15-213 Recitation 14

Introduction to Computer Systems

Fahim Dalvi

4 December, 2013

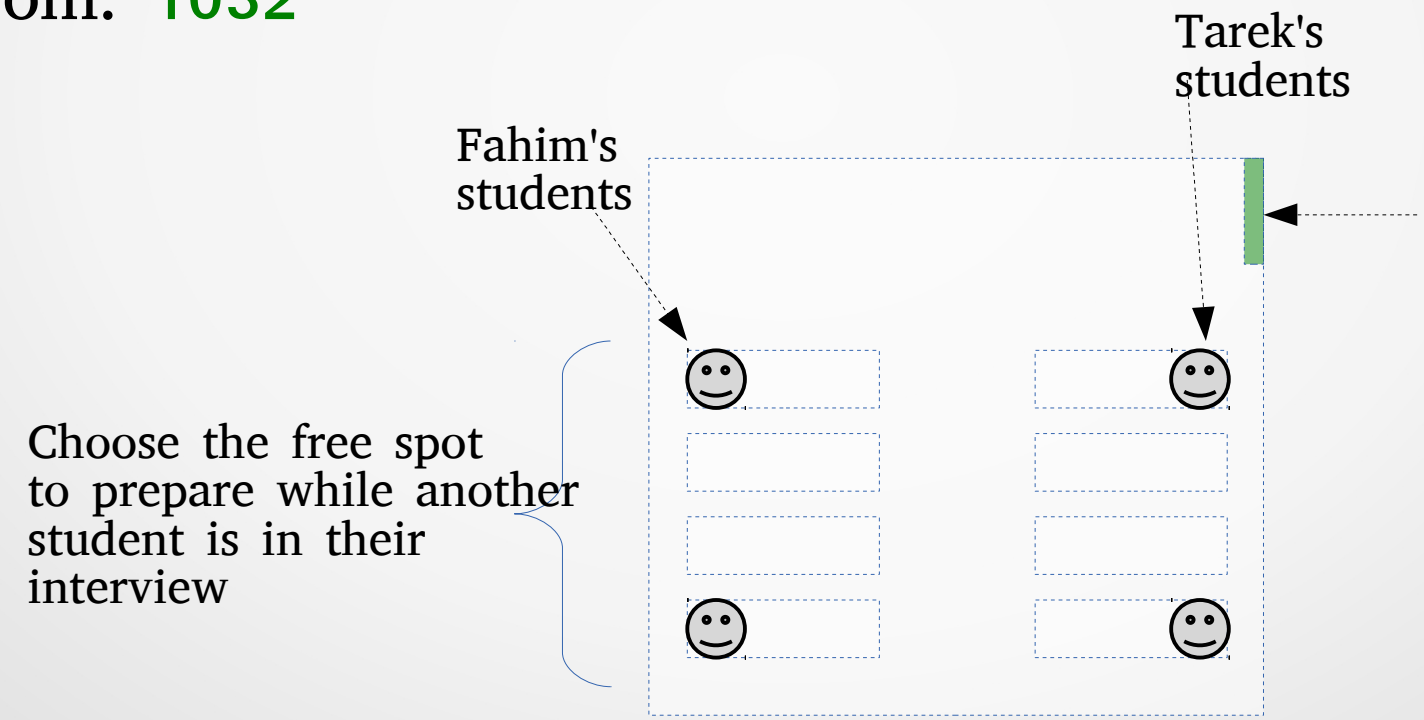
Proxy Lab

Due tonight!

No late days for the assignment!

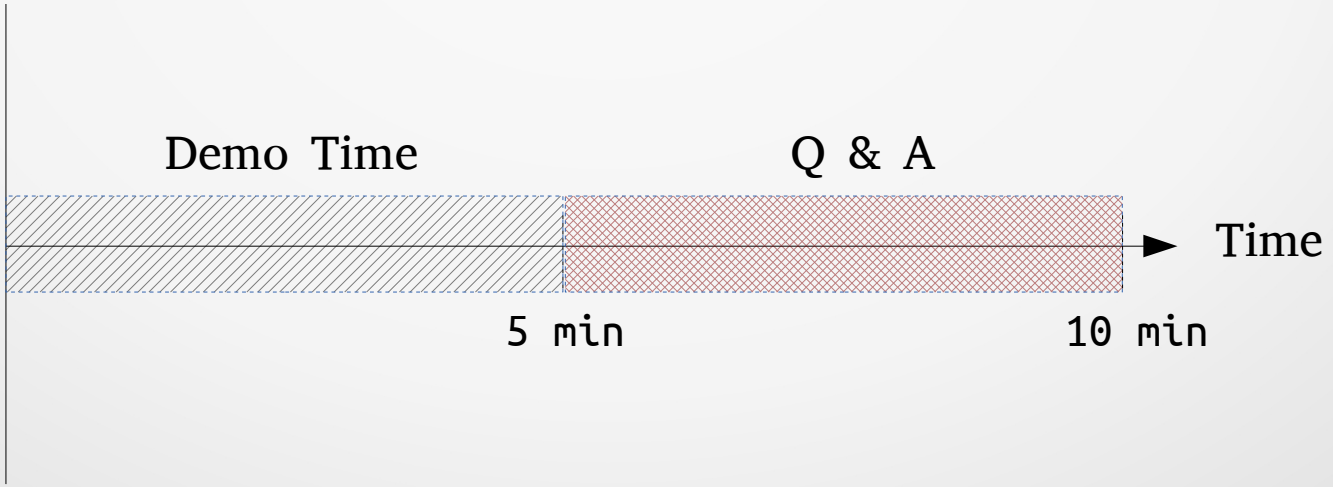
Proxy lab - Interviews

- 10th of December, 2013
- Reading day
- Room: 1032



Proxy lab - Interviews

- Schedule is up: http://www.qatar.cmu.edu/~kharras/courses/15213-f13/recitations/proxy-schedule.html
- Please make sure you arrive before time!
- Any lost time will be deducted from your interview time



Proxy lab - Demo

- The rubric is roughly as follows:
  - Basic proxy operations
    - Basically demo to us that your proxy works
    - Go to a few websites
      - www.qatar.cmu.edu/~kharras/ ← Good test for flash
      - www.qatar.cmu.edu ← Good test for fancy webpages
      - www.cs.cmu.edu ← Good test for simple pages
      - www.cnn.com ← Good test for webpages with external content, like ads
      - www.espncricinfo.com ← Good test for video streaming websites

Proxy lab - Demo

- The rubric is roughly as follows:
  - Handling concurrent requests
    - Show us how you actually handle concurrent requests
    - Load up a page with several images, the images should all load in parallel pretty quickly!
  - Caching
    - Explain to us what datastructures/mechanisms you used for caching
    - Show us how you handled locking/unlocking parts of the cache
    - Show us how you implemented the LRU policy!

Proxy lab - Demo

- General guidelines
  - Use the demo time to convince us that the proxy works well
  - Don't spend time on useless details → Don't try to explain each and every line to us, you will not have enough time
  - Our questions will be straightforward if you have understood how you have implemented your proxy :)

Proxy lab - Demo

- Avoiding pitfalls:
  - Make sure that 'caching' is disabled in your browser. In Firefox, this can be found under `Preferences > Advanced`
  - Setup the proxy on the shark machine and your browser before your interview begins!

Proxy lab - Style

- Same general stuff
  - Name/AndrewID
  - No memory leaks
  - Comments
  - No dead code
  - No magic numbers
  - Check for return values from system calls such as opening, binding, listening to sockets etc...

Proxy lab – Cache testing

- Some suggestions to test your cache:
  - First way:
    - You can host webpages/images on your unix account under `www`
    - Delete the file and try reloading the page, your proxy should be able to serve it
  - Second way:
    - `python -m SimpleHTTPServer 12345`
    - Instantly serves the file in the current directory
    - Much faster than waiting for the andrew publishing stuff to happen!

Any Questions?

Final Exam Review

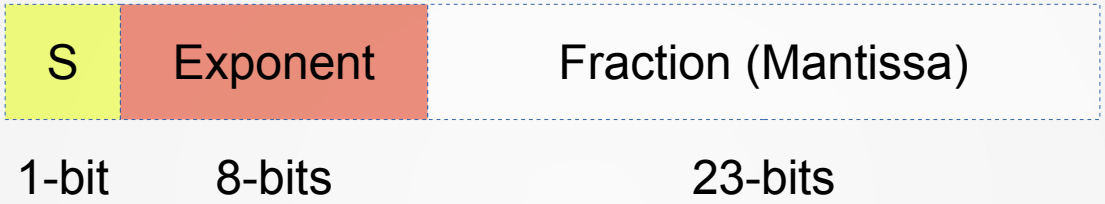
- Final exam is on the 12th of December, 2013
- 9:00 AM in 1199
- Last day of finals, so gives you a good amount of time to study!
- Covers everything from the beginning to the end

Final Exam Review

- Let us see a real exam:
  - http://www.cs.cmu.edu/~213/oldexams/final-f12.pdf

Floating Point refresher

- Basic Format



$$-1^{S} \times M \times 2^{E}$$

Where E is based on the Bias

$$Bias = 2^{(k-1)} - 1 = 2^{(8-1)} - 1 = 127$$

k = exponent bits

Interpreting the Bits

- Exponent
  - 0000..000 → Denormalized Form
    - E = -Bias + 1
    - Frac = 0.FFFFF…
  - eeee..eee → Normalized Form
    - E = Exponent – Bias
    - Frac = 1.FFFFF….
  - 1111..111 → Special
    - If Frac = 0000.000 → Infinity
    - Else → NaN

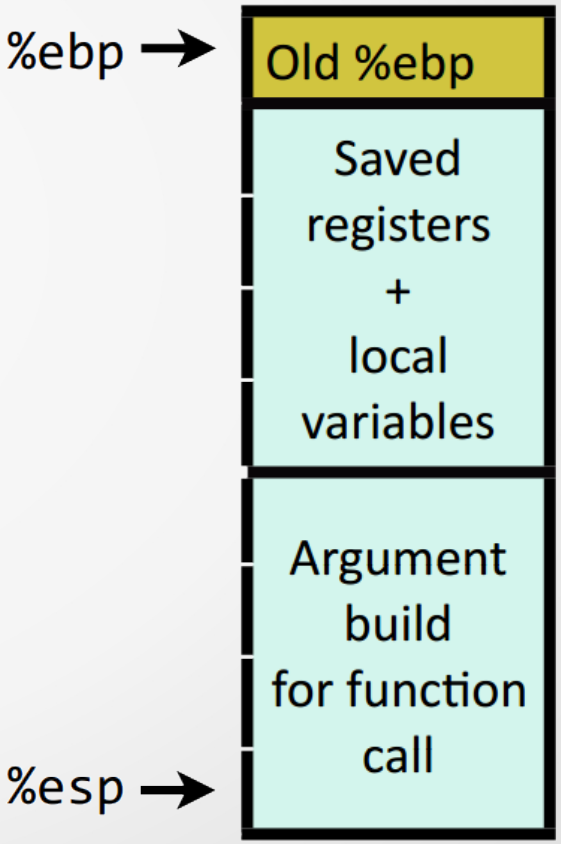
Arithmetic Operations

- Two operand commands → Always src,dest
  - Format / Result

Format	Result
`addl src,dest`	`dst+=src`
`subl src,dest`	`dst-=src`
`imull src,dest`	`dst*=src`
`sall src,dest`	`dst<<=src`
`sarl src,dest`	`dst>>=src`
`xorl src,dest`	`dst^=src`
`andl src,dest`	`dst&=src`
`orl src,dest`	`dst|=src`

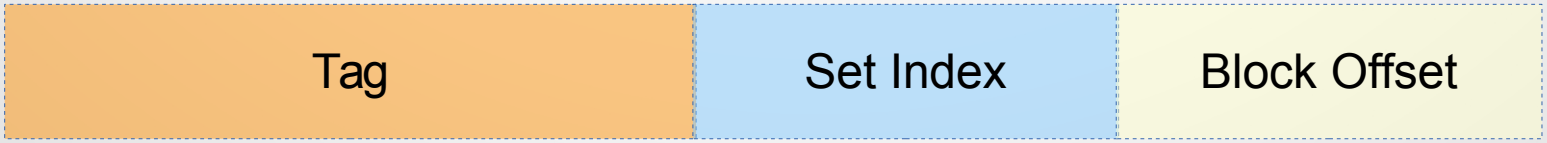
Stack Frames

- Every function call is given a stack frame
- What does a C function need?
  - Local Variables
  - Space to save callee saved registers
  - Space to put computations
  - A way to give arguments and call other functions

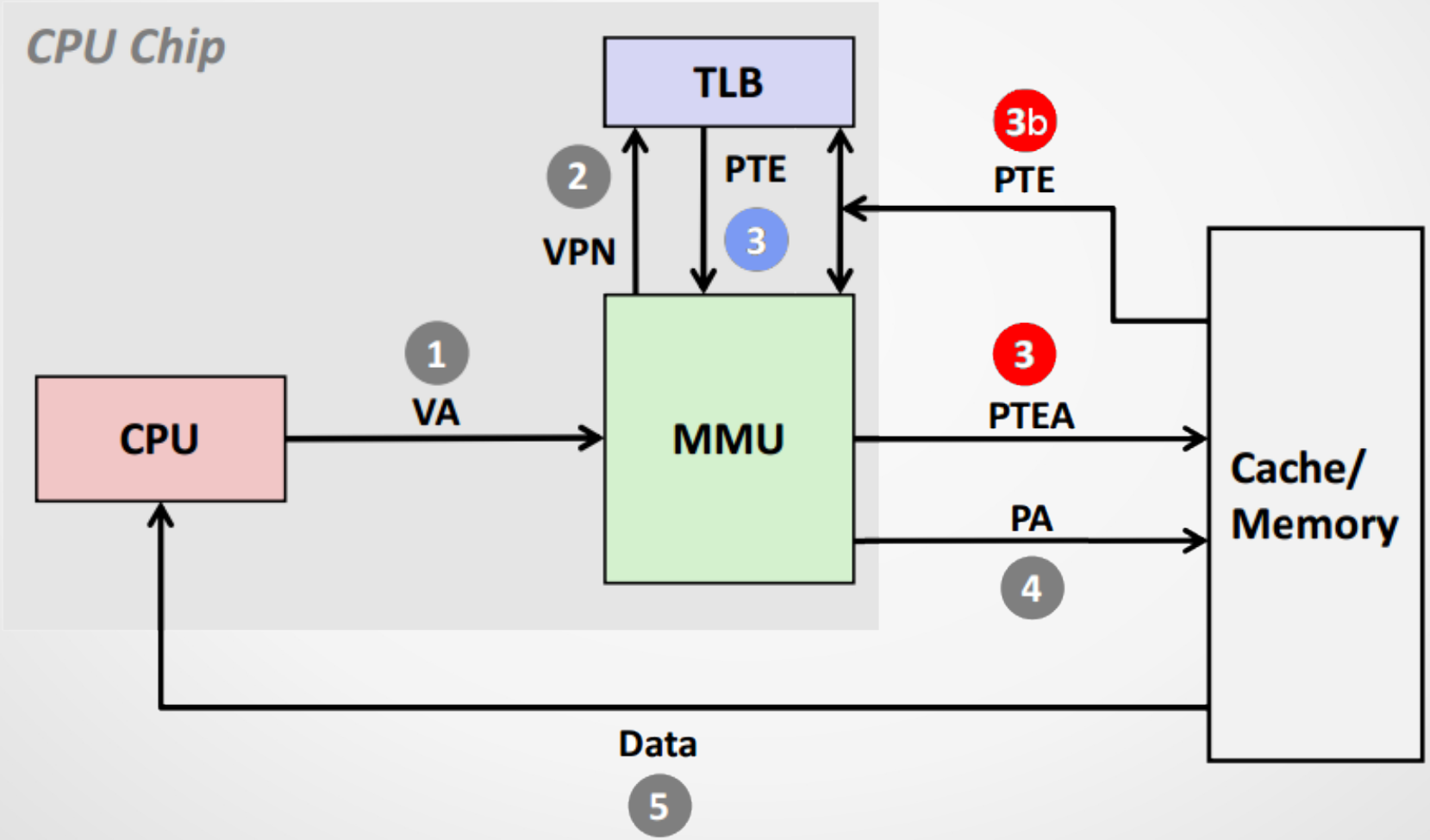


Caches

- A *cache* is a set of 2^s cache sets
- A *cache set* is a set of E cache lines
  - E = 1 → Direct-mapped
  - E-way associative
- Each cache line stores a block
  - Each block has 2^b bytes



An Overview


CPU Chip
TLB
2
VPN
PTE
3
3b
PTE
1
VA
CPU
MMU
3
PTEA
PA
4
Cache/
Memory
Data
5

Any More Questions?