


Usual stuff

- n Project 2 back today
 - n Average: 66.8/80
- n Today:
 - n Project 3
 - n A few project 2 comments


1



Project 3

- n Out now and due next Friday
- n Given: vmtrace
 - n Simulates virtual memory on a memory trace file
 - n Tracefile = a list of all VA references during execution
- n Takes in:
 - n a memory trace file (given, netscape.exe.et.gz)
 - n physical memory size
 - n page size
- n Outputs:
 - n # of memory references
 - n # of page faults
 - n compulsory faults
 - n page evictions
 - n pageouts


2



P3 Goals

- n Implement some page replacement algorithms
- n Design and perform an experiment on some aspect of virtual memory
 - n Important to pick a good topic, ask us if not sure!


3



Replacement Algorithms

- n Given:
 - n random
- n You need to write:
 - n FIFO
 - n LRU Clock
 - n One of your choice
 - n A few possibilities:
 - n True LRU (e.g. via storing full timestamp)
 - n Variations on LRU Clock (enhanced second-chance, etc)
 - n LFU/MFU
 - n Your own!
- n You can write more than 3 if your experiment focuses on replacement algorithms.


4



Project 3 experiment

- n Have a hypothesis
 - n "Algorithm y is better than algorithm x"
 - n "Big pages are better"
 - n "Prefetching will reduce the number of page faults"
 - n "If we understand why x happens, we can fix it"
- n Explain why you think it will turn out that way
- n Two steps
 - n Determine baseline behavior
 - n New test
 - n Change one aspect of the system, observe differences


5



Good experiment ideas

- n What is the ideal page size for this trace under different amounts of main memory?
- n Compare performance of various replacement algorithms. How much better/worse is page replacement algorithm X than Y?
 - n Compare "real" LRU and LRU clock, FIFO, 2Q, ARC, etc
- n How close can we come to LRU without doing any work between page faults?
 - n No scanning, constant work per page fault
- n How important is recency vs. frequency in predicting page re-use?


6



Not so good ideas

- n What kind of music is made when I convert the address trace to notes?
- n Can I make a fractal out of this data?

7



Tips

- n vmtrace is not an execution simulation
- n You control what happens on a page fault
- n You control what happens on a memory access
- n You can modify formats for PTE, page, etc
- n Refresh your scripting skills
- n vmtrace is very CPU-intensive
 - n spinlock/coredump: PIII-800/256MB
 - n Find faster machines (such as Linux boxes in the lab)
 - n Copy the trace file to local machine

8