CS 537
Section 6
Project 2: Virtual Memory

Michael Swift

Quiz # 3

Group size

- What size group should we do this project in
  - 2
  - 3
  - 4
- How should we pick groups?
  - Pick your self
  - I pick

Group grading

- The project grade is:
  - 50% functionality
  - 30% code and write-up
  - 20% group grade
- The group grade:
  - We distribute the rest of the grade among the group
  - Example:
    - if you get 100% grades for functionality and code/write-up, you can distribute a 100% grade for the group (75 points total)
    - If you get 66% grade on the results, you distribute 50 points through your group
- How should we assign this?

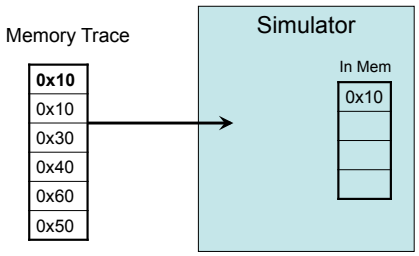
Dividing the Work

- On this assignment, it is up to you
- Some recommendations:
  - Roles
    - Architect: come up with a design
    - Developer: write the code
    - Tester: develop tests to show the code works
    - Writer: write up final document
  - Modules
    - Part 1, part 2, part3 of the assignment
  - You don't all have to do all the jobs

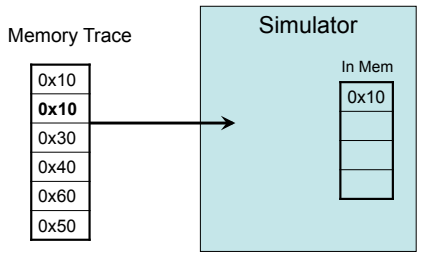
Project 2: Simulating VM

- Your task (should you choose to accept it):
  - You are working on a team to design a new mobile device (mp3/phone/portable computer)
  - You need to make a recommendation as to how to organize the memory system
    - Page replacement algorithm for the OS
    - Amount of memory for the programs
    - Page size for the hardware
  - You are given: a list of page references from common programs

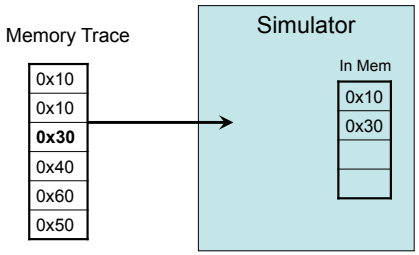
Trace Based Simulation



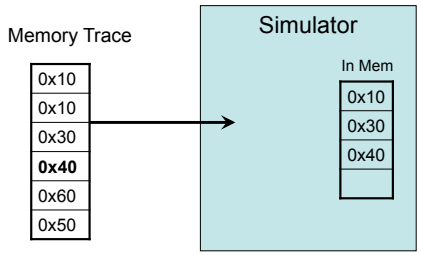
Trace Base Simulation



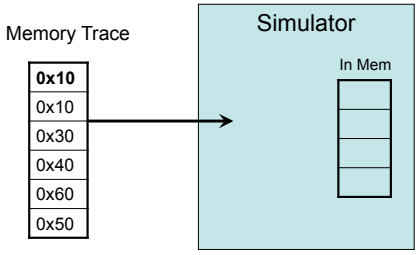
Trace Base Simulation



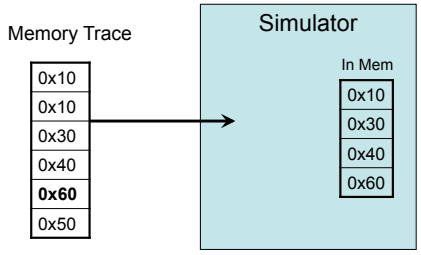
Trace Base Simulation



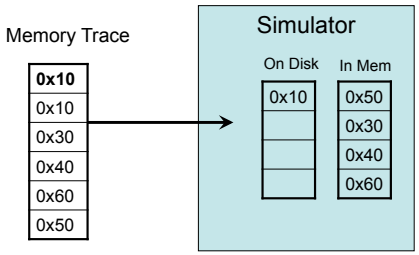
Trace Base Simulation



Trace Base Simulation



Trace Base Simulation



Memory Simulation

- On each memory reference
  - Check if page is in memory.
    - If so, continue
    - If not, page fault
      - Find page to evict
- You need to maintain
  - Page table for each process, saying where in memory its pages are (on disk or in memory)
- You do not need to maintain
  - A precise physical - virtual mapping
  - Just knowing that a page is in memory or on disk is enough
- Note: the traces are big (5MB each) -- don't copy them if you don't need to.

What else to do

- Implement 4 page replacement strategies:
  - fifo, 2nd chance fifo, LRU, and clock
- Experiment with 3 page size
  - 4 kb, 8 kb, 16 kb
- Experiment with 3 memory sizes
  - 50 pages, 500 pages, 5000 pages
- If you have time: implement your own replacement strategy
  - Worth up to 15%
  - but only if you are the fastest in the class

How to do this project

- Come up with a design
  - Break the problem into modules
  - Figure out what each module does
  - Figure out the interface
    - function calls in to the module
    - function calls out of the module
- Write and test modules separately
  - Page table
  - Page replacement
- Integrate them all

Schedule

- This week: come up with a plan
- Next week: implement page tables without replacement (infinitely sized)
- Following week: implement replacement strategies