Requirements Document for Test Generation and Coverage System

1 Overview

This system generates unit tests for C/C++ files in a given repository, measures test coverage before and after test generation, and removes low-coverage tests. The pipeline includes multiple stages:

- **Cloning the repository**
- **Generating test files**
- **Compiling and cleaning up tests**
- **Measuring coverage**
- **Removing tests that do not increase coverage by over 2.5%**

2 System Requirements

To run this system, the following software and libraries are required:

2.1 Operating System

- **Linux** (for compatibility with MPI system as per the requirements).
- Alternatively, **Windows** or **macOS** can be used but will require modifications for MPI compatibility.

2.2 Python Version

- Python **3.7** or above.

2.3 Required Libraries

- **openai**: For interacting with the OpenAI API to generate unit tests.
- **subprocess**: For running shell commands such as `git` and `gcc`.
- **os**: For interacting with the file system to navigate and manipulate directories and files.

These can be installed via `pip`:

```
pip install openai>=1.0.0
```

2.4 Required Tools

- **Git**: To clone the repository.
- **C/C++ Compiler**: Required for compiling the generated test files and ensuring they compile successfully. On Linux, you can use GCC:
  - Install: `sudo apt install build-essential`
- **gcov**: For measuring test coverage on C/C++ code.
  - Install: `sudo apt install gcov`

2.5 Environmental Variables

- **OPENAI_API_KEY**: The environment variable holding your OpenAI API key. This key is used to interact with the OpenAI API for generating unit tests.

Example:

```
export OPENAI_API_KEY="your-api-key-here"
```

2.6 Optional Tools

- **Make**: If using `Makefiles` for easier compilation of C/C++ tests.
  - Install: `sudo apt install make`

3 File Structure

The following file structure is expected for this system:

```
/<project-root>
    /cloned_repo                # Directory where the repo is cloned
    test_gen.py                 # Main test generation script
    test_coverage_comparison.py # Script to measure and compare coverage
    compile_and_cleanup.py      # Script to compile and clean up tests
    requirements.txt            # Python dependencies
    README.md                   # Documentation about the system
```

4 How to Run the System

4.1 Install Python Dependencies

Create a virtual environment for your project and install required Python libraries:

```
python -m venv .venv
source .venv/bin/activate  # On Windows use .venv\Scripts\activate
pip install -r requirements.txt
```

4.2 Clone the Repository

Ensure that the repository URL you want to test is set in `test_gen.py` under the `repo_url` variable:

```
repo_url = 'https://github.com/hpcg-benchmark/hpcg.git'
```

4.3 Run the Test Generation Pipeline

Run the main script to start the process:

```
python test_gen.py
```

The program will:

- Clone the specified repository.
- Find `.cpp`/`.c` files and generate test files.
- Generate unit tests using the OpenAI API.

- Measure initial coverage before any tests are created.
- Compile the test files, removing those that fail.
- Measure the final coverage after tests are generated and cleaned up.
- Report the improvement in coverage and remove low-coverage tests.

4.4 Check Results

After running, the program will print:

- The total number of tests processed.
- The total number of tests removed for low coverage.
- The coverage improvement after the test generation.

5 Maintenance

- **Coverage Script (`test_coverage_comparison.py`)**: Maintains and compares test coverage before and after test generation.
- **Cleanup Script (`compile_and_cleanup.py`)**: Compiles the tests and removes those that do not compile successfully.

6 Troubleshooting

- **Missing dependencies**: Ensure that the system has all required tools installed (Git, C/C++ compilers, etc.).
- **OpenAI API Key**: Make sure the OpenAI API key is set in your environment as `OPENAI_API_KEY`.
- **Test Failures**: If tests fail to compile, ensure that the files are correct and that the required compilers are installed.

7 Appendix

7.1 requirements.txt

```
openai>=1.0.0
```

7.2 GitHub Repository Example

An example repository URL for testing could be:

`https://github.com/hpcg-benchmark/hpcg.git`

Make sure to adjust the repository URL to match your desired project.