Project 1

$\boxed{\text{Feedback}}$

Statistics

Report sizes: 18, 21, 12, 10 (excluding program and data).

Architectures: Pentium III, SGI O_2, Sparc Station 4.

Can parameters be identified: Yes, very nicely. Correspond with specifications.

Values:

	Size	*Blocksize*	*Access time*
L1	8–32 Kb	16–32 b	–
L2	0–1 Mb	32–64 b	$\times$ 2–9
RAM	64–1024 Mb	4 Kb	$\times$ 2–8
Disk	–	–	$\times 10^4$

Observations

- Sequential access just as good as (or better than) random access for finding memory sizes.
- Not all RAM available to processors.
- Hard to run programs out of RAM when size of it is 1 Gb. Many crashes.
- When finding the virtual page size, the plot descends immediately after peak, often oscillating. Solutions: try array lengths which are prime numbers, or try `mmap`.
- Random permutation is space consuming to produce (factor of two). Solution: scale up small permutation, use only every k'th element of large array.

Comments

Good:

- Beautiful reports: LaTeX'ed, nice plots.
- Good discussions, everybody did some actual thinking.
- Everybody use the method of pointer based jumping in array.
- Several machines.
- Actual specifications looked up.

Comments

Problems:

- State all relevant details of experiments (algorithm, machine, method of timing and averaging, any parameters not obvious from axes of graph).
- Units on axes of graphs. Captions with info on experiment.
- Run experiments for ≥ 1 second to minimize fluctuations.
- Don't always use powers of two.
- Make program publicly available (but no need to print it out).
- Just allocating random pointers results in short cycles.
- Try to remove overhead of loop when calculating access times. Try to run program twice, with the loop unrolled 1 time vs. 10 times, and solve the two linear equations.