﻿3.11 Including the initial parent process, how many processes are created by

the program shown in Figure 3.32?

16

3.12 Explain the circumstances under which the line of code marked

printf("LINE J") in Figure 3.33 will be reached.

The call to exec() replaces the address space of the process with the program specified as the parameter to exec(). If the call to exec() succeeds, the new program is now running and control from the call to exec() never returns. The line printf("Line J") would never be performed. However, if an error occurs in the call to exec(), the function returns control and therefor the line printf("Line J"); would be performed.

3.13 Using the program in Figure 3.34, identify the values of pid at lines A, B,

C, and D. (Assume that the actual pids of the parent and child are 2600

and 2603, respectively.)

A = 0, B = 2603, C = 2603, D = 2600

3.14 Give an example of a situation in which ordinary pipes are more suitable

than named pipes and an example of a situation in which named pipes

are more suitable than ordinary pipes.

Simple communication works well with ordinary pipes. For example, assume we have a process that counts characters in a file. An ordinary pipe can be used where the producer writes the file to the pipe and the consumer reads the files and counts the number of characters in the file.

3.15 Consider the RPC mechanism. Describe the undesirable consequences

that could arise from not enforcing either the “at most once” or “exactly

once” semantic. Describe possible uses for a mechanism that has neither

of these guarantees.

If an RPC mechanism cannot support either the “at most once” or “at least once” semantics, then the RPC server cannot guarantee that a remote procedure will not be invoked multiple occurrences. Consider if a remote procedure were withdrawing money from a bank account on a system that did not support these semantics. It is possible that a single invocation of the remote procedure might lead to multiple withdrawals on the server.

3.16 Using the program shown in Figure 3.35, explain what the output will

be at lines X and Y.

The child is a copy of the parent, any changes the child makes will occur in its copy of the data and won’t be reflected in the parent. As a result, the values output by the child at line X are 0, -1, -4, -9, -16. The values output by the parent at line Y are 0, 1, 2, 3, 4