Respuestas:

1.

En el código, siempre se imprime el mismo valor de *x*. Lo que veremos en pantalla será el valor 1 correspondiente a la variable *x* del padre. Esto ocurre porque al forkear, cada proceso maneja su propio stack de variables, por lo que la línea *x++;* dentro de la cláusula *pid==0* sólo va a afectar a la variable *x* ubicada en el stack del PCB del proceso hijo. Luego se imprimirá la variable *x* dentro del stack del proceso padre.

2.

Proceso 1 ┬───Proceso 2 ┬ Proceso 3 ┬Proceso 4 ┬Proceso 5 ─Proceso 6

│ │ │ │

│ │ │ └Proceso 7

│ │ │

│ │ ├Proceso 8 ─Proceso 9

│ │ │

│ │ └Proceso 10

│ │

│ ├Proceso 11 ┬Proceso 12 ─Proceso 13

│ │ │

│ │ └Proceso 14

│ │

│ ├Proceso 15 ─Proceso 16

│ │

│ └Proceso 17

│

├───Proceso 18 ┬ Proceso 19 ┬Proceso 20 ─Proceso 21

│ │ │

│ │ └Proceso 22

│ │

│ ├Proceso 23 ─Proceso 24

│ │

│ └Proceso 25

│

├───Proceso 26 ┬ Proceso 27 ─Proceso 28

│ │

│ └ Proceso 29

│

├───Proceso 30 ─ Proceso 31

│

└─── Proceso 32

3.

(a) y (c) Adjunto archivo "*Arbol de Abraham.cpp*"

b) Correr el programa varias veces nos muestra que no siempre se muestra la misma secuencia gracias a que es decisión del scheduler quién procesa en cada momento, por lo que no podríamos saber de esa forma qué proceso terminaría primero.

d) En la primera implementación, el padre podría no esperar nunca a sus hijos, por lo que éstos se convertirían en Zombies incluso habiendo terminado de ejecutarse, dejando una entrada ocupada en la lista de procesos donde su estatus estará guardado.

4.

La secuencia ”Proceso1-Terminaron todos-Proceso0-Proceso4-Proceso2-Proceso3” es posible ya que el proceso padre es el encargado de crear 5 hijos y esperar a que por lo menos uno termine para avisar erróneamente que han terminado todos. Esto se debe a que el scheaduler podría hacer que el proceso principal cree todos los procesos hijos y llegar a la línea "*wait(&status);*" donde al terminar alguno de los hijos (indistintamente cuál) habilite al padre a continuar su ejecución, imprimiendo la línea "Terminaron todos".

Adjunto código modificado en el archivo ej4.cpp