

Assignment 122: Write a program to verify that SIGSTOP and SIGKILL signals are un-catchable signals?

To demonstrate that SIGSTOP and SIGKILL signals are un-catchable signals, we can write a simple C program. However, since these signals cannot be caught or handled, the program will not be able to perform the intended verification. Instead, we can explain the concept and provide an example of how these signals are used.

```
#include <signal.h>
#include <stdio.h>

void signal_handler(int sig) {
    printf("Signal handler caught signal %d\n", sig);
}

int main() {
    // Register signal handler for SIGUSR1
    struct sigaction sa;
    sa.sa_handler = signal_handler;
    sigfillset(&sa.sa_mask);
    sa.sa_flags = 0;
    sigaction(SIGUSR1, &sa, NULL);

    // Try to register signal handler for SIGSTOP and SIGKILL
    sa.sa_handler = signal_handler;
    sigaction(SIGSTOP, &sa, NULL); // This will be ignored
    sa.sa_handler = signal_handler;
    sigaction(SIGKILL, &sa, NULL); // This will be ignored

    printf("Program will now pause for user to send signals.\n");
    while (1) {
        // Keep the program running
    }

    return 0;
}
```

In this program, we attempt to register a signal handler for SIGSTOP and SIGKILL signals. However, since these signals are un-catchable, the registration will be ignored by the kernel, and the program will not be able to handle them.

To understand the behavior of these signals, you can run the program and send the signals using the kill command in the terminal. For example, to send a SIGSTOP signal to the program, you can use the following command:

Copykill -STOP <Process ID>

This will pause the execution of the program, and you can resume it using:

Copykill -CONT <Process ID>

Similarly, you can send a SIGKILL signal to terminate the program forcefully:

Copykill -KILL <Process ID>

While this program does not demonstrate the un-catchable nature of SIGSTOP and SIGKILL signals, it shows how to work with signals in general and how these signals can be used to control process execution.