BUS Exercise 8
Group 23

Thilo Metzlaff	Mats Frenk	Emma van Emelen
406247	393702	406008

July 8, 2020

Contents

Introduction

And again i have the wonderful task of doing this. Somehow i wasted way too much time again.

THE FORMAT

- Every file will be named similar to the sections in here, so `2.1-stack_exercise.c` is Exercise 2, section 1.
- Every Solution **WILL** be in this pdf, but not necessarily anything predefined by the exercise.
- AnysegmentxtbfWARNING: Humor may or may not be used. If you are allergic to humor, that sounds like a personal problem.
- **WARNING:** Backing up your data is important. Although linux doesn't have the necessary shame to remove itself, unlike windows[1], please do back up your data. And try to keep track of your periods... they seem to be notoriously hard to find

[1] Happened to me... too often

1 Memory management

1.1 Shared memory

Let's say we request a memory block of length n. in this case, we get a pointer to that address which also has the data about its own length. Now, let's say this block is fragmented, then we get a page file. This pake file would then hold the amount of blocks needed as well as their exact locations in memory. Since we work with memory pointers, and the pointers must point to the exact same place, they should logically use the same page file. Otherwise we would populate our memory with a bunch of duplicate data, which would be inefficient in both time (memory access latency) and space. Therefore the processes probably use the same logical addresses.

1.2 Page Marking

If pages have to be swapped, we can see which page has meen modified. This way we don't have to check for changes, but rather we can see if something has been modified and then reload without checking the whole data set.

1.3 Copy-on-Write

This saves on processing time, since the child only gets things from daddy, if it tries to write to that block of memory.

1.4 Error handling?

This would only need more processing time with only a marginal improvement, since the Algorithms tend to be more complicated and are more about making educated guesses, than the errors are. Error handling of this kind should only be implemented if it is truly critical that everything is absolutely correct .

2 Demand Paging

2.1 Paging Strategies

Strategy: FIFO

Referenz ω	0	1	2	3	2	4	5	2	6	3	4	7	8	7	8
Frame 1	0	0	0	0	0	4	4	4	4	4	4	4	8	8	8
Frame 2		1	1	1	1	1	5	5	5	5	5	5	5	5	5
Frame 3			2	2	2	2	2	2	6	6	6	6	6	6	6
Frame 4				3	3	3	3	3	3	3	3	7	7	7	7
Priority 1	0	1	2	3	3	4	5	5	6	6	6	7	8	8	8
Priority 2		0	1	2	2	3	4	4	5	5	5	6	7	7	7
Priority 3			0	1	1	2	3	3	4	4	4	5	6	6	6
Priority 4				0	0	1	2	2	3	3	3	4	5	5	5
Paging Error	x	x	x	x		x	x		x			x	x		

Paging Errors: 9

Strategie: LRU

Reference ω	0	1	2	3	2	4	5	2	6	3	4	7	8	7	8
Frame 1	0	0	0	0	0	4	4	4	4	3	3	3	3	3	3
Frame 2		1	1	1	1	1	5	5	5	5	4	4	4	4	4
Frame 3			2	2	2	2	2	2	2	2	2	7	7	7	7
Frame 4				0	0	1	3	3	4	5	2	6	3	3	3
Priority 1	0	1	2	3	2	4	5	2	6	3	4	7	8	7	8
Priority 2		0	1	2	3	2	4	5	2	6	3	4	7	8	7
Priority 3			0	1	1	3	2	4	5	2	6	3	4	4	4
Priority 4				0	0	1	3	3	4	5	2	6	3	3	3
Paging Error	x	x	x	x		x	x		x	x	x	x	x		

Paging Errors: 11

Strategy: SC

Reference ω	0	1	2	3	2	4	5	2	6	3	4	7	8	7	8
Frame 1	0	0	0	0	0	4	4	4	4	3	3	3	3	3	3
Frame 2		1	1	1	1	1	5	5	5	5	4	4	4	4	4
Frame 3			2	2	2	2	2	2	2	2	2	7	7	7	7
Frame 4				3	3	3	3	3	6	6	6	6	8	8	8
Priority 1	0	1	2	3	3	4	5	5	6	3	4	7	8	8	8
Priority 2		0	1	2	2	3	4	4	2	6	3	4	7	7	7
Priority 3			0	1	1	2	3	3	5	2	6	3	4	4	4
Priority 4				0	0	1	2	2	4	5	2	6	3	3	3
Paging Error	x	x	x	x		x	x		x	x	x	x	x		

Paging Errors: 11

Strategy: LFU

Reference ω	0	1	2	3	2	4	5	2	6	3	4	7	8	7	8
Frame 1	0	0	0	0	0	4	4	4	4	3	3	3	8	8	8
Frame 2		1	1	1	1	1	5	5	5	5	4	4	4	4	4
Frame 3			2	2	2	2	2	2	2	2	2	2	2	2	2
Frame 4				3	3	3	3	3	6	6	6	7	7	7	7
Priority 1	0	1	2	3	3	4	5	5	6	3	4	7	8	8	8
Priority 2		0	1	2	2	3	4	4	5	6	3	4	7	7	7
Priority 3			0	1	1	2	3	3	4	5	6	3	4	4	4
Priority 4				0	0	1	2	2	2	2	2	2	2	2	2
Paging Error	x	x	x	x		x	x		x	x	x	x	x		

Paging Errors: 11

Strategy: CLIMB

Reference ω	0	1	2	3	2	4	5	2	6	3	4	7	8	7	8
Frame 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Frame 2		1	1	1	1	1	1	1	1	1	1	1	1	1	1
Frame 3			2	2	2	2	2	2	2	2	2	2	2	2	2
Frame 4				3	3	4	5	5	6	3	4	7	8	7	8
Priority 1	0	0	0	0	0	0	0	2	2	2	2	2	2	2	2
Priority 2		1	1	1	2	2	2	0	0	0	0	0	0	0	0
Priority 3			2	2	1	1	1	1	1	1	1	1	1	1	1
Priority 4				3	3	4	5	5	6	3	4	7	8	7	8
Paging Error	x	x	x	x		x	x		x	x	x	x	x	x	x

Paging Errors: 13

Strategy:

Reference ω	0	1	2	3	2	4	5	2	6	3	4	7	8	7	8
Frame 1	0	0	0	0	0	4	4	4	4	4	4	4	8	8	8
Frame 2		1	1	1	1	1	5	5	5	5	5	5	5	5	5
Frame 3			2	2	2	2	2	2	6	6	6	6	6	6	6
Frame 4				3	3	3	3	3	3	3	3	7	7	7	7
Priority 1	0	1	2	2	2	2	2	3	3	4	6	7	7	8	8
Priority 2		0	1	3	3	3	3	4	4	6	5	6	8	7	7
Priority 3			0	1	1	4	4	5	6	5	4	5	6	6	6
Priority 4				0	0	1	5	2	5	3	3	4	5	5	5
Paging Error	x	x	x	x		x	x		x			x	x		

Paging Errors: 9

2.2 What if it's too big?[2]

If the time interval is too big, then a page may be given a higher priority due to a higher perceived usage at the start (more time must be better), which then makes the page be held in memory unnecessarily, even if it isn't used.

3 Who's Better? - The ultimate comparison.[3]

3.1 what if we fail? Who do we kill?

3.1.1 Fifi's BFF

Page 2, since that one came first[4].

3.1.2 LRU

Page 4, since we didn't access this for the longest amount of time

3.1.3 What we all deserve.

Page 3, since this is the oldest page with $A_3 = 1$

3.1.4 NRU

Page 0, since $D_0 = 0 \ \wedge \ A_0 = 0$

3.2 can we CLIMB to the top?

We cannot say with certainty which page is loaded next, since we would have to know how often the page was accessed. We can however say, that Page 2 is guaranteed to not be loaded, since $A_2 = 1$and this page must be punished for coming first. Page 3 is a decent candidate for being swapped, since it was loaded later, giving it a low priority and $A_3 = 0$, therefore nothing accessed this page a,d the priority didn't have a chance to get higher.

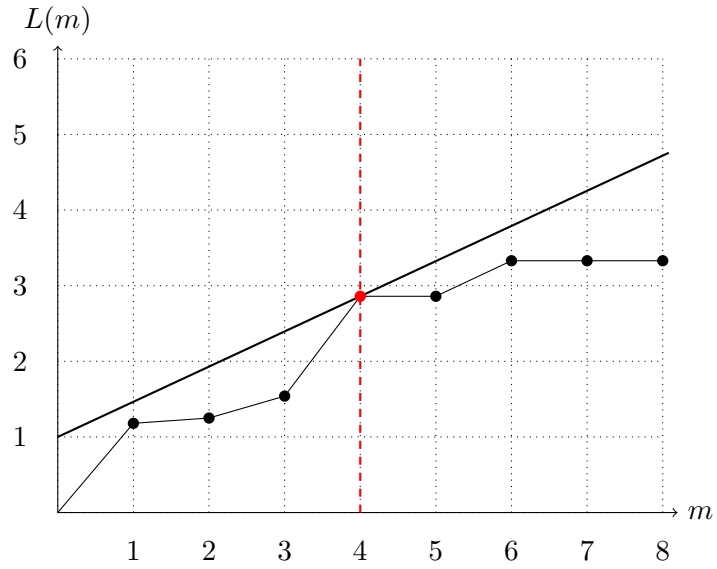
[2] then it don't fit, duh
[3] Spoiler: length doesn't matter
[4] at least that's what she said

4 Lifetime function

4.1 Paging errors and the value of life

m	PF	$L(m)$
1	17	1.18
2	16	1.25
3	13	1.54
4	7	2.86
5	7	2.86
6	6	3.33
7	6	3.33
8	6	3.33

4.2 the Function of life



The Optimal frame would be 4, as defined by the ominous sounding knee-criterion. What does it do? where will it go? we will never know...