

EOPSY Lab #4

Dominik Kurasbediani

302155

The task of this laboratory consisted of mapping any 8 pages of physical memory to the first 8 pages of virtual memory, reading from one virtual memory address on each of the 64 virtual pages.

With the simulator provided, it happens to be impossible to map only 8 pages and read from 64. The number of pages mapped (numpages in memory.conf) is the sum of number of virtual and physical pages mapped, e.g. numpages 64 maps 32 physical to 32 virtual pages.

Memory Management				
run	step	reset	exit	status: STOP
virtual	physical	virtual	physical	time: 0
page 0	page 0	page 32		instruction: NONE
page 1	page 1	page 33		address: NULL
page 2	page 2	page 34		page fault: NO
page 3	page 3	page 35		
page 4	page 4	page 36		virtual page: 63
page 5	page 5	page 37		physical page: -1
page 6	page 6	page 38		R: 0
page 7	page 7	page 39		M: 0
page 8	page 8	page 40		inMemTime: 0
page 9	page 9	page 41		lastTouchTime: 0
page 10	page 10	page 42		low: fc000
page 11	page 11	page 43		high: fffff
page 12	page 12	page 44		
page 13	page 13	page 45		
page 14	page 14	page 46		
page 15	page 15	page 47		
page 16	page 16	page 48		
page 17	page 17	page 49		
page 18	page 18	page 50		
page 19	page 19	page 51		
page 20	page 20	page 52		
page 21	page 21	page 53		
page 22	page 22	page 54		
page 23	page 23	page 55		
page 24	page 24	page 56		
page 25	page 25	page 57		
page 26	page 26	page 58		
page 27	page 27	page 59		
page 28	page 28	page 60		
page 29	page 29	page 61		
page 30	page 30	page 62		
page 31	page 31	page 63	page 31	

Initial setup

In the commands file, I READ the pagesize (16384) 64 times in order to cause a page fault to observe the page replacement algorithm. After it read through the 32 pages, it started

replacing them one by one in order of FIFO.

Memory Management				
run	step	reset	exit	status: STOP
virtual	physical	virtual	physical	time: 330 (ns)
page 0		page 32	page 0	instruction: READ
page 1	page 1	page 33		address: 80001
page 2	page 2	page 34		
page 3	page 3	page 35		page fault: YES
page 4	page 4	page 36		
page 5	page 5	page 37		virtual page: 32
page 6	page 6	page 38		physical page: 0
page 7	page 7	page 39		R: 0
page 8	page 8	page 40		M: 0
page 9	page 9	page 41		inMemTime: 10
page 10	page 10	page 42		lastTouchTime: 10
page 11	page 11	page 43		low: 80000
page 12	page 12	page 44		high: 83fff
page 13	page 13	page 45		
page 14	page 14	page 46		
page 15	page 15	page 47		
page 16	page 16	page 48		
page 17	page 17	page 49		
page 18	page 18	page 50		
page 19	page 19	page 51		
page 20	page 20	page 52		
page 21	page 21	page 53		
page 22	page 22	page 54		
page 23	page 23	page 55		
page 24	page 24	page 56		
page 25	page 25	page 57		
page 26	page 26	page 58		
page 27	page 27	page 59		
page 28	page 28	page 60		
page 29	page 29	page 61		
page 30	page 30	page 62		
page 31	page 31	page 63		

Step 33

Memory Management				
run	step	reset	exit	status: STOP
virtual	physical	virtual	physical	time: 430 (ns)
page 0		page 32	page 0	
page 1		page 33	page 1	instruction: READ
page 2		page 34	page 2	address: a8001
page 3		page 35	page 3	
page 4		page 36	page 4	page fault: YES
page 5		page 37	page 5	
page 6		page 38	page 6	virtual page: 42
page 7		page 39	page 7	physical page: -1
page 8		page 40	page 8	R: 0
page 9		page 41	page 9	M: 0
page 10		page 42	page 10	inMemTime: 0
page 11	page 11	page 43		lastTouchTime: 0
page 12	page 12	page 44		low: a8000
page 13	page 13	page 45		high: abfff
page 14	page 14	page 46		
page 15	page 15	page 47		
page 16	page 16	page 48		
page 17	page 17	page 49		
page 18	page 18	page 50		
page 19	page 19	page 51		
page 20	page 20	page 52		
page 21	page 21	page 53		
page 22	page 22	page 54		
page 23	page 23	page 55		
page 24	page 24	page 56		
page 25	page 25	page 57		
page 26	page 26	page 58		
page 27	page 27	page 59		
page 28	page 28	page 60		
page 29	page 29	page 61		
page 30	page 30	page 62		
page 31	page 31	page 63		

Step 43

FIFO (“First-In First-Out”) is a page replacement algorithm that uses the frame whose page has been in memory the longest. The page frames are kept in a queue and the frame that was used last is moved to the tail and in the next replacement the next page from the queue is used.

This is by no means an effective algorithm due to its inability to distinguish which pages are used frequently, and which are not used at all. A better algorithm for page replacement would be a priority queue.