3.12 Including the initial parent process, how many processes are created by the program below (also given in Figure 3.32)?

There would be 16 total processes due to each iteration of I (for i =0,1,2, and 3) calling fork on the current process.

3.14 Using the program below (also given in Figure 3.34), identify the values of pid and pid1 that are output at lines A, B, C, and D. Assume that the actual pids of the parent and child are 2600 and 2603, respectively.

Parent: 2600

Child: 2603

A: pid = 2600

B: pid1 = 2603

C: pid = 0

D: pid1 = 2600

Using the Linux Source Code Browser, find do\_fork(), the fundamental routine for creating a new process (i.e. the main fork-routine)

• Within do\_fork(), what is the purpose of the call to copy\_process()?

It looks like the copy\_process is stored in a task struct. It also seems that it is passes the same clone\_flags, stack\_start, child\_tidptr, and other information about the current process. It essentially makes a copy of the process, and if successful it kick-starts the copy and waits for it to finish using the Virtual Machine if required. It determines whether and which event to report to the ptracer. When called from thekernel thread or from CLONE\_UNTRACED is explicitly requested; otherwise it will report the event for the type of forking is enabled.

o Provide some detail as to what it actually does

• Specifically, how does this function guard against fork() bombs?

The function guards against fork bombs by determining wheter and which event to report to ptracer. “When called from kernel\_thread or CLONE\_UNTRACED is explicitly requested, no event is reported.” This will prevent an overload of forking and determine not to report it back to the program tracer.