Assignment 124: Write a program that blocks the SIGTERM signal during execution of the SIGINT signal?

To block the SIGTERM signal during the execution of the SIGINT signal, you can modify the sigterm_handler function to ignore the SIGTERM signal when the SIGINT signal is being handled. Here's an updated version of the program:

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

static int sigterm_ignored = 0;

void sigint_handler(int sig) {
    printf("SIGINT signal caught. Shutting down gracefully...\n");
    // Perform necessary cleanup or save state here
}

void sigterm_handler(int sig) {
    if (sigterm_ignored) {
        printf("SIGTERM signal ignored during SIGINT handling.\n");
        return;
    }
    printf("SIGTERM signal caught. Terminating the program...\n");
    // Perform necessary cleanup or save state here
    exit(0);
}

int main() {
    // Register signal handlers
    signal(SIGINT, sigint_handler);
    signal(SIGTERM, sigterm_handler);

    // Your program's main logic here
    // ...

    return 0;
}
```

In this updated version, we introduce a global boolean variable sigterm_ignored to track whether the SIGTERM signal is being ignored during the execution of the SIGINT signal. Inside the sigterm_handler function, we first check the value of sigterm_ignored. If it is set to 1, we print a

message indicating that the SIGTERM signal is being ignored during the handling of the SIGINT signal and return from the function.

In the sigint_handler function, you can set sigterm_ignored to 1 to indicate that the SIGTERM signal should be ignored during the execution of the SIGINT signal. Once the SIGINT handling is complete, you can reset sigterm_ignored to 0 to allow the SIGTERM signal to be processed normally.

This way, the program will block the SIGTERM signal during the execution of the SIGINT signal by ignoring it temporarily.