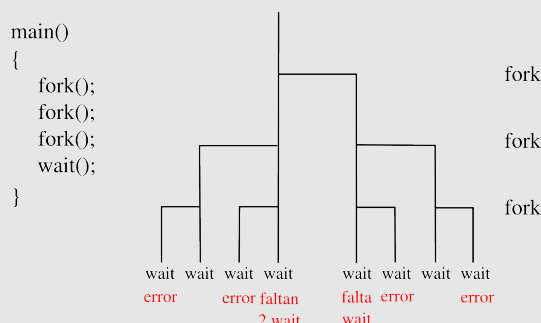


3

1. Se realiza el código adjunto. Realiza un diagrama con los procesos creados. Explica a través del diagrama cómo finalizan los procesos, si quedan procesos zombies o huérfanos y cuáles son si los hubiese.

Solución:

Con el primer fork se lanza un hijo. En el segundo fork, tanto el padre como el hijo lanza cada uno un proceso y por tanto hay un padre, dos hijos y un nieto. En el tercer fork el padre, los dos hijos y el nieto lanzan un nuevo proceso. Así quedan un padre, tres hijos, tres nietos y un biznieto. Todos ellos hacen un wait. Como se puede ver en la figura, el biznieto, dos nietos y un hijo no tienen descendientes. De esta forma, el wait de estos tres devuelve un error al no detectarse hijos y por tanto pueden finalizar correctamente. De igual forma un hijo y un nieto hacen un wait y sólo tienen un descendiente y es uno de los anteriores. Por tanto, en algún momento estos procesos terminarán correctamente. Por otro lado, uno de los hijos tiene dos descendientes y hace un sólo wait, como sus descendientes terminan correctamente, eso implica que uno de sus descendientes quedará en el estado de zombie. El padre tiene tres descendientes y hace un sólo wait. Como sus descendientes terminan correctamente eso implica que al menos dejará dos de sus descendientes en el estado de zombie o huérfanos, depende de quién acabe primero.

3

2. Se ejecutan 4 procesos según la tabla adjunta utilizando la planificación Round-Robin. Téngase en cuenta que en el caso en que se apliquen varios niveles de planificación simultáneamente se ejecutará primero el de largo plazo, posteriormente el de E/S y finalmente el de corto plazo. En caso de empates tendrá prioridad el que haya entrado antes en el sistema. Posteriormente calcula las siguientes cantidades:

$$30/31 = 96,77\%$$

Uso de la CPU

Rendimiento

$$4/31 \text{ proc/q} = 0,129 \text{ proc/q}$$

$$2,255 \text{ q}$$

Tiempo de retorno

Tiempo de espera medio

$$12,75 \text{ q}$$

Proceso	T_0	CPU	E/S	CPU	E/S	CPU	E/S	CPU
A	0	3	2	3	1	4	1	1
B	1	2	1	2	2	2	1	1
C	2	1	2	2	3	2	1	2
D	3	1	1	2	3	1	2	1

Solución:

Ejecutando



Listo



Bloqueado



Finalizado



T	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
A																																		
B																																		
C																																		
D																																		