

Assignment 123: Write a program to handle the SIGINT and SIGTERM signals?

To handle SIGINT (Interrupt) and SIGTERM (Termination) signals in a program, you can use the signal function or its more advanced version sigaction. Here's an example using signal:

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

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) {
    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 example, we define two signal handler functions, `sigint_handler` and `sigterm_handler`, which will be called when the corresponding signals are sent to the program. When the user presses Ctrl+C (SIGINT) or when the system initiates a clean termination (SIGTERM), the respective handler functions will be invoked.

In the `sigint_handler` function, we print a message indicating that a SIGINT signal has been caught and perform any necessary cleanup or save the current state of the program. In the `sigterm_handler` function, we print a message indicating that a SIGTERM signal has been caught and then terminate the program using the `exit` function.

You can replace the placeholder comments with your own cleanup or save state logic as needed.

