Modern Operating Systems | (4th Edition)

Problem

Consider the following piece of C code:

void main( ) {fork( );fork( );exit( );}

How many child processes are created upon execution of this program?

Step-by-step solution

Step 1 of 2

• Multitasking operating systems create a multitasking environment using child processes.

• A child process is a copy of the parent and inherits most of its attributes.

• A child process is created using a fork() statement which is a system call implemented in the kernel. It returns the process ID of the process just created.

Comment

Step 2 of 2

• Consider the C code given in the exercise.

• To find how many child processes were created when the program was executed follow below steps:-

o The main function calls fork() statement twice before exiting.

Figure 1: demonstrates the number of processes after the first fork call.

o After the second fork() call, both child1 and parent process make a copy of themselves. the status of child processes created is given using a tree structure below:-

27P
28P
29P
30P
31P
32P
33P
34P
35P
36P
37P
38P
39P
40P
41P

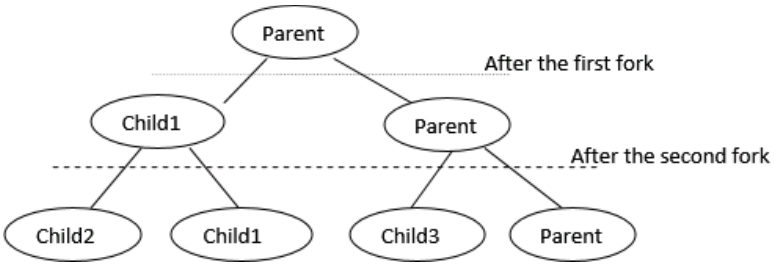

Figure 2: demonstrates the number of processes after the second fork call.

o From the above diagram it is clear that after the second fork, four processes exist. They are Parent, child1, child2 and child3. So, three child processes were created while the parent already existed.

Hence, when executed the given 'C' code creates 3 child processes.

Comment

Was this solution helpful? 0 0