# Meeting 7: Register Allocation

IF ANYONE HITS ME WITH A SNOWBALL, I'LL HIT HIM WITH 250 SNOWBALLS!
WHAT IF SOMEBODY HITS YOU WITH 250 SNOWBALLS?
..SIGHHH..

## Announcements

- HW2 due today at 6pm
  - Same submission as HW1
- HW3 due 9/~~22~~ 29 -- start today! -2 week lab

## Submission

- Sign-up for interview
- Code
  - Submit (push) to GitHub
  - Submit zip on COG (as many times as you like)
  - Upload zip to Moodle
- Test
  - Submit (push) tests to https://github.com/csci4555-f17/pyyc-tests-contrib
- Survey: any non-empty answer to the following will receive full credit for this part of the lab.

## FAQ and Reminders

- Are the interview questions supposed to be "ambiguous"? No.
- Interviews are a learning activity. Work with your interviewer to get the feedback that you need.
- Need to be on-time for interviews with laptop ready. Otherwise, you will not be able to interview.
- Zoom locations are for distance students.

- Too many websites? Moodle is your "home base".
- Course assistant hours: review concepts, discuss instructor tests, discuss reference compiler

Questions

① Strategy for new lines (HW2) ✓

② Parse table output ✓

③ Liveness analysis

HW1 - HW2 - HW3

↑ PO (under HW1)

↑ parser for PO (under HW2)

current compilers variables were on the stack

works but slow!

accessing registers - 1 cycle
cache -> tens
memory -> hundreds

register allocation

↑

optimization

semantics-preserving but (hopefully) faster x86: 8 registers

9 variables

x = - -   · {x}

y = x

⋮ } no longer use x

liveness

□ a variable is <u>live</u> at a location (program location) if it may be read at a later location with no intervening assignment

---

Environment

x → -4(%ebp)   HW1

---

Source (P0)   Target (x86)

x → %eax

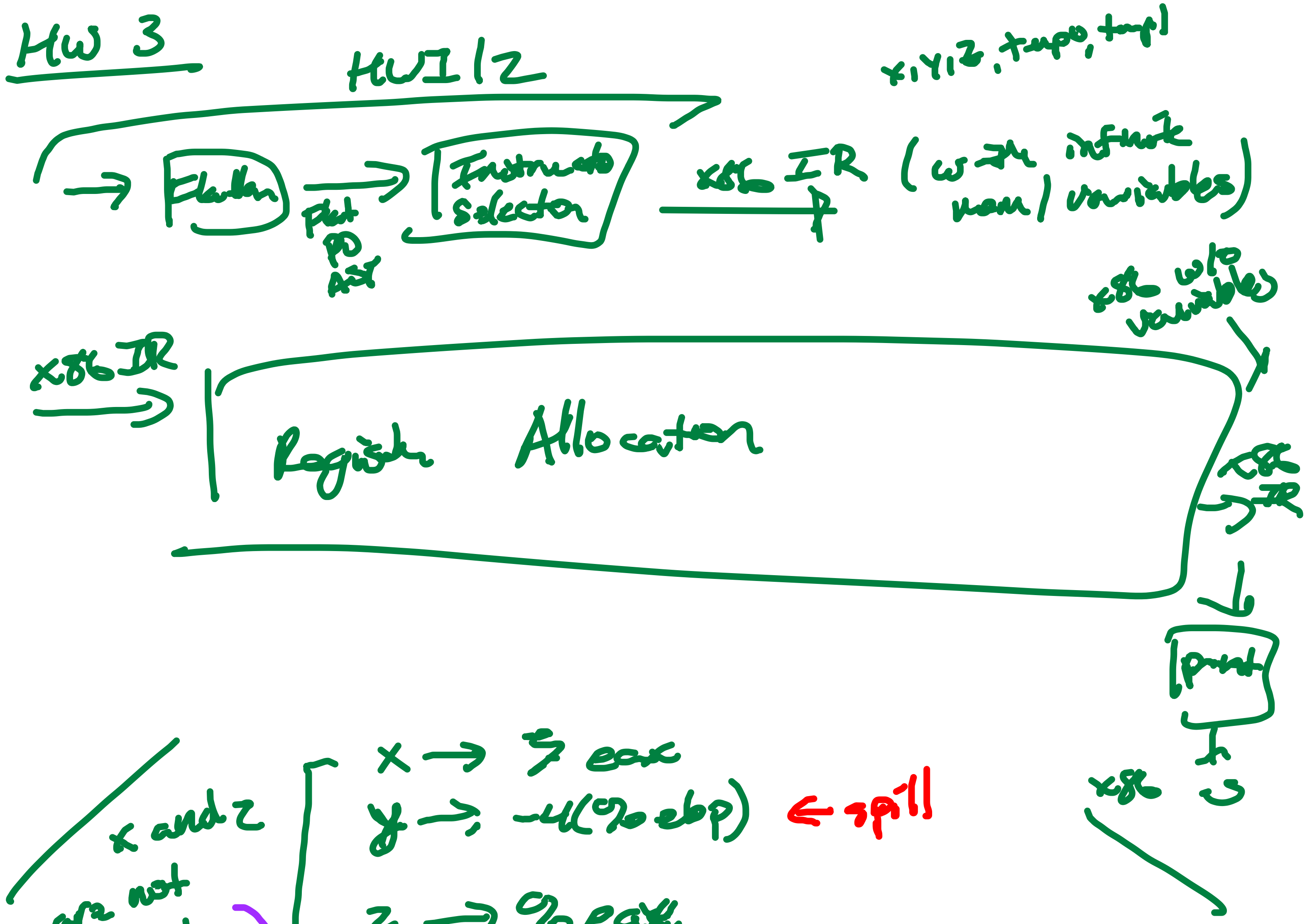
y → -4(%ebp)

← Liveness Analysis

↑

How?

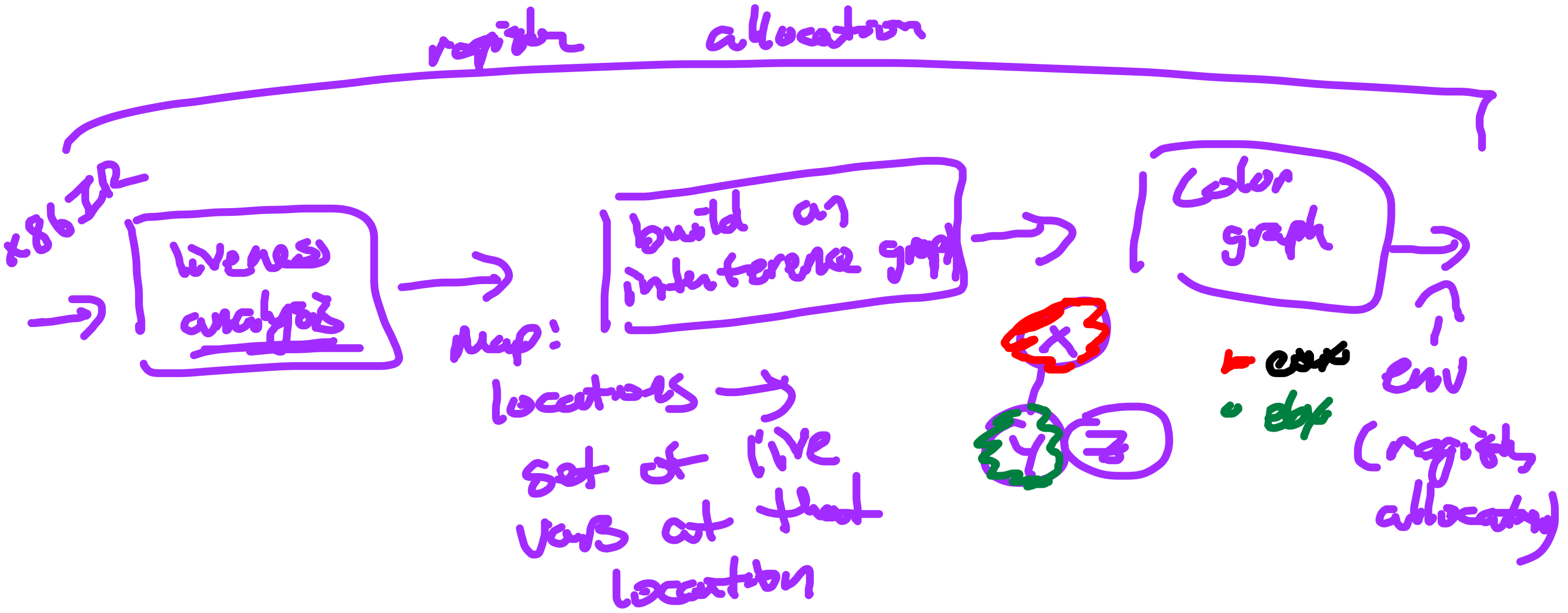
# HW 3



x → % eax
y → -4(%ebp) ← spill
z → % eax

x and z are not live at same time (or hold the same value)

x and z interfere if they are live at the same time



Map: locations → set of live vars at that location

%eax caller-save register

%edi callee-save register [?]

6 normal use registers

(2 %esp , %ebp)