Simon Yoon
compOS

Linux, x86_64
signal, sig#, [elaboration]
1a) SIGINT, 2
1b) SIGSEGV, 11 , dereference the nullptr and attempt to assign a value to that memory, which theoretically results in undefined behavior, but will result in an access violation exception on Linux, x86_64.
1c) SIGKILL, 9
SIGSTOP, 19
1d) SIGTTIN, 21, to stop reading to terminal
SIGTTOU, 22, to stop writing to terminal
SIGCONT, 18, to resume process

2a) ps a | grep ‘a.out’
kill -s USR1 pid

2b) Because of the SA_NOMASK flag, the signal which triggered the handler will not be blocked. The handler does not mask the signal. sigprocmask() fetches no signal as the blocking bit is not set to allow signals. The struct does not fully define the signals masked. This should be altered in the presence of an infinite loop in order to not prevent the signal sent in case it was with the intention to stop the infinite loop. To resolve this issue, we could implement a longjmp to break out of the loop or to add specifications to the mask i.e sa.sa_mask = sigemptyset.. etc.