmaciones:

- Procesos "forked" no comparten recursos como ".data", ".text", stack, heap. Por lo que se debe crear una copia de memoria de el proceso padre.
- Para poder acceder a datos compartidos se crea un segmento de memoria compartido con shmget(2).
- El cambio de procesos requiere ayuda del sistema operativo, mientras que los threads pueden hacer 'switch' en modo user, siempre cuando existan suficientes kernel threads asociados.
- Crear multiples procesos es mas recurso intensivo porque es necesario crear una copia entera de la memoria de un procesos. Mientras que con threads no, tiene su propio stack.

En conclusion, la eleccion entre procesos y threads en terminos de eficiencia depende del problema que se quiere resolver. Para el caso especifico de la tarea, dado que cada map es independiente. Es mas eficiente usar procesos y despues juntarlos. El unico factor que puede ser decisivo es que tan costos es tener el espacio de memoria compartido con shmget(2). Si lo hubiera podido haber implementado, hubiera sido interesante.

Pregunta 2

Mi sistema tiene un limite de threads de 32 mil threads, pero no vale la pena tener tantos si el overhead de crearlos y mantenerlos es muy grande. En mi tarea, creo un numero de reduce según yo minimiza el tiempo de ejecución porque use un arbol binario incompleto.