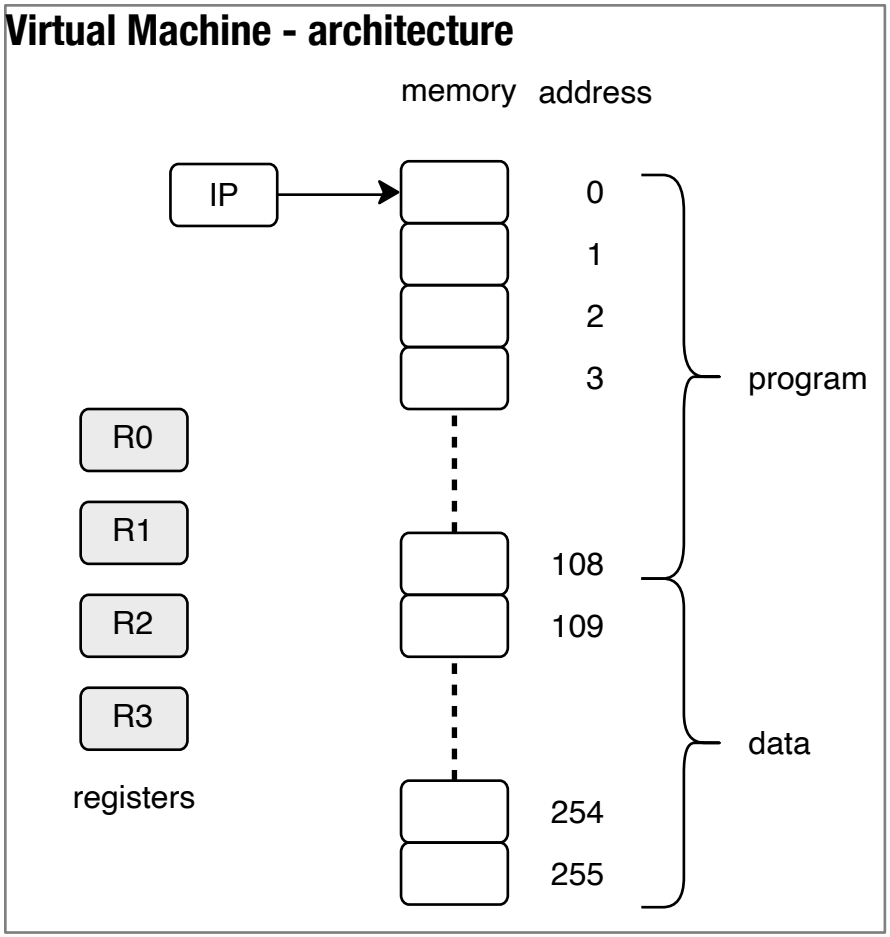
Virtual Machine - architecture



Virtual Machine - op codes

Name	Code	Format	Action	Example	Equivalent
hlt	1	--	Halt program	hlt	sys.exit(0)
ldc	2	rv	Load constant	ldc R0 99	R0 = 99
ldr	3	rr	Load register	ldr R0 R1	R0 = memory[R1]
cpy	4	rr	Copy register	cpy R0 R1	R0 = R1
str	5	rr	Store register	str R0 R1	memory[R1] = R0
add	6	rr	Add	add R0 R1	R0 = R0 + R1
sub	7	rr	Subtract	sub R0 R1	R0 = R0 - R1
beq	8	rv	Branch if equal	beq R0 99	if (R0==0) PC = 99
bne	9	rv	Branch if not equal	bne R0 99	if (R0!=0) PC = 99
prr	10	r-	Print register	prr R0	print(R0)
prm	11	r-	Print memory	prm R0	print(memory[R0])

architecture.py

```
NUM_REG = 4  # number of registers
RAM_LEN = 256  # number of words in RAM

OPS = {
    "hlt": {"code": 0x1, "fmt": "--"},  # Halt program
    "ldc": {"code": 0x2, "fmt": "rv"},  # Load value
    "ldr": {"code": 0x3, "fmt": "rr"},  # Load register
    "cpy": {"code": 0x4, "fmt": "rr"},  # Copy register
    "str": {"code": 0x5, "fmt": "rr"},  # Store register
    "add": {"code": 0x6, "fmt": "rr"},  # Add
    "sub": {"code": 0x7, "fmt": "rr"},  # Subtract
    "beq": {"code": 0x8, "fmt": "rv"},  # Branch if equal
    "bne": {"code": 0x9, "fmt": "rv"},  # Branch if not equal
    "prr": {"code": 0xA, "fmt": "r-"},  # Print register
    "prm": {"code": 0xB, "fmt": "r-"},  # Print memory
}

OP_MASK = 0xFF  # select a single byte
OP_SHIFT = 8  # shift up by one byte
OP_WIDTH = 6  # op width in characters when printing
```

For loop

```
# Count up to 3.
# - R0: loop index.
# - R1: loop limit.
ldc R0 0
ldc R1 3
loop:
prr R0
ldc R2 1
add R0 R2
cpy R2 R1
sub R2 R0
bne R2 @loop
hlt
```