

1. ¿Podría correr la simulación en un sistema de procesador único? ¿Hace falta asumir alguna característica en tal caso? ¿Qué cambiaría si se corre en un sistema multiprocesador?

Las simulaciones de Los Ángeles y Evitas pueden ejecutarse en un sistema de un solo procesador siempre que el scheduler tenga la capacidad preemptive. Para cambiar entre procesos, el scheduler debe ser del tipo scheduling, round-robin, o shortest-job-first. Adam, Lilith y sus respectivos hijos correrían en paralelo con la simulación en nuestro caso si se ejecuta en un sistema multiprocesador.

2. Si el sistema donde corro mi simulación utiliza un scheduler First-Come, First-Served, ¿podría pasar que se den ejecuciones donde nunca termine? Asumiendo que mi simulación convive con otros procesos en el sistema, y que se adopta un scheduler con política Shortest Job First con preemption, ¿podría haber starvation?

No, no podría pasar. Las ejecuciones dependen de la cantidad de días máximos. Si, la simulación podría tener starvation con un scheduler con shortest job first con preemptive, porque al crearse Adan, y lilith, crean o intentan crear un hijo y al crear alguno de estos su hijo podría ser un proceso que el proceso sea más corto que el del otro hijo creado entonces se pondría primero que el en la fila y si esto se repitiera varias veces, se generaría starvation

3. En su implementación, ¿podría ocurrir que la simulación no se de en orden? Es decir, que algún proceso creado en el día t entregue su output luego de que otros procesos creados en un día k con $k > t$ hagan el suyo. Si es así, ¿podría solucionarse?

La simulación adopta un enfoque en cascada en el que los padres esperan hasta que sus hijos proporcionen el resultado, lo que resulta en el hecho de que un proceso "K" es siempre el que entrega la salida antes que un proceso "T" porque este último es el primero. padre. Debido a la proximidad a la estructura en cascada, esto no se puede cambiar sin volver a ejecutar la simulación desde cero.

4. ¿Utilizaron algún esquema de comunicación entre procesos? ¿Podría pasar que dos procesos intenten modificar el mismo espacio de memoria concurrentemente?

Se utilizó el sistema Mmap, puntero que permite la creación de una lista accesible a todos los procesos; Debido a la utilización de este puntero global podría pasar que dos procesos choquen queriendo modificar el mismo espacio de memoria. Como solución se implementó el uso de <atomic> para evitar este tipo de problemas.