The program generates a binary tree of processes. The input includes the number of levels in the tree. The maximum number of levels is set to five; however the program works for a general case of any number. If the number of arguments is not equal to two or the number of levels is greater than five or less than or equal to zero, the program exits. If the argument list is correct the program will print the header for the output. After the header has printed, the program begins generating processes.

Once in the loop, the initial process spawns its first child and checks to make sure the process spawned correctly. Before the second child is created, the program checks to make sure it is the parent process. If it is not, the first child process repeats the same procedure as its parent process from the start of the loop. If it is the parent process, the parent generates its second child and checks for errors. After the second child is created the parent process exits the loop, sleeps for a random amount of time (since each process has a different child ID, using the child ID as the seed for the random function will result in a different sleep time for each process) and prints its own process ID, its parents process ID and both child process IDs. If the process does not have any children, a 0 is printed in the appropriate place. Note that since it is possible for a parent process to exit before its children, some children may claim ID 1 as its parent. The second child will also repeat the same procedure as its parent process. This will continue until the number of desired levels has been created.