Operating Systems - Practical #5

Memory Management – part 1

Sébastien Vaucher

15 May 2015

Introduction

This report is the result of the fifth practical project of the course Operating Systems taught at the University of Neuchâtel. The goal is to implement a simulator of a *Memory Management Unit* (MMU) combined with a *Translation Look-aside Buffer* (TLB). They will be tested against memory traces of real programs, and the number of reads and writes to the physical memory will be monitored.

This homework is the first part of the practical exercise on memory management.

Analysis of results

In this section, we analyze the results presented in Table 1.

Using a TLB only improves the number of memory reads. The page table still needs to be written to the physical memory. Also, the MMU does not deal with memory accesses that are done by the user application.

We can observe that even a very small TLB (4 entries) already provides a substantial performance gain of $\sim 6.5\%$ on the number of read accesses. With a TLB 4 times this size, we get a reduction of $\sim 17.9\%$ on the number of read accesses. With a TLB of size 16 versus a TLB of size 4, the reduction in terms of number of read accesses is 2.8×.

Without a TLB, the number of memory accesses when combining multiple tasks is simply the sum of the number of memory accesses of the individual tasks. With a TLB, this is no longer the case for the number of read accesses. This is because the individual executions do not share access to the TLB, while with combinations of different programs, the TLB is shared.

The difference of efficiency between different programs is subtle. It depends on how often the program asks for the same page. The deletion policy of the TLB can help improve the efficiency by removing entries that are less likely to be accessed in the near future.

	Without TLB		TLB (size 4)			TLB (size 16)		
Workload	Reads	Writes	Reads	Writes	Hit Rate	Reads	Writes	Hit Rate
`ls`	1639244	312504	1557096	312504	0.063	1400810	312504	0.183
`echo`	1127276	222768	1065568	222768	0.069	960590	222768	0.185
`touch`	1469536	286012	1393180	286012	0.065	1254940	286012	0.183
`bc`	2881384	546292	2723902	546292	0.069	2496668	546292	0.168
`ls+touch`	3108780	598516	2953242	598516	0.063	2666342	598516	0.179
`ls+echo+touch`	4236056	821284	4018566	821284	0.065	3632924	821284	0.179
`ls+echo+touch+bc`	7117440	1367576	6745094	1367576	0.066	6136038	1367576	0.174

Table 1: Results with and without a TLB

Conclusion

Associating a TLB to an MMU greatly improves the performance of memory reads. The bigger the TLB the better. The performance also depends on the nature of the processes that are executed on the system.