

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

3) Арифметические операции над числами с плавающей точкой

- В модуле ArithmeticOperations написана реализация операций F+, F-, F/, F*, F>S, S>F.
- Команды F+, F-, F/, F*, F>S, S>F

- i) Реализация этих команд очень похожа друг на друга. Мы сначала получаем текущий стек и память, далее пытаемся извлечь два верхних элемента из стека и если оба числа FloatValue, то выполняем соответствующую операцию(+, -, *, /), а результат кладем в стек. В реализации команд F>S, S>F, тоже самое только извлекаем 1 элемент из стека, пытаемся преобразить соответственно и результат кладем в стек

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
- В парсер добавил case для push'а значения в стек, так как возникали проблемы с пушем

Тестирование

Для тестов использовал библиотеки Tasty, HUnit, QuickCheck

Тестами покрыто:

- 1) Создание массива/переменной и выделение памяти под массив
- 2) Арифметические операции над числами с плавающей точкой

Результаты:

```
Colon Language Tests
F+: OK
F-: OK
F*: OK
F/: OK
FToS: OK
SToF: Создана переменная/массив: 'myarray'
Размер ячеек: 80
Выделено 80 байтов для 'myarray'
Размер массива: 10 ячеек
OK
CREATE myarray 10 CELLS ALLOT: OK
All 7 tests passed (0.01s)
```

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. Также пока проблема и сложность реализации команд над массивом, над этим продолжаю работу на данный момент. Также возникла проблема с тестированием создания массива, так как там нужно проверять не только стек, но стек и память, для этого написал отдельный тест

Также проблемы возникли с парсингом пуша Int и Float значений. Push парсилась в зависимости от положения команды в cmdParser. Т.е если PushInt стоял выше чем PushFloat, то при пуше значения 3.4, он не понимал что я пытаюсь запустить float и наоборот. Решил проблему case и методом который ищет точку в значении которое пушим.