

VM instructions for branching & functions

- label name \rightarrow ~~(label)~~ (functionName\$label)

Functions

Recall: 3 VM instructions.

- function name m \leftarrow # locals
- call name n \leftarrow # args
- return.

Bootstrap code @ beginning!

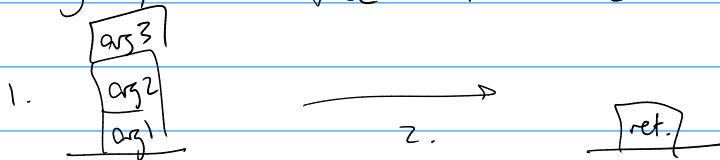
- SP = 256
- call "Sys.init".

\downarrow
Sys.init is provided by OS, does some extra setup, then calls 'main'.

How to call a function.

- From POV of the caller:

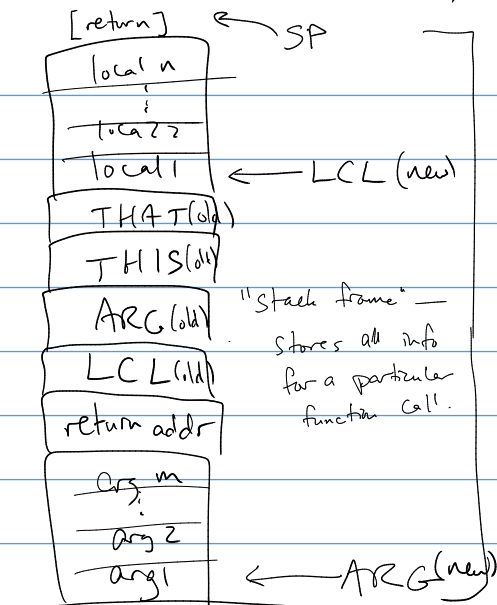
1. Push arguments on the stack
2. Call the function.
3. When the function returns, the arguments will be gone, and the ret. value will be on top of stack.



- From POV of called function (callee):

1. The argument segment contains the arguments.
2. The local segment is filled w/ the right number of 0's.
3. This that, temp are undefined, but I am free to use them.
4. Before returning, push return value on the stack
5. return.

How do we actually make this work?



Call name m

push @ret.

push LCL, ARG, THIS, THAT

LCL = SP

ARG = SP - 5 - m

Jump to vmfile.name
(ret)

pp. 155-161

return

frame, ret
temp var. eg.
R13, 14.

frame = LCL

ret = value @ (frame - 5)

pop from stack 4 put
value @ ARG.

SP = ARG + 1

THAT = value @ (frame - 1)

THIS = value @ (frame - 2)

etc ARG, LCL.

function name n

- (vmfile.name)

push 0 n times.