

VOWEL

string manipulation language

ΛΟΜΕΓ

Coby Simler

Language Guru

Lex Mengenhauser

System Architect

Vikram Rajan

Tester 

Aidai Beishekeeva

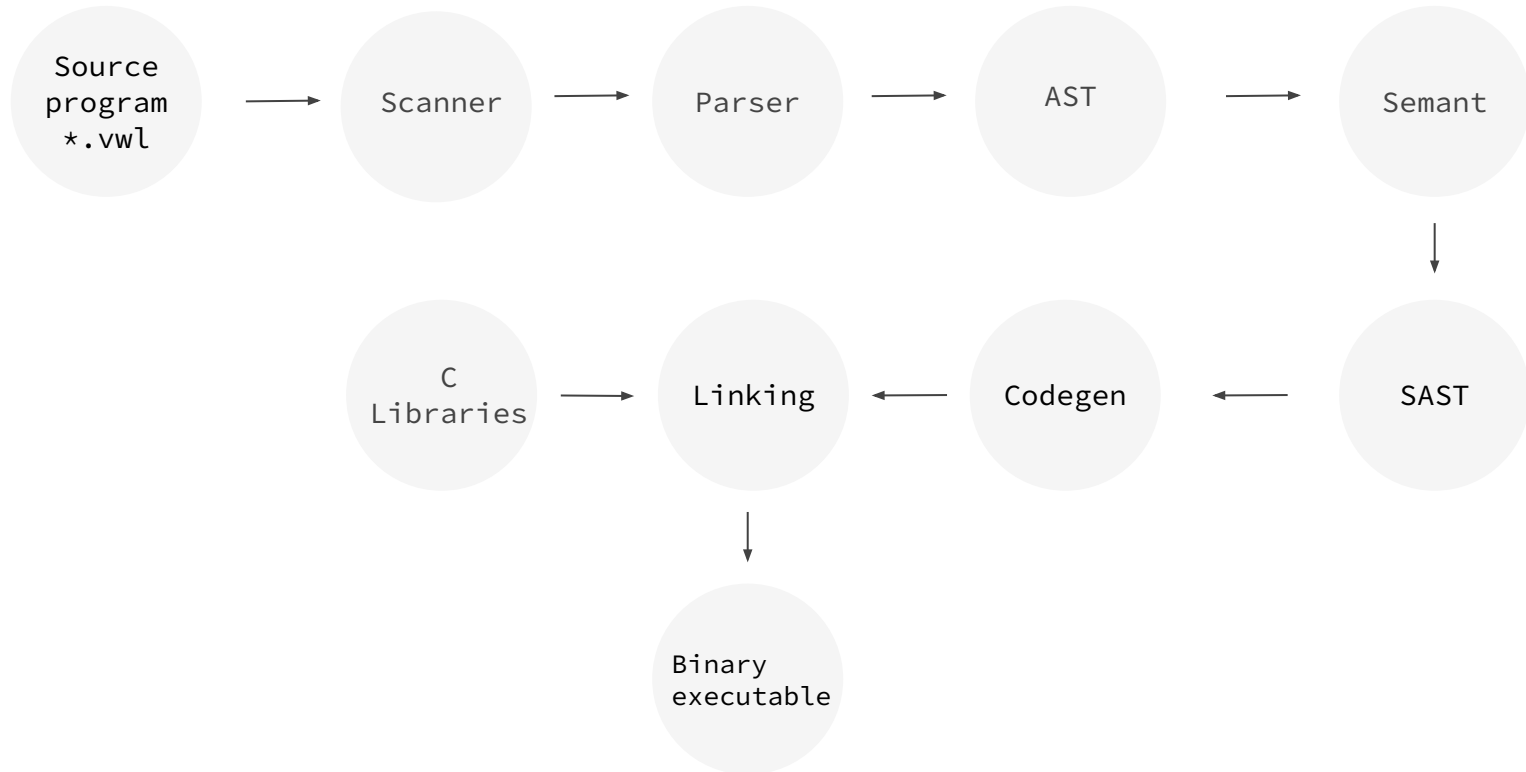
Manager



Overview

- High-level, imperative, statically-typed programming language intended to be used to iterate over, operate on, and manipulate large text inputs.
- Vowel aims to provide users handling large texts with a more intuitive syntax and complex functionality for manipulating strings.
- Mix of Python & C.

Architectural design



Data types

Primitive

Int	Arithmetic operations
Float	Arithmetic operations
Boolean	Logical operations
String	String operations

Array

Collection of primitive elements of the same type.
Support iteration, access by index, reassignment of values

Variable declarations

- In Vowel, variables must be both declared and assigned an initial value:

```
int x = 4;
```

```
bool b = true;
```

```
string s = "test string";
```

```
int[] arr = [1, 4, 10]
```

- Scope of variables are defined by curly braces, '{ }'
 - Global variables are those defined outside a method
 - Local variables are defined within the curly braces of a method

No main method

C

```
int main() {  
    printf("Hello World");  
}
```

Vowel

```
printstr("Hello World");
```

OR

```
string hello() {  
    printstr("Hello World");  
}
```

Array

Array Type

Array Literal

```
int[] arr = [1, 2, 3, 4];  
string[] str_arr = ["first", "second", "third"];
```

```
int x = arr[2];  
print(x); // Prints 3
```

Array Access

Array Update

```
str_arr[1] = "fourth";  
print(str_arr[1]); // Prints fourth
```

Strings

String length

```
string name = "vowel";  
int l = len(name); // 5
```

Slice

```
string name = "vowel is great!";  
string new_l = slice(name, 8, 14);  
// great
```

Concatenation

```
string name = "vowel is" + "great!";  
// "vowel is great"
```

Multiplication

```
string name = "hello!" * 3;  
// "hello!hello!hello!"
```

Increment

```
string name = "Bob";  
name += " Smith"; // "Bob Smith"
```

Difference

```
string s1 = "My name is Vowel";  
String s2 = "My name is C";  
  
String[] res = s1 - s2;  
// ["Vowel"]
```

Intersection

```
string s1 = "My name is Vowel";  
String s2 = "My name is C";  
  
String[] res = s1 & s2;  
// ["My", "name", "is"]
```


Testing

Test suite

```
fail-assign1.err
fail-assign1.vwl
fail-assign2.err
fail-assign2.vwl
fail-assign3.err
fail-assign3.vwl
fail-dead1.err
fail-dead1.vwl
fail-dead2.err
fail-dead2.vwl
fail-expr1.err
fail-expr1.vwl
fail-expr2.err
fail-expr2.vwl
fail-expr3.err
fail-expr3.vwl
fail-float1.err
fail-float1.vwl
fail-float2.err
fail-float2.vwl
fail-for1.err
fail-for1.vwl
fail-for2.err
fail-for2.vwl
fail-for3.err
fail-for3.vwl
fail-for4.err
fail-for4.vwl
fail-for5.err
```

Example Test File

```
test-add1.vwl x test-strlen.mc
tests > test-add1.vwl
1 int add(int x, int y)
2 {
3     return x + y;
4 }
5
6 print( add(17, 25) );
7
8
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
./testall.sh
test-add1...OK
test-arith1...OK
test-arith2...OK
test-arith3...OK
test-array-access...OK
test-cobytest...OK
test-decl-infunc...OK
test-declare-array...OK
test-fib...OK
test-float1...OK
test-float2...OK
test-float3...OK
test-for1...OK
test-for2...OK
test-func1...OK
test-func2...OK
test-func3...OK
test-func4...OK
test-func5...OK
test-func6...OK
test-func7...FAILED
./vowel.native failed on ./vowel.native tests/test-func7.mc > test-func7.ll
```

- Add tests to test all language functionality
- Run all tests at once with script
- Check Failed tests to find bugs

Demo

```
string getLongestSharedWord(string a, string b){
    string[] sharedWords = a & b;
    int longestWordLen = 0;
    string longestWord = "";
    int i = 0;
    for ( ; i<2; i=i+1){
        int currentlen = len(sharedWords[i]);
        if (currentlen > longestWordLen){
            longestWordLen = currentlen;
            longestWord = sharedWords[i];
        }
    }
    return longestWord;
}
```

```
string x = getLongestSharedWord("abra cadabra
baba","babra cadabra baba");
printstr(x);
```

```
string order1 = "Your receipt contains: Jameson |
                Glenlivet | Seagrams | Fireball |
                Glenfiddich | Glenmorangie";
string order2 = "Your receipt contains: Glenmorangie |
                Oban | Glenlivet | Glenfiddich |
                Bowmore | Laphroaig";
```

```
string[] unique1 = order1 - order2;
int i = 0;
for ( ; i < 3 ; i=i+1 ){
    printstr(unique1[i]);
}
```

Future work

- Union operator

```
string s = "Jameson, Glenlivet";  
string s2 = "Seagrams, Jameson";  
string[] union = s | s2;  
// ["Jameson", "Glenlivet", "Seagrams"]
```

- Slicing with delimiters

```
string s = "the firework will explode at 3:00PM";  
string when = s["at ";0:] // "3:00PM"  
string vikWears = "gucci shirts gucci belts gucci  
tie";  
string noCashLeft = vik["gucci ";2:] // "tie"
```