

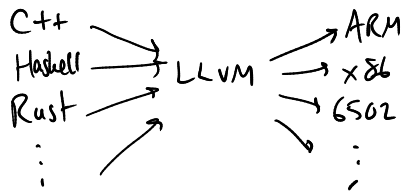
Virtual Machine — imaginary computer/machine that we can write code for, but is ultimately simulated on some underlying machine.

Turing — Universal Turing machine — can simulate any other.

- Java Virtual Machine

- Microsoft .NET CLI
 Common lang. interface — C#, F#, ...

- LLVM
 (low-level virtual machine)



- Wine

- WSL

- VirtualBox, VMWare

- Docker

Hack Virtual Machine.

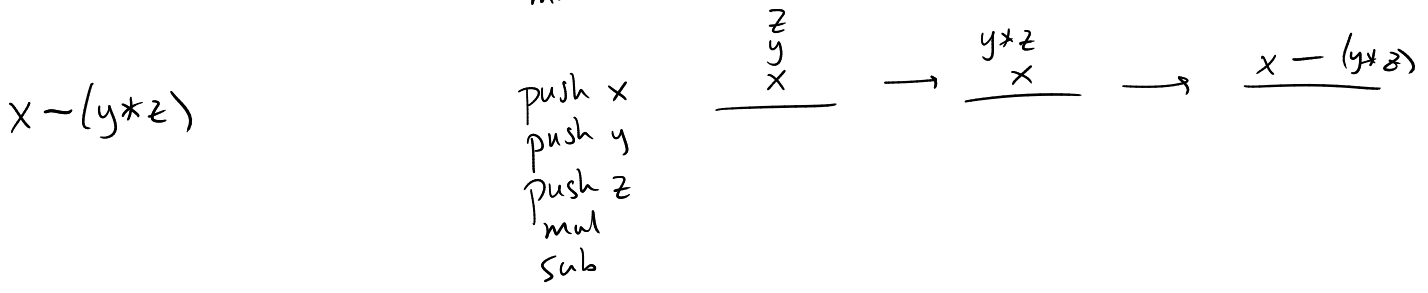
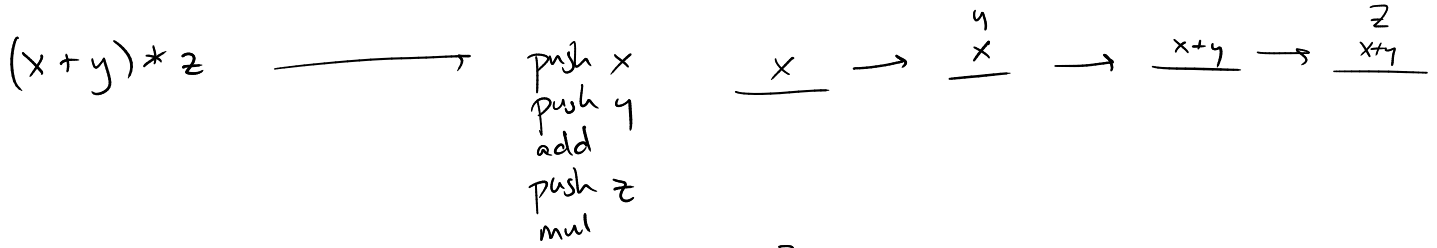
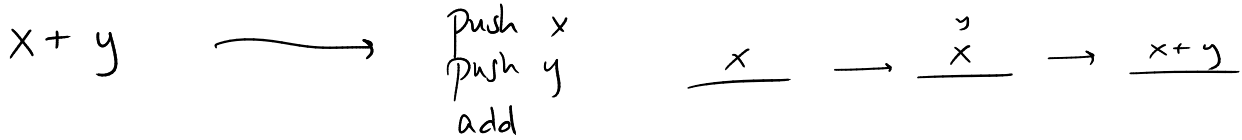
Stack machine — everything centers around a stack.

- Arithmetic, logic, etc. operations on stack.
- Saving/loading from memory always to/from stack.
- Functions are saved on stack.

Commands

- Arith., logic + comparison
 - Memory access ←
 - Branching
 - Function calls
- } Proj. 7, today + Thurs.
 } Proj. 8, next week.

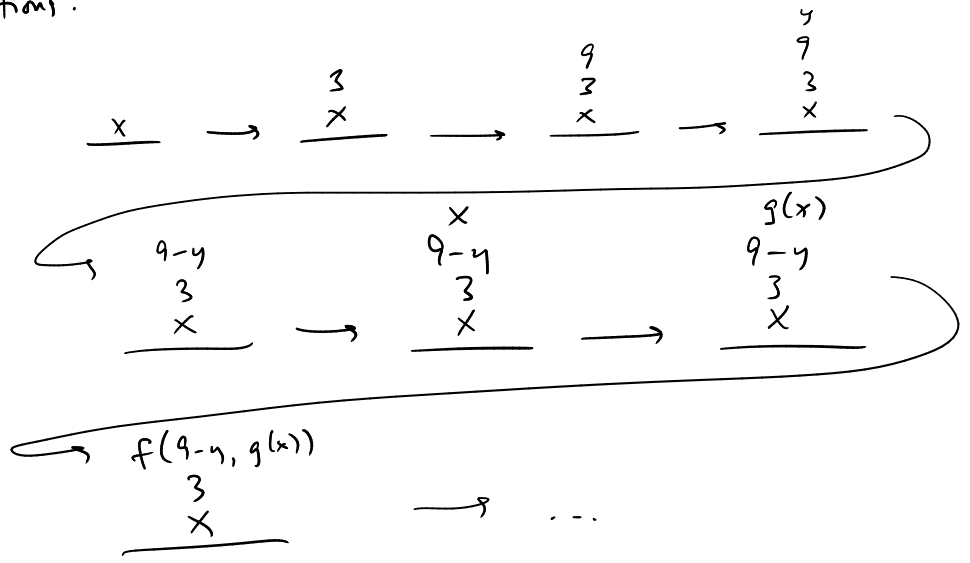
Stack Arithmetic



Can always push in order $L \rightarrow R$ — only question is when to do the operations.

$x + 3 * f(9 - y, g(x))$

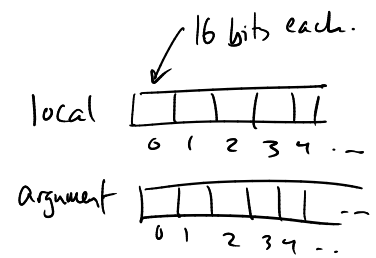
- push x
- push 3
- push 9
- push y
- sub
- push x
- call g
- call f
- mul
- add



Hack VM memory

- Memory consists of a collection of segments.

eg-



- Every push or pop operates to/from a particular segment.

- push <segment> <index>.

eg. push local 2. = read from index 2 of local seg. + push onto stack.

- pop <segment> <index>.

Segments

Constant — push-only, push constant 3 = push value 3.

Argument — holds arguments passed to current function.

local — holds local variables of current function.

temp — temporary storage.

static — storage for static (ie. global) vars.

this, that, pointer — next week.

def f(x, y):

z = x + 2

p = (z > 3) and (y < x)

return p.

High-level

VM code

// x + 2

push argument 0

push constant 2

add

// z = ...

pop local 0

// z > 3

push local 0

push constant 3

gt

// y < x

push argument 1

push argument 0

lt

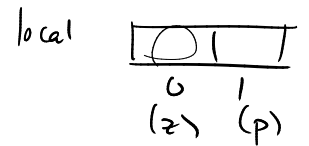
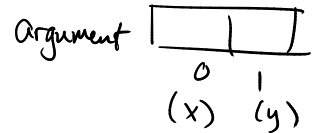
and

pop local 1

// p = ...

push local 1

return



Translating VM Code \rightarrow Assembly?

1. push constant 3.

@ 3

D = A

@ SP

A = M

M = D

@ SP

M = M + 1

@ SP = stack pointer = address of top of stack — address of first empty mem. loc. above the stack.

