

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?

Sí y no. La simulación utiliza más de un proceso en simultáneo.

Puede ser ejecutado si el procesador tiene los hilos suficientes.

Habiendo más procesadores se permitiría un mejor funcionamiento del programa. Un procesador puede tener muchos hilos, pero al dividirse tanto, no asegura un buen funcionamiento.

Tener un multiprocesador da más posibilidades de que la simulación sea ejecutada de manera óptima, al tener más procesadores.

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?

Si puede pasar, si se hace una mala utilización de los exit de los procesos y un mal uso de la espera de los procesos.

Si, podría solucionarse mirando que todos los procesos sean esperados para no crear zombis y que estos a su vez terminen.

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

si la comunicación entre procesos local, esto es así porque se corre en una misma máquina, podría pasar que dos procesos intentan modificar el mismo espacio de memoria concurrente, pero con el atomic específicas que no pisen el mismo espacio de memoria.