It would be interesting to analyze the performance of priority queues under this model.

The Modified LRU replacement policy, suggested in Section 5.1, was found to work well only in some settings. An alternative page replacement policy could be one which always keeps in main memory the page that contains the highest level nodes (as opposed to the page containing the *root*). Node alignment in K-way trees was found to be useful and deserves further treatment.

REFERENCES

1. A. Aggrawal, A. K. Chandra and M. Snir, Hierarchical memory with block transfer, *Proc. of the 28th Annual IEEE Symposium on Foundations of Computer Science, Los-Angeles*, 204–216 (1987).
2. A. V. Aho, J. E. Hopcroft and J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley (1974).
3. A. V. Aho, J. E. Hopcroft and J. D. Ullman, *Data Structures and Algorithms*, Addison-Wesley, Reading, Mass. (1983).
4. G. Bell, The future of high performance computers in science and engineering. *Communications of the ACM* **32**, (9) 1091–1101 (1989).
5. P. Bratley, B. L. Fox and L. E. Schrage, *A Guide to Simulation*, Springer-Verlag, New York, N.Y. (1983).
6. L. Devroye, *Non Uniform Random Variate Generation*, Springer-Verlag (1986).
7. R. Engelbrecht-Wiggans and W. L. Maxwell, Analysis of Time Index List Procedure for Synchronization of Discrete Event Simulations. *Management Science* **24** (13), 1417–1427 (1978).
8. D. B. Johnson, Priority queues with update and finding minimum spanning trees. *Information Processing Letters* **4** (3), 53–57 (1975).
9. D. W. Jones, An Empirical Comparison of Priority-Queue and Event-Set Implementations. *Communications of the ACM* **29** (4), 300–311 (1986).
10. J. H. Kingston, Analysis of tree algorithms for the simulation event list. *Acta Informatica* **22**, 15–33 (1985).
11. J. H. Kingston, Analysis of Henriksen's algorithm for the simulation event set. *SIAM Journal of Computing* **15** (3), 887–902 (1986).
12. D. E. Knuth, *The Art of Computer Programming, Sorting and Searching (Vol. 3)*, Addison-Wesley, Reading, Mass. (1973).
13. G. Lamka, *A Software Page Fault Analyzer*, Masters Thesis, Computer Science Division, UC Davis.
14. S. J. Leffler, M. K. McKusick, M. J. Karles and J. S. Quarterman, *The Design and Implementation of the 4.3BSD UNIX Operating System*, Addison-Wesley (1989).
15. W. Malkawi and J. H. Patel, Performance measurement of paging behavior in multi-programmed systems. *Proc. of the 13th Annual Symposium on Computer Architecture*, 111–118 (1986).
16. W. M. McCormack and R. G. Sargent, Analysis of future event set algorithms for discrete event simulation. *Communications of the ACM* **24** (2), 801–812 (1981).
17. K. Mehlhorn, *Data Structures and Algorithms 1: Sorting and Searching*, Springer-Verlag (1984).
18. I. Mitrani, *Simulation Techniques for Discrete Event Systems*, Cambridge University Press (1982).
19. J. L. Paterson and A. Silberschatz, *Operating Systems Concepts*, Addison-Wesley, Reading, Mass. (1985).
20. C. M. Reeves, Complexity analyses of event set algorithms. *The Computer Journal* **27** (1), 72–79 (1984).
21. E. M. Reingold, J. Nievergelt and N. Deo, *Combinatorial Algorithms: Theory and Practice*, Prentice-Hall, Inc. Englewood Cliffs, N.J. (1977).
22. D. D. Sleator and R. E. Tarjan, Self adjusting binary trees. *Journal of the ACM* **32** (3), 652–686 (1985).
23. J. G. Vaucher, On the distribution of event times for the notices in a simulation event list. *INFOR* **15** (2), 171–182 (1977).
24. J. G. Vaucher and P. Duval, A comparison of simulation event list algorithms. *Communications of the ACM* **18** (4), 223–230 (1975).
25. A. I. Verkamo, Performance of quicksort adapted for virtual memory use. *The Computer Journal* **30** (4), 362–371 (1987).

Announcement