

OPERATING SYSTEM – LABORATORY 4: MEMORY MANAGEMENT

Student Name: Phong Hai Do

Student ID: 309047

In the file named **memory.conf**, I map the first 8 pages physical memory to the first 8 pages virtual memory, and the rest of physical memory page (24 pages) will be mapped by default.

In the file named **command**, I read from one virtual memory address on each of the 64 virtual memory pages. Because the size of a page is 16384 (2^{14}) bytes, the first virtual page will cover all address from 0 to 16383, the second virtual page will cover all address from 16383 to 32767, and so on, the final virtual page – the 64th page will cover all address from 1032192 to 1048575. The **memory.conf** and the **command** file should be like this:

```
memset 0 0 0 0 0 0    READ 0
memset 1 1 0 0 0 0    READ 16384
memset 2 2 0 0 0 0    READ 32768
memset 3 3 0 0 0 0    READ 49152
memset 4 4 0 0 0 0    READ 65536
memset 5 5 0 0 0 0    READ 81920
memset 6 6 0 0 0 0    READ 98304
memset 7 7 0 0 0 0    READ 114688
```

Because there are in total 32 pages of physical memory and 64 pages of virtual memory, so all 32 pages of physical memory are mapped to the first 32 pages of virtual memory. When simulator tries to read from one specified virtual memory address from the 1st page to the 32nd page, there is no “page fault”, because all of these virtual pages are mapped to physical pages.

run	stop	reset	exit	status: STOP
virtual	physical	virtual	physical	time: 320 (ns)
page 0	page 0	page 32		instruction: READ
page 1	page 1	page 33		address: 7c000
page 2	page 2	page 34		page fault: NO
page 3	page 3	page 35		virtual page: 31
page 4	page 4	page 36		physical page: 31
page 5	page 5	page 37		R: 0
page 6	page 6	page 38		M: 0
page 7	page 7	page 39		inMemTime: 310
page 8	page 8	page 40		lastTouchTime: 310
page 9	page 9	page 41		low: 7c000
page 10	page 10	page 42		high: 7ffff
page 11	page 11	page 43		
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		

But when the simulator tries to access the 33rd virtual page and further, it will be a “page fault” because these virtual pages are not mapped to any physical pages. And then, a page replacement algorithm will be called to replace this current physical page. In the simulator GUI, the physical pages are replaced in the same order from the oldest to the newest (page 0, page 1, ..., page 31). Thus, the page replacement algorithm follows “First in, First out” strategy.

run	step	reset	exit	status: STOP	
virtual	physical	virtual	physical	time: 640 (ns)	
page 0		page 32	page 0		
page 1		page 33	page 1	instruction: READ	
page 2		page 34	page 2	address: fc000	
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: 63	
page 7		page 39	page 7	physical page: 31	
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 43	page 11	lastTouchTime: 0	
page 12		page 44	page 12	low: fc000	
page 13		page 45	page 13	high: fffff	
page 14		page 46	page 14		
page 15		page 47	page 15		
page 16		page 48	page 16		
page 17		page 49	page 17		
page 18		page 50	page 18		
page 19		page 51	page 19		
page 20		page 52	page 20		
page 21		page 53	page 21		
page 22		page 54	page 22		
page 23		page 55	page 23		
page 24		page 56	page 24		
page 25		page 57	page 25		
page 26		page 58	page 26		
page 27		page 59	page 27		
page 28		page 60	page 28		
page 29		page 61	page 29		
page 30		page 62	page 30		
page 31		page 63	page 31		

```

READ 6c000 ... okay
READ 70000 ... okay
READ 74000 ... okay
READ 78000 ... okay
READ 7c000 ... okay
READ 80000 ... page fault
READ 84000 ... page fault
READ 88000 ... page fault
READ 8c000 ... page fault
READ 90000 ... page fault

```

In this algorithm, the simulator (the operating system) keeps track of all pages in the memory in a queue, the oldest page is in the front of the queue. When a “page fault” occurs, the current page needs to be replaced, the front of the queue will be removed to place right this position.