

Colon

Второй этап

Реализованный функционал:

1) BEGIN UNTIL:

- в модуле ExecuteCommand написана реализация BEGIN UNTIL команды
- В цикле loop сначала выполняются все команды из тела цикла, затем из стека берем значение которое определяет нужно ли продолжить цикл. Если значение = 0, повторяем цикл, иначе выходим из цикла

2) CREATE ... ALLOT CELLS

- В модуле ExecuteCommand написана реализация операций create, allot, cells. В дальнейшем планируется перенести в отдельный модуль
- Команда Create
 - i) создает новую переменную или массив с переданным именем в памяти. В памяти появится запись (переданное имя, [])
- Команда Cells
 - i) Вычисляет размер для заданного количества ячеек в байтах, результат кладется в стек.
- Команда Allot
 - i) Выделяет память для переменной или массива, указанной в памяти. Верхний элемент массива делится на размер одной ячейки для получения количества ячеек. К массиву добавляется нужное количество нулей

Реализация и взаимодействие

Парсинг команд

- В модуле MainParser использовал Megaparsec для преобразования текста в команды типа Command. Для create, beginUnitl, allot, cells написал свои парсеры

Тестирование

Для тестов использовал библиотеки Tasty, HUnit, QuickCheck

Тестами покрыто:

- 1) Создание массива/переменной и выделение памяти под массив
- 2) Работа цикла BEGIN ... UNTIL

Результаты:

```
BEGIN UNTIL Test: OK
CREATE myarray 10 CELLS ALLOT: OK

All 2 tests passed (0.00s)
*** Exception: ExitSuccess
```

```
Создана переменная/массив: 'myarray'
Значение 10 добавлено в стек
Размер ячеек: 80
Выделено 80 байтов для 'myarray'
Размер массива: 10 ячеек
```

Личные выводы

Научился работать с памятью и стеком. Понял что нужно разделять что может храниться в памяти, а что в стеке.

Возникшие проблемы и сложности:

Проблемы возникли с парсингом и с выделением памяти. Как я понял некоторые команды конфликтовали между собой и была неразбериха. Так

же пока проблема и сложность реализации команд над массивом, над этим продолжаю работу на данный момент