

Question: Multithreading and Signals in a Warehouse Robot Simulation

Scenario:

You are tasked with developing a simulation for an automated warehouse system. The warehouse has several robots (threads), each responsible for performing a repetitive task—such as moving items. These robots operate under the supervision of a central controller, which can issue new tasks to all robots or initiate an emergency stop through signals.

Your task is to implement the robot simulation where:

- Each robot thread performs a repetitive task (e.g., computing the sum of integers up to a given number) and prints the result.
- The system responds to two types of signals:
 1. **SIGINT**: Simulates an emergency stop, which signals all robots to stop and terminate.
 2. **SIGUSR1**: Simulates the controller issuing a new task for the robots, which requires all robots to stop their current task and begin a new one with a different number to sum.

Requirements:

1. Create three robot threads, each calculating the sum of integers from 1 to a specified number (initially 100).
2. Handle two signals:
 - **SIGINT**: When received, all robots must stop their current task and terminate, printing a termination message.
 - **SIGUSR1**: When received, the robots should switch to summing integers from 1 to a new task value (e.g., 50) and continue.
3. Each thread should keep running indefinitely until an emergency stop signal is received.

Specifics:

- **Robot Task**: Sum the numbers from 1 to a specific number, print the sum, then repeat indefinitely until a new task is given or termination occurs.
- **Test Cases**: Simulate the behavior of the system by sending different signals:
 - **Test Case 1**: Send a SIGUSR1 signal to change the task while robots are running.
 - **Test Case 2**: Send a SIGINT signal to stop all robots.

Test Case 1: Regular Operation with Task Update

- **Input**: Initially, all threads calculate the sum up to 100. After a SIGUSR1 signal, all robots will switch to calculating the sum up to 50.
- **Expected Output**:

```
Robot 1: Sum = 5050
```

```
Robot 2: Sum = 5050
Robot 3: Sum = 5050
Robot 1: Received new task, Sum = 1275
Robot 2: Received new task, Sum = 1275
Robot 3: Received new task, Sum = 1275
```

Test Case 2: Emergency Stop

- **Input:** After running for some time, a SIGINT signal (Ctrl+C) is received to stop all robots.
- **Expected Output:**

```
Robot 1: Emergency stop received. Terminating.
Robot 2: Emergency stop received. Terminating.
Robot 3: Emergency stop received. Terminating.
```

Explanation of the Code:

1. **Thread Creation:** Three threads (robots) are created using `pthread_create()`. Each thread runs the `robot_operation()` function.
 2. **Signal Handling:**
 - **SIGINT:** When SIGINT (Ctrl+C) is received, the `handle_sigint()` signal handler sets the global `running` flag to 0, which causes all threads to terminate gracefully.
 - **SIGUSR1:** When SIGUSR1 is received, the `handle_sigusr1()` signal handler updates the global `task_value` to 50, making all robots change their task and start summing numbers from 1 to 50.
 3. **Thread Operation:** Each robot continuously computes the sum of integers up to the current `task_value` and prints the result. The robots check the `running` flag to determine if they should terminate.
 4. **Thread Termination:** When SIGINT is received, the program exits all threads cleanly, printing a termination message for each robot.
-

How to Test:

1. **Run the Program:** Compile and run the program:

```
gcc -o warehouse_simulation warehouse_simulation.c -pthread
./warehouse_simulation
```

2. **Send SIGUSR1:** To simulate a task update, send the SIGUSR1 signal:

```
kill -SIGUSR1 <PID>
```

You will see the robots start summing up to the new task value (50).

3. **Send SIGINT (Ctrl+C):** To stop the robots, press Ctrl+C. This will send a SIGINT signal to terminate all the robots.