

Проект: Виртуальная Машина и Тестирование

Этот проект включает в себя реализацию ассемблера, интерпретатора и автоматизированного тестирования виртуальной машины для выполнения инструкций.

Структура проекта

- **assembler.py** — Сборщик инструкций из XML в бинарный файл.
 - **interpreter.py** — Интерпретатор для выполнения бинарного файла.
 - **test_vector_max.py** — Тестовая программа для проверки выполнения операций `max()` над векторами.
 - **test_input_test.xml** — Входной XML-файл, описывающий инструкции.
 - **test_program_test.bin** — Скомпилированный бинарный файл, созданный ассемблером.
 - **result_test.xml** — Выходной файл интерпретатора, содержащий память после выполнения.
-

Файлы и их функции

1. **assembler.py**

Этот скрипт читает входной XML-файл, преобразует инструкции в бинарный формат и записывает результат в файл.

Пример использования:

```
python assembler.py test_input_test.xml test_program_test.
```

`bin test_log_test.xml interpreter.py` Выполняет бинарный файл, изменяет память виртуальной машины согласно инструкциям и записывает результат в XML-файл. Пример использования:

```
python interpreter.py test_program_test.bin result_test.xml 0:10
```

`test_vector_max.py` Автоматически генерирует тестовый XML-файл, компилирует его в бинарный файл и выполняет интерпретатор. Если оба этапа успешны, выводится:

```
Все тесты пройдены успешно.
```

Пример запуска: `test_input_test.xml` XML-файл, описывающий входные инструкции для виртуальной машины. Генерируется автоматически тестовым скриптом. `test_program_test.bin` Скомпилированный бинарный файл, созданный ассемблером. Содержит инструкции для выполнения. `result_test.xml` XML-файл, содержащий финальное состояние памяти виртуальной машины после выполнения программы. Как запустить тесты Убедитесь, что все файлы находятся в одной директории. Запустите тестовый скрипт:

```
python test_vector_max.py
```

Если всё работает корректно, вы увидите: Все тесты пройдены успешно.

Пример работы

Генерация входного файла: Входной XML-файл (test_input_test.xml) автоматически создаётся тестовым скриптом. Ассемблирование:

assembler.py компилирует инструкции из XML в бинарный файл. Исполнение:

interpreter.py выполняет бинарный файл и сохраняет результат в XML. Тестирование: Тестовый скрипт проверяет успешность всех этапов.