EE 4720 **Homework 6** **Due: Not Collected**

To answer the questions below you need to use the PSE dataset viewer program. PSE (pronounced see) runs on Solaris and Linux; you can use the computer accounts distributed in class to run it, a Linux distribution may also be provided for running it on other systems.

Procedures for setting up the class account and using PSE are at
`http://www.ece.lsu.edu/ee4720/proc.html`; preliminary documentation for PSE is at
`http://www.ece.lsu.edu/ee4720/pse.pdf`.

Problem 1: The code in `http://www.ece.lsu.edu/ee4720/2002f/hw6.pdf` includes two routines to perform a linear search, `lookup_array` and `lookup_ll`. Routine `lookup_array(aws,foo)` searches `aws` for element `foo`. The list itself is an ordinary C array, structure `aws` (array with size) includes the array and its size. Routine `lookup_ll(head,foo)` searches for `foo` in the linked list starting at `head`.

The code calls the search routines under realistic conditions: Before the linked list is allocated dynamic storage is fragmented and before the searches are performed the level-1 cache is flushed. See the code for more details.

The code was executed on a simulated 4-way superscalar dynamically scheduled machine with a 64-entry reorder buffer and a two-level cache. The simulation was recorded in `hw6.ds`; view this dataset file using PSE to answer the questions below.

The code initializes the lists with identical data and then calls the search routines looking for the same value. Answer the following questions about the execution of the two lookup routines. When browsing the dataset be aware that the time spent in the lookup routines is dwarfed by the time needed for setting everything up and so only the last few segments need to be examined.

(*a*) Would increasing the ROB size improve the performance of the linked list routine, `lookup_ll`? Explain.

(*b*) Would increasing the ROB size improve the performance of the array routine, `lookup_array`? Explain.

(*c*) As can be seen viewing the PED plots, the array routine follows a regular pattern while the linked list code starts off slowly but as it nears completion it runs much faster. Why does the linked list code speed up like that?

(*d*) How could one determine the line size from the PED plots? Be specific and use numbers from the dataset. (The line size can be found two other ways, if you come upon them by all means use them to check your answer that is based on the PED plot.)

(*e*) Before people stopped replacing $2,500 computers every six months computer engineers would loose sleep worrying about The Memory Wall, the growing gap in performance between processors and memory (*e.g.*, the number of instructions that could have been executed while waiting for memory). What is it about the array routine that lets it sail over the memory wall while the linked list routine is stopped dead? The answer should take into account certain load instructions and the critical path. Discuss how the performance of the routines change as the L1 miss time gets longer and longer.