

Warsaw University of Technology  
Faculty of Electronics and Information Technology



Kamil Borkowski - 300166

Laboratory task 4

Operating Systems – 2020/2021 Summer Semester

I declare that this piece of work is which is the basis for recognition of achieving learning outcomes in the Operating Systems course was completed on my own.

Kamil Borkowski  
300166  
06.06.2021

## Introduction

In this lab we are to get familiar with paged memory management with the use of MOSS Memory Management Simulator. 8 First pages of virtual memory will be mapped to physical memory and we'll read from an address on each of the 64 virtual pages. This will show us the behavior of page faults and the page replacement algorithm.

## What is virtual memory

If a computer runs out of physical memory it can map some portion of it into secondary memory (such as external storage devices). This part of the memory acts as if it was a part of the main memory and is called virtual memory. The implementation can be done via either paging or segmentation.

## What is paging

Paging consists in dividing virtual memory into blocks of identical size called pages, dividing the main memory into page frames and transferring these between virtual and physical memory. This is done with the use of a page table which contains the mapping between virtual and physical addresses and contains a frame table which keeps the information on which frames are mapped.

## What is page replacement and page fault

If, after checking the page table, we find out that the page we want to reference is not present in physical memory a "page fault" exception is raised and we need to decide which page to either page (or swap) in and which to page (or swap) out. This is done via one of many page replacement algorithms.

## Page replacement algorithm used

By checking the source files we can find in "PageFault.java" that the default page replacement algorithm is first-in first-out (FIFO). That means all pages in the memory are kept in a queue (with the oldest page at the front), and if there is a need for a page to be replaced, the front of the queue is selected.

## Comments

After assigning the first 8 virtual memory pages to physical memory pages we turn on MOSS (but not yet run it) and we can check by clicking on page 0-7 buttons that the physical memory pages were mapped correctly. The other virtual memory pages up to page 31 have been mapped automatically by the program.

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

Trying to read the pages up to page 31 all goes well, but then we reach page 32 which means we are trying to reference a virtual memory page which does not have a physical memory page mapped. This results in a page fault. The situation is the same for all pages >31. Since our page replacement algorithm is FIFO the page to get replaced is the oldest one, in our case the one that was mapped to virtual memory page 0, as it was the first one read and thus was in the front of the in the queue. After that we go in order and see what was mapped to virtual memory page number 1, 2, 3 and so on and repeat the process.

<a href="#">run</a>	<a href="#">step</a>	<a href="#">reset</a>	<a href="#">exit</a>	status: STOP
virtual	physical	virtual	physical	time: 450 (ns)
page 0		page 32	page 7	
page 1		page 33	page 6	instruction: READ
page 2		page 34	page 5	address: b0000
page 3		page 35	page 4	
page 4		page 36	page 3	page fault: YES
page 5		page 37	page 2	
page 6		page 38	page 1	virtual page: 0
page 7		page 39	page 0	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 43	page 11	lastTouchTime: 0
page 12		page 44	page 12	low: 0
page 13	page 13	page 45		high: 3fff
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		

We finish in a situation in which the original mapping between virtual and physical pages has been shifted by 32, and we have read from all 64 virtual pages addresses.

run	step	reset	exit	status: STOP
virtual	physical	virtual	physical	time: 640 (ns)
page 0		page 32	page 7	
page 1		page 33	page 6	instruction: READ
page 2		page 34	page 5	address: fc000
page 3		page 35	page 4	
page 4		page 36	page 3	page fault: YES
page 5		page 37	page 2	
page 6		page 38	page 1	virtual page: 0
page 7		page 39	page 0	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 43	page 11	lastTouchTime: 0
page 12		page 44	page 12	low: 0
page 13		page 45	page 13	high: 3fff
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	