1. I believe that the system designers implemented signals that are only handled by the operating system so that there are overrides that can be called like KILL and STOP so that if the program isn’t responding properly or doing something dangerous then the operating system will be able to terminate it with out having to wait for it to finish.
2. The pause system call just proves to be much more efficient. An infinite while loop is constantly running, whether we want it to or not. The pause system call is simply waiting for some kind of response from the program so that it can continue with its function.
3. Other signals are masked in the signal handler because there are certain signals that will not run until the previous process has completed. Using a signal mask allows for a new process to run without waiting for the signal that is currently running to provide feedback or finish.
4. The goal of SIGALRM is to timeout the process and terminate the program, thus, if we did not unmask SIGALRM then it would not interrupt the program and execute the timeout function properly.