

Lab 9

Write a program to implement the following:

1. Catch the *SIGTERM* signal, ignore *SIGINT* and accept the default action for *SIGSEGV*. Later let the program be suspended until it is interrupted by a signal. Implement using `signal` and `sigaction`
2. Create a child process. Let the parent sleeps of 5 seconds and exits. Can the child send *SIGINT* to its parent if exists and kill it? Verify with a sample program.
3. Implement sleep using `signal` function which takes care of the following:
 - a. If the caller has already an alarm set, that alarm is not erased by the call to alarm inside sleep implementation.
 - b. If sleep modifies the current disposition of *SIGALRM*, restore it
 - c. Avoid race condition between first call to alarm and pause inside sleep implementation using `setjmp`.

Test the implementation of sleep by invoking it in various situations.

4. “Child inherit parent’s signal mask when it is created, but pending signals for the parent process are not passed on”. Write appropriate program and test with suitable inputs to verify this.