PARROT - REGISTERS

http://www.tutorialspoint.com/parrot/parrot_registers.htm

The current Perl 5 virtual machine is a stack machine. It communicate values between operations by keeping them on a stack. Operations load values onto the stack, do whatever they need to do and put the result back onto the stack. This is easy to work with, but it is slow.

To add two numbers together, you need to perform three stack pushes and two stack pops. Worse, the stack has to grow at runtime, and that means allocating memory just when you don't want to be allocating it.

So Parrot is going to break the established tradition for virtual machines, and use a register architecture, more akin to the architecture of a real hardware CPU. This has another advantage. We can use all the existing literature on how to write compilers and optimizers for register-based CPUs for our software CPU!

Parrot has specialist registers for each type: 32 IV registers, 32 NV registers, 32 string registers and 32 PMC registers. In Parrot assembler, these are named I1...I32, N1...N32, S1...S32, P1...P32 respectively.

Now let's look at some assembler. We can set these registers with the set operator:

```
    set I1, 10
    set N1, 3.1415
    set S1, "Hello, Parrot"
```

All Parrot ops have the same format: the name of the operator, the destination register and then the operands.