Stitch Language Proposal

Daniel Cole, Megan Skrypek, Rashedul Haydar, Tim Waterman
dhc2131, ms4985, rh2712, tbw2105

September 30, 2015

Motivation

Most "modern" programming languages trace their origins back decades to before the advent of cheap, general purpose multicore CPUs. They were designed for a distinctly mono-threaded environment. While libraries and enhancements to mainstay languages such as C/C++ and Java have added multithreading capabilities, it remains in many ways bolted on kludge. While newer frameworks such as Node.js provide more integral support for asynchronous operations, they lack the depth of support and power of a fully compiled language. With Stitch, we aim to build a language that has the power and flexibility of a fully compiled C style language, while having native threading support for modern multi-threaded applications. Our goal is to create a translator from Stitch to C.

Description

Stitch is inspired by C, which has a very well known syntax, and has been one of the most widely used languages since it was released over forty years ago. Stitch is a general purpose language that supports all standard mathematical and logical operations. Like C, Stitch is strongly typed, and whitespace does not matter.

In addition to the standard C primitive types (`int, double, char`, etc.), Stitch has native support for the `string` type. This includes concatenation, and a inbuilt length operator. Stitch also has support for the `bool` type.

```
//Stitch comments are similar to C comments
//this is a single line comment
/*
You can also have multi-line comments
*/

//functions are declared using the 'def' keyword, like Python
def int main() {

  //the var keyword declares a variable
  var int x = 7;
  //booleans are a primitive data type
  var bool b = true;

  //strings are first class citizens in Stitch
  var string s = "This is a String\n";
  var string h = "Stitch also supports " + "string concatenation!";
  var unsigned long l = lengthof(h);

  let double PI = 3.14; //let is used to define constants

  return 0;
}
```

What sets Stitch apart is its native support for multithreading using the `async` keyword. This keyword can be applied to any function call, as well as to any loop construct. When called in this way, functions and loops will run in their own thread.

```
/*
async keyword: used on function calls and on loop constructs, to
make the loop execute asynchronously
*/

def int main() {

  var int even_sum = 0;
  var int odd_sum = 0;

  //adds up all the even numbers from 0 to 100 million
  async for(var int i = 0; i < 100000000; i+=2) {
    even_sum += i;
  }

  //adds up all the odd numbers from 0 to 100 million
  async for(var int i = 1; i < 100000000; i+=2) {
    odd_sum += i;
  }

  printf("The sum of all values is %d", even_sum + odd_sum);

  return 0;
}
```

Example Program

```
//example program goes here
```