Chandransh Singh

22CS30017

Assignment 5: Non-Blocking Task Queue Server for Distributed Job Processing

> ## Handling Client Misbehaviors in the Task Queue System

1. Client Sends Repeated GET_TASK Without Completing Previous Task

- When a client requests a task, the server first checks if it already has a pending task
- If so, the server re-sends the same task rather than assigning a new one
- This prevents task hoarding by clients and ensures fair distribution of work

2. Client Sends GET_TASK and Then Does Not Respond

- Each task has an assignment_time that records when it was assigned
- The server periodically checks if any tasks have been assigned but not completed for more than 30 seconds
- If a timeout is detected, the task is marked as unassigned and returned to the pool
- This function is called in both the main server loop and client handler function

3. Client Connects to Server and Does Not Respond Further

_ The server implements a client inactivity timeout mechanism to detect and handle unresponsive clients
_ Each client connection tracks a `last_activity` timestamp that records when the client last communicated with the server
_ The server periodically compares the current time with this timestamp using `CLIENT_IDLE_TIMEOUT` constant

4. Client Connects, Sends GET_TASK, and then Closes Connection Arbitrarily

- When a client disconnects abruptly, the child process terminates, generating a SIGCHLD signal
- The SIGCHLD handler detects this termination and releases any tasks assigned to that client
- It also decrements the active client count, ensuring proper server termination when all tasks are complete
- The handler uses the proper `waitpid()` call with `WNOHANG` to prevent creating zombie processes