

13-stack-notes

The Stack

Agenda

- 0. Re-Orienting
- 1. Subroutines
- 2. Registers and Subroutines
- 3. Recursion

Reading:

- 3.2.0, 3.2.1: general background on ISA
 - 3.4.0, 3.4.1: addressing---but for the more general x86/IA32 case
 - 3.4.2: discussing the stack!
 - 4.1.4, 4.1.5: Y86 examples
-

0. Re-Orienting

The computer scientist.... all the way down to the laws of physics

the ISA

manipulating the registers and such with instructions

assembling the instructions into sequences of machine code

- by hand
- or with an assembler

processor status

arithmetic and logic

condition codes

jumps

comparisons

built-in stack

1. Subroutines

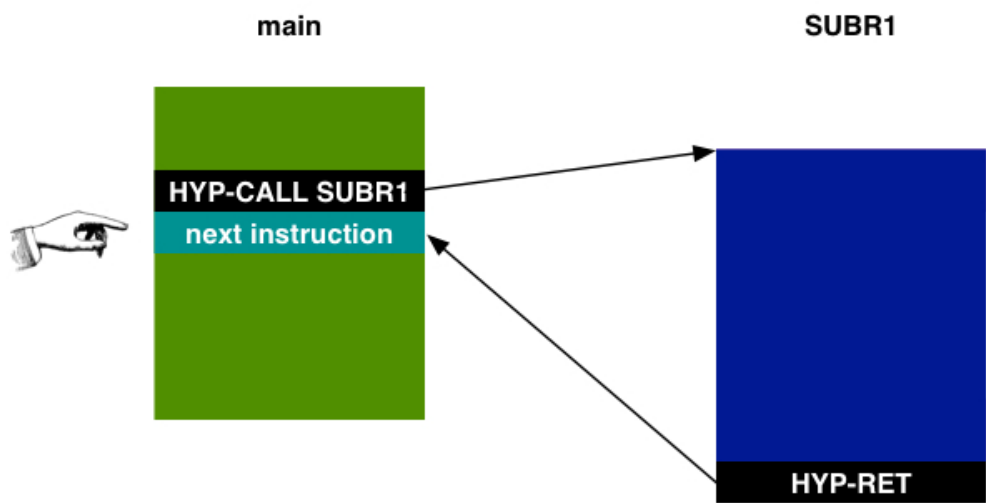
What if you want to jump to a subroutine, but then return to where you came from?

- Gosh, better need a way to record the **return address**....
- ...that is, the address of the **next** instruction after the "jump to sub"

So let's imagine a pair of hypothetical instructions:

- HYP-CALL LABEL: go to the subroutine at LABEL (but remember your return address)
- HYP-RET: go back to the stored return address!

Does this work for a subroutine?

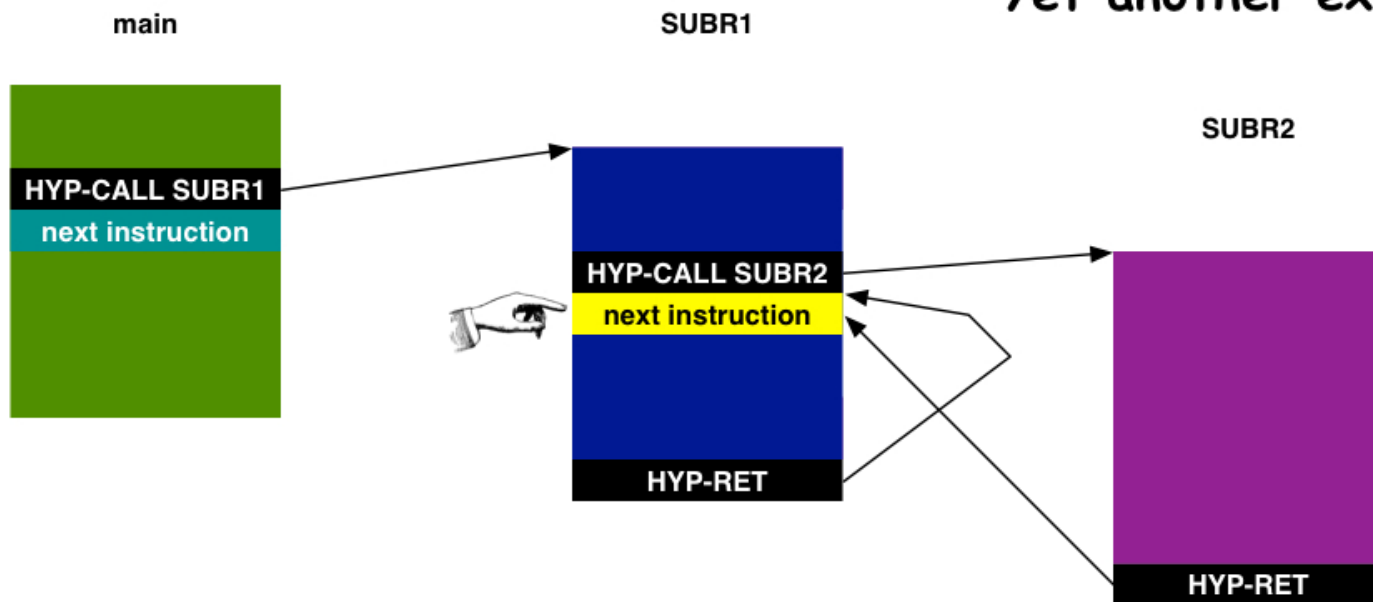


(hypothetical storage place)

return to main

What if you went two levels deep?

Yet another example



(hypothetical storage place)

return to SUBR1

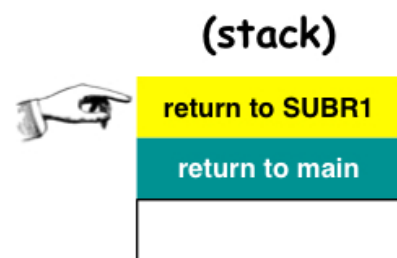
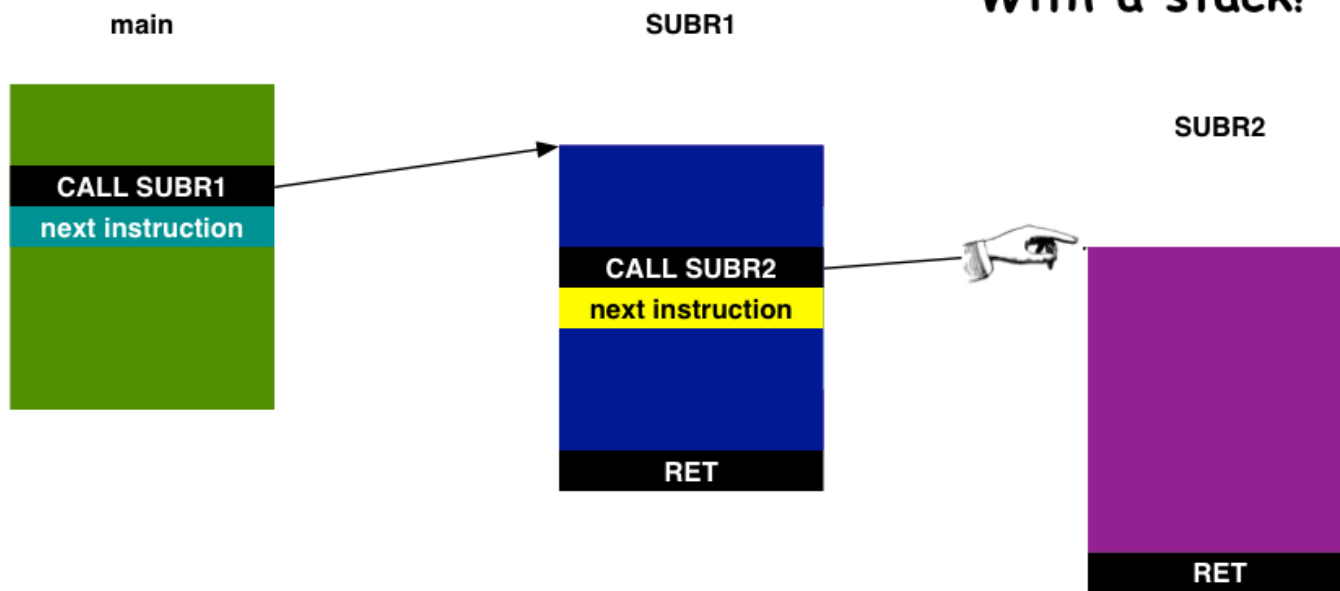
Uh oh..

- The history of how you got there, and how you should go back
- matches a STACK!

So the actual instructions:

- CALL LABEL: push the return address on the stack, and jump to LABEL
- RET: pop the return address from the stack, and go there

With a stack!

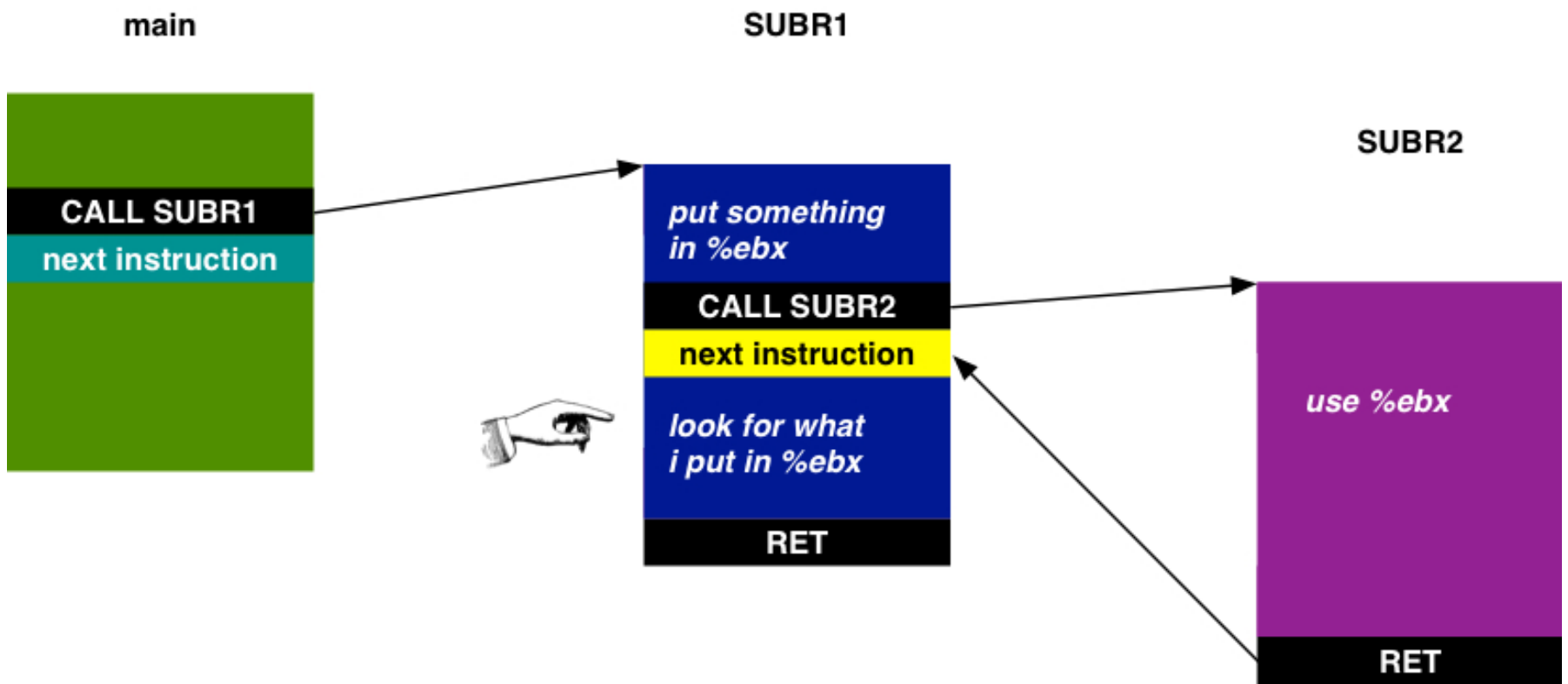


[call0.ys \(https://ssl.cs.dartmouth.edu/~sws/cs51-s15/13-stack/call0.zip\)](https://ssl.cs.dartmouth.edu/~sws/cs51-s15/13-stack/call0.zip)

2. Registers and Subroutines

Do you see any problems with the NEGATE subroutine?

```
# subroutine: %eax
```



How shall we solve it?

Two general styles:

- "callee save"
- "caller save"

[call1.ys](https://ssl.cs.dartmouth.edu/~sws/cs51-s15/13-stack/call1.zip) (<https://ssl.cs.dartmouth.edu/~sws/cs51-s15/13-stack/call1.zip>)