Hardware, Operating Systems, and Revision Control

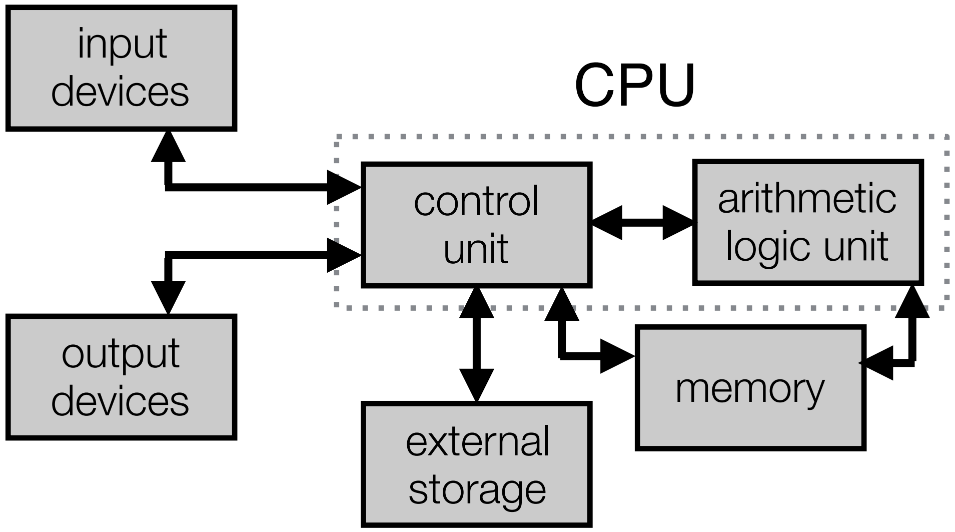
Week 2

Craig Rasmussen (Research Support Services, University of Oregon)

How do these things work?

Computer Architecture Basics

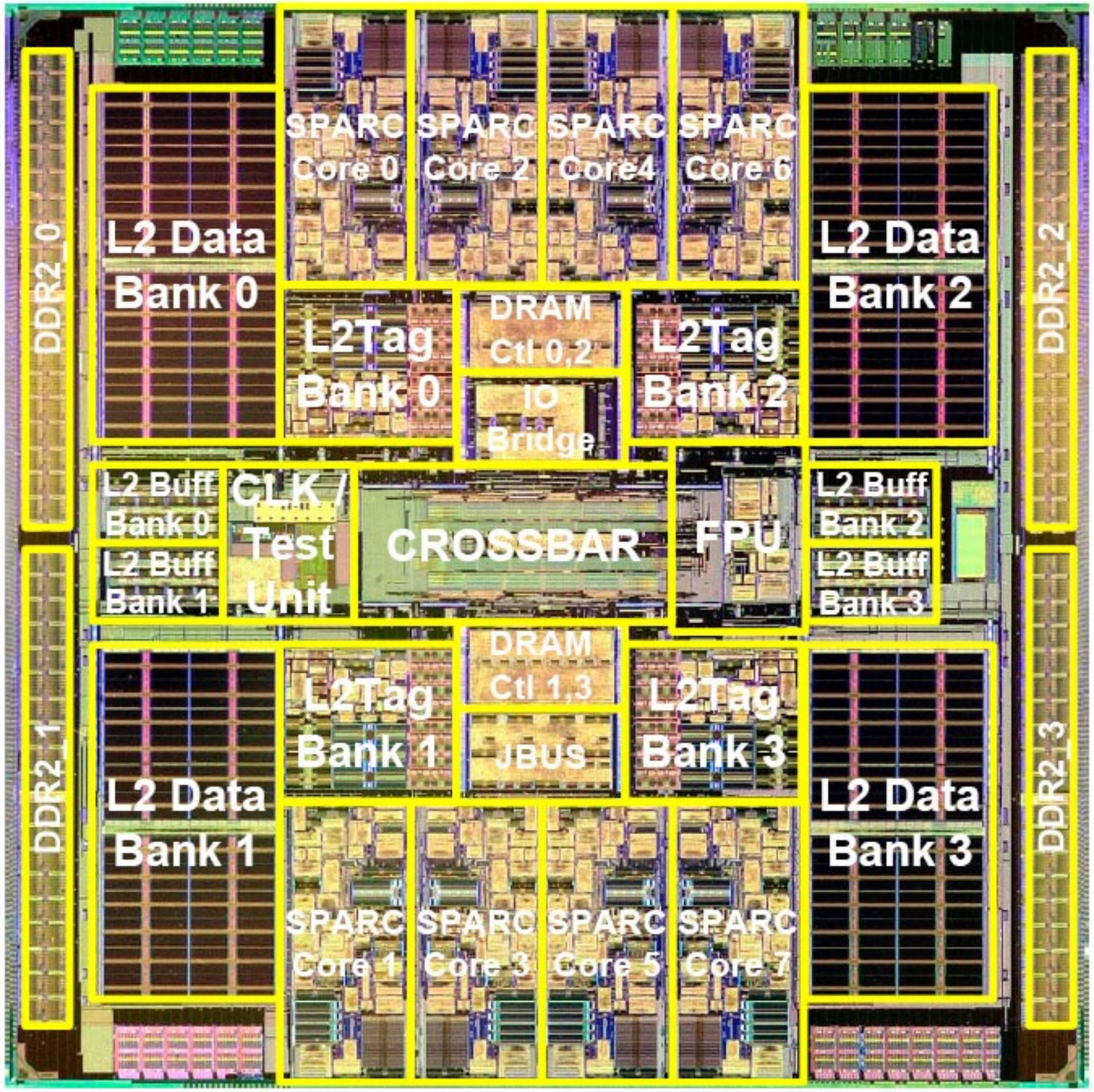
- A computation
  - input, compute, output



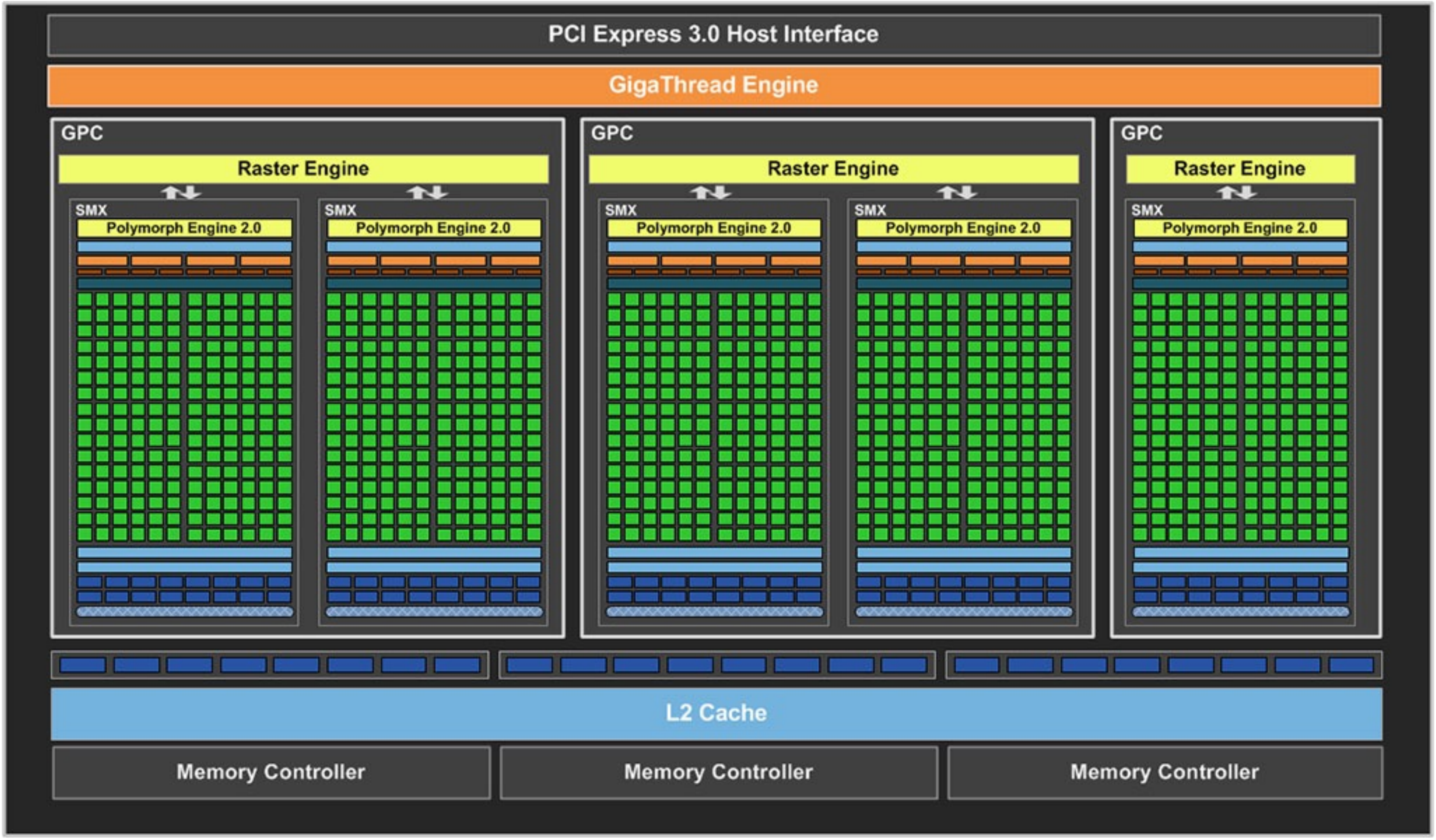
- Control unit
  - operates on instructions (load, add, store)
  - contains limited set of registers for very fast storage
- Arithmetic logic unit
  - performs calculations (register1 + register2; result stored in register3)
- A bus connects CPU to main memory and memory caches (L1, L2, …)
  - load memory into a register
  - store from a register into memory

Microprocessor Architectures:
A chip

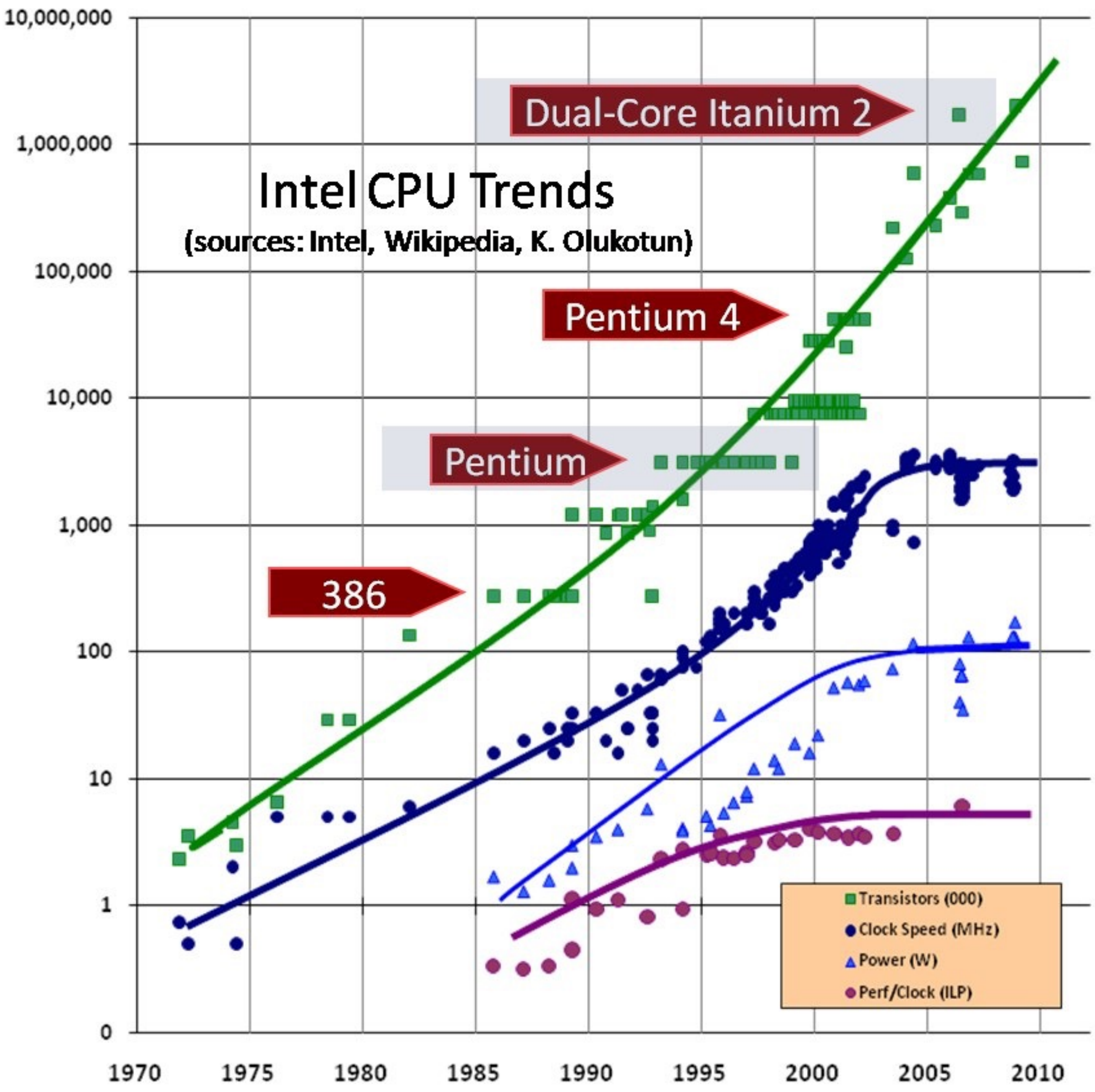
Microprocessor Architectures:
8 Core CPU Architecture (Sparc)



Microprocessor Architectures: GPU Architecture



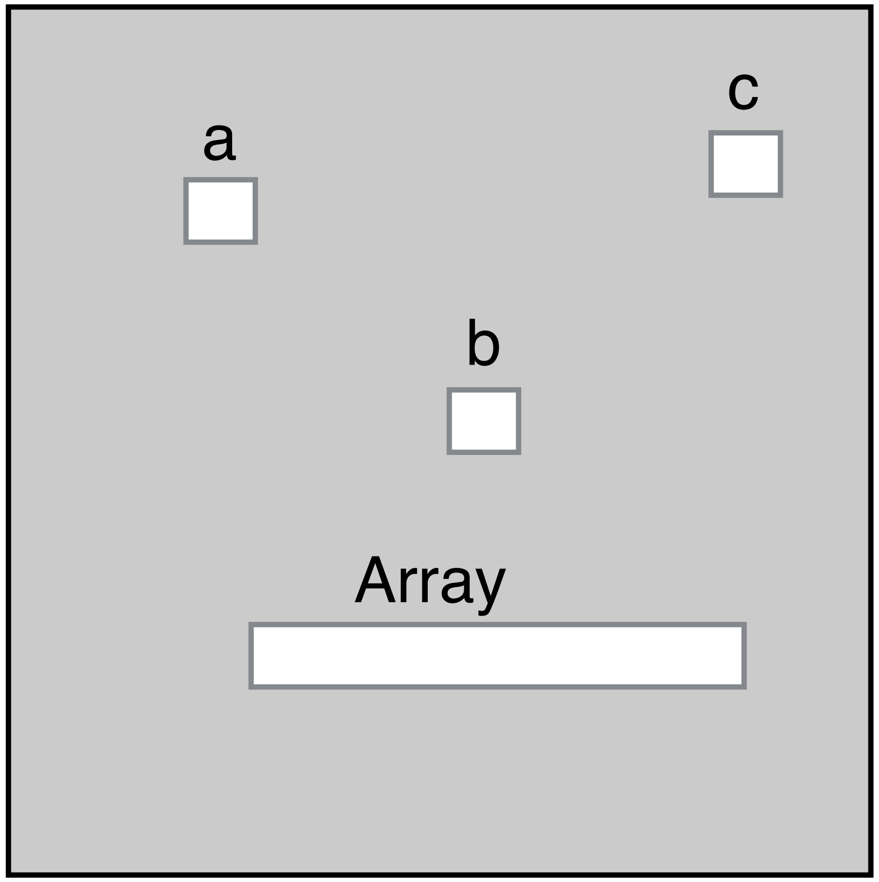
Computer Architecture Trends — energy is an issue — LANL Roadrunner required 2.35 MW



Memory Basics

- Disk storage - slow
- Main memory - fast
- Caches (I, II, III) - faster
- Registers - fastest
- Variables are stored in main memory
    - located by an *address*
    - hexadecimal (0xa32b0fd)
    - random access

c = a + b



Operating System Basics

- Unix is an operating system
  - so is Windows
  - what is Mac OS X?
- Bash is a program (also Python)
  - called a shell (ls, pwd, cd, date, echo …)
- A program is a set of instructions
- The operating system manages resources and handles events
  - keyboard input, memory, disk drives, network
- The operating system schedules execution of things *(fairly)*
  - processes and threads

What’s the most important job of an operating system?

- It’s the immune system
    - protects the system from idiot programmers
    - protects the system from evil hackers
        - viruses, malware, denial of service attacks (?)
- What is a virus anyway?
    - self replicating RNA (recall instructions)
- But can it protect the system from the CIA?
    - how about from China?
- Back to the fairness issue
    - core wars at Stanford

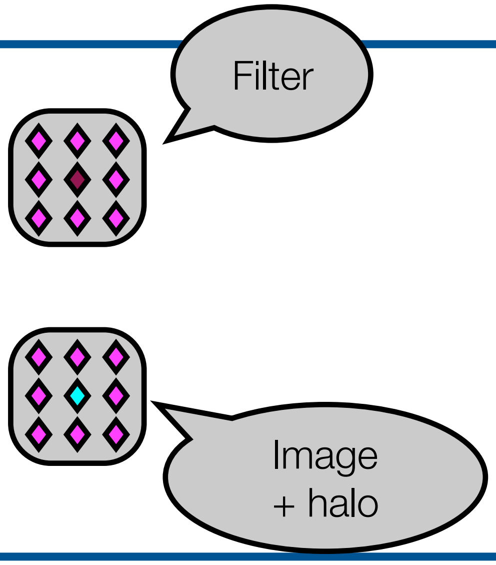
Compiler Basics

- Compilers parse input files and produce object code (instructions)

```
pure CONCURRENT subroutine convolve(Image, Filter)
   real, intent(in out), HALO(:,:) :: Image
   real, intent(in) :: Filter(-3:3,-3:3)

   Image(0,0) = sum(Filter * Image)

end subroutine convolve
```

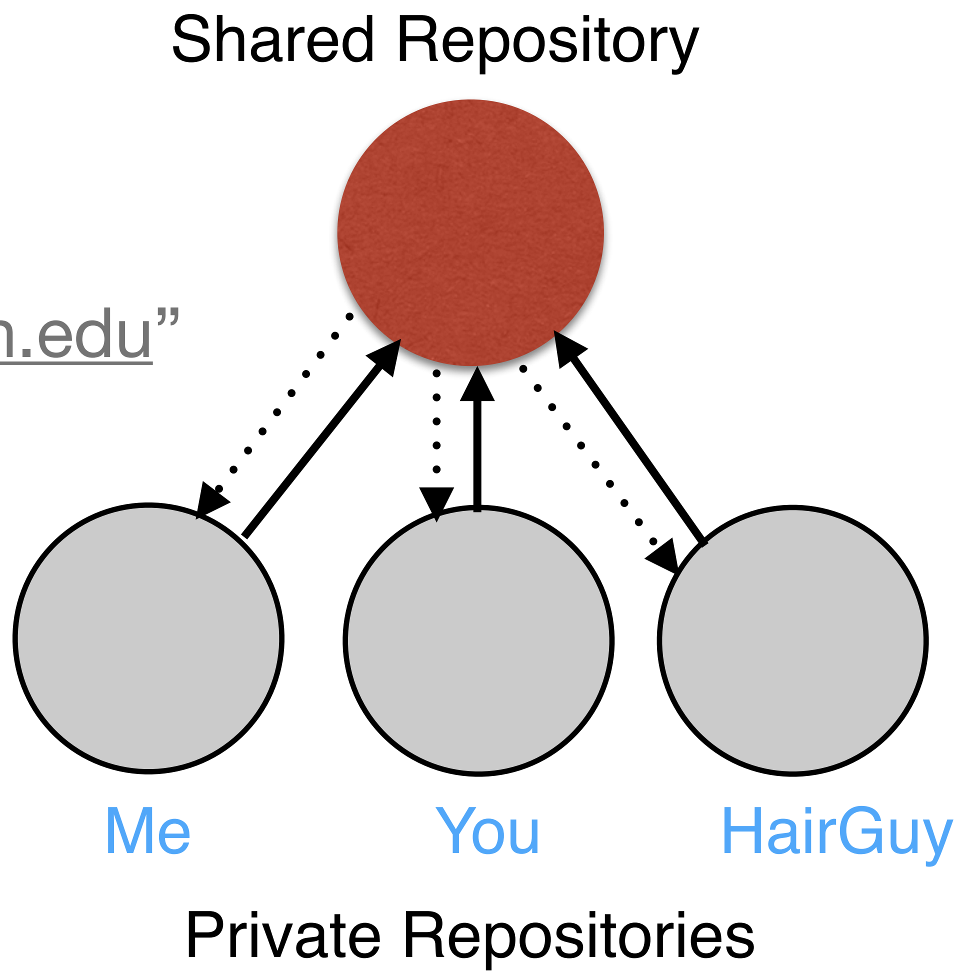


Software Engineering Basics — Revision Control and Other Tools

- What do you do when you make changes to your program code files?
  - use names for version control (file1, file2, file3, …)
  - what if working as a team
    - use google docs?
    - what if your partner screws up the code somehow (it happens!)
- Use revision control to save changes
  - allows you can back out changes
- Everyone should get an account on github.com
  - keep your class and team projects there?
- Debuggers, memory leaks, performance tools

Git is now the standard version control system

- Configuring Git
  - git config —global user.name “Your Name”
  - git config —global user.email “user@uoregon.edu”
  - git config —global core.editor emacs
- Creating a new shared repository
  - mkdir repo
  - cd repo
  - git init —bare



Shared Repository

Me

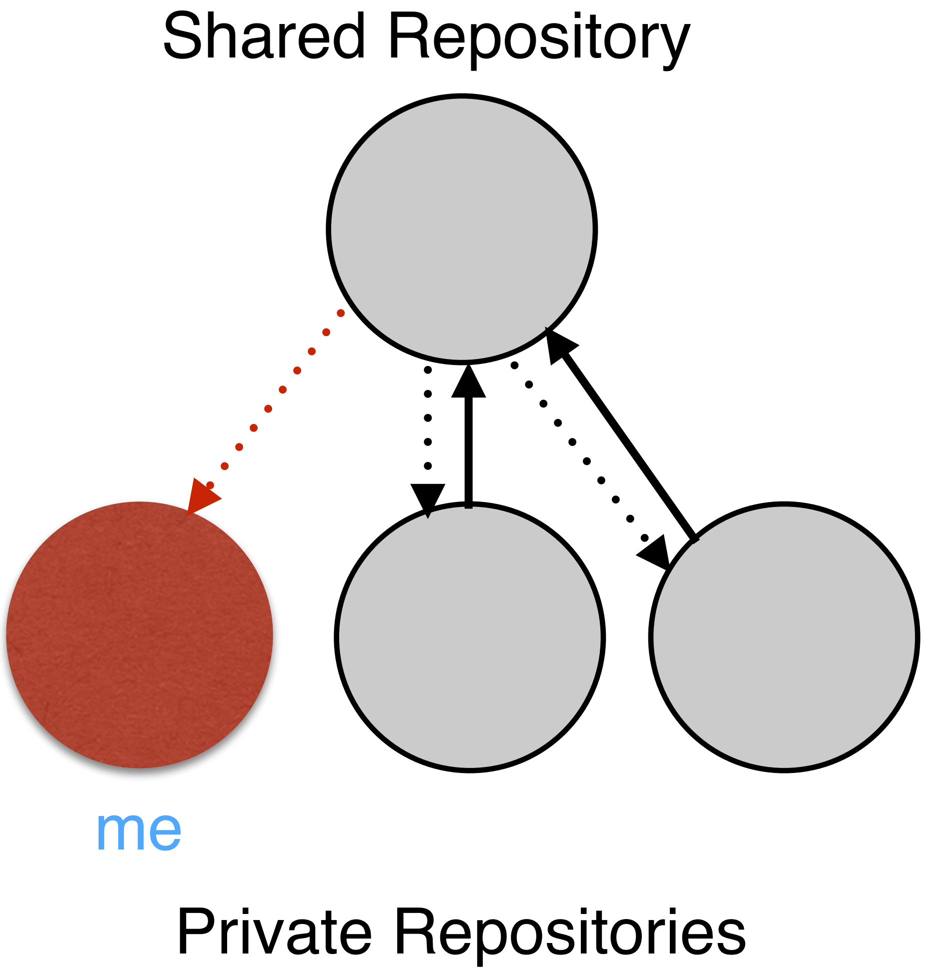
You

HairGuy

Private Repositories

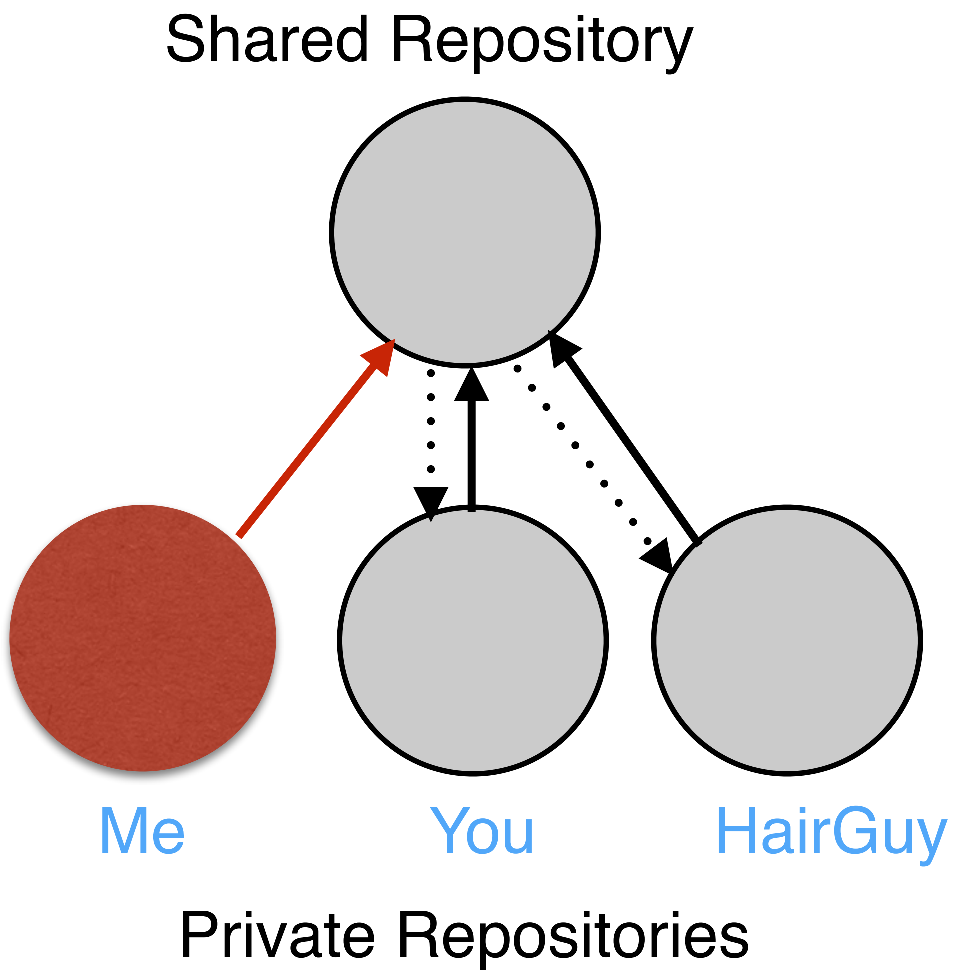
Creating a local copy of an existing repository and adding files

- Clone a repository
  - git clone /usr/local/repos/repo me
  - git clone URL
- Add a file
  - cd me
  - touch README
  - git add README
  - git commit -m”Initial version.” README



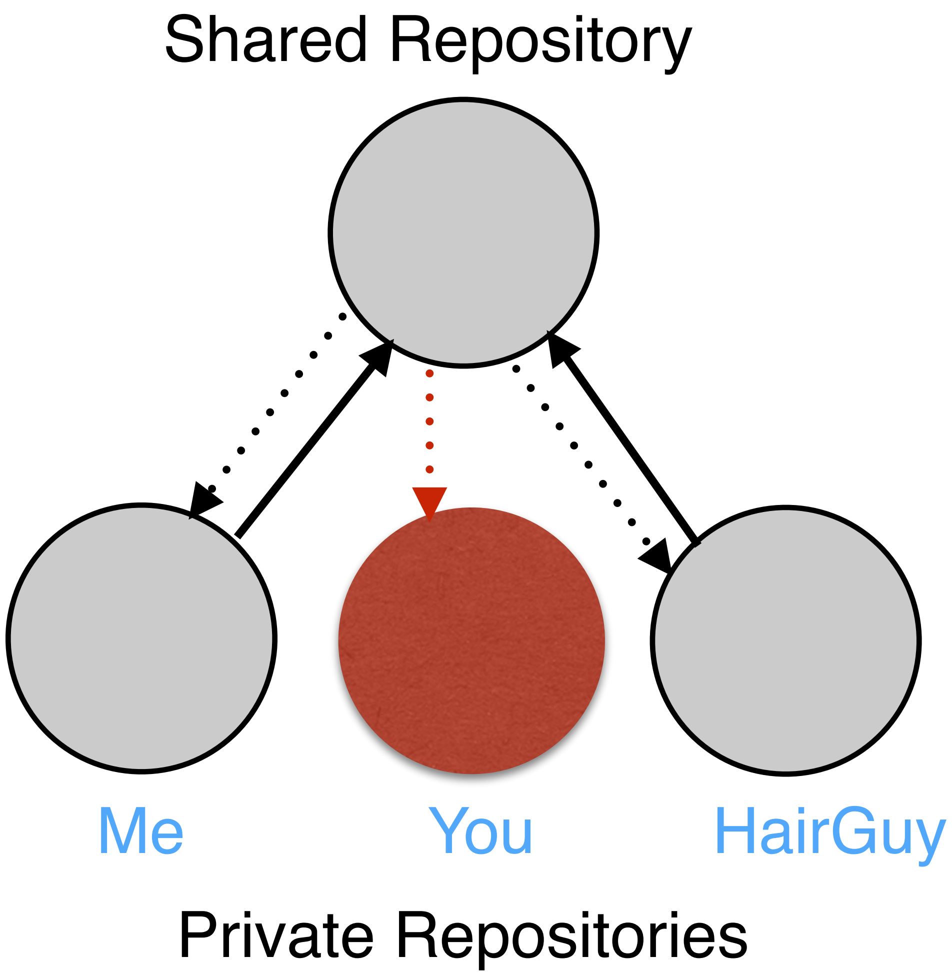
Sharing changes to a file

- Edit the file then compare changes
    - emacs README
    - git diff README
- Discovery
    - git status
- Commit changes to local repository
    - git commit -m”a message” README
- Push changes to shared repository
    - git push



Retrieving changes that a teammate has made

- Fetch the latest from the shared repository
  - git fetch
- Merge with local private repository
  - git merge origin/master
- If you have merge conflicts you must fix them
  - emacs README
  - git add README
  - git commit -m”Liked my changes better”


Shared Repository
Me
You
HairGuy
Private Repositories