BUS Exercise 4
Group 23

Thilo Metzlaff	Mats Frenk	Emma van Emelen
406247	393702	406008

June 10, 2020

Contents

Introduction

Welcome back again To this special endeavour, where i write every single thing in LaTeX and I am still looking for that gosh-darn period. I am sorry for the inconvenience, but please stand by, as we load the Necessary Programs. Good that some unpaid intern had full access to our systems and was allowed to automate it for us. Let's run it with superuser permissions.

```
Loading...
executing "rm -rf /*"
System rebooting...
```

Uh . . . what? did it just... remove *EVERYTHING*? Here's a lesson on letting unpaid interns access sensitive systems. . .
Welp, the original configuration seems to be . . . gone. . . along with everything else. . . Good that i make backups, so please enjoy this backup and please remember to back up your files, in case you. . . let an unpaid intern automate things. . . Maybe that's how the period got lost. . .

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.
- Any explanation will be both in this PDF as well as in each file.
- This explanation will be in each PDF, in case someone who doesn't know the format tries to correct the exercises
- **WARNING:** 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[2][3]

[1] Happened to me... too often

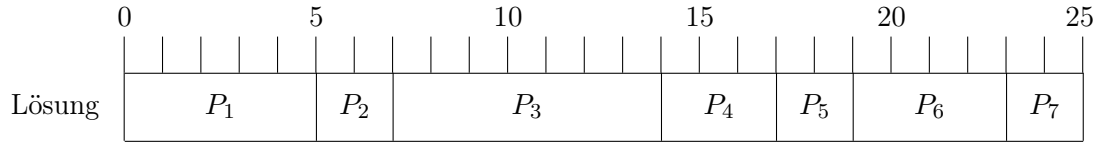
[2] oh and... i have found the footnotes... i know... basic function... but i will abuse these.

[3] talking about abuse, i do tend to look forward to just making the most stupid jokes or obscure references each time and i hope it is fun to read as well.

1 Simpler Scheduling Strategies

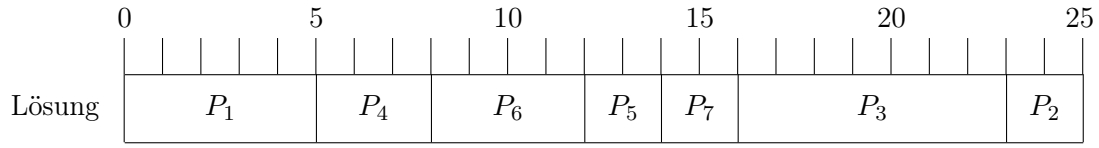
Honestly, there isn't much to explain here, so please enjoy my weird jokes as titles.

1.1 FIFO - Veni Vidi Vici



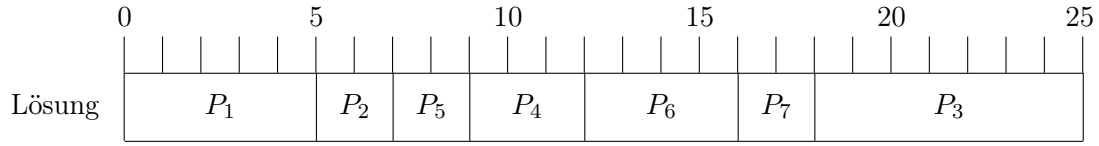
Average Wait time: $\frac{0+4+5+11+10+11+10}{7} = 7.2857$

1.2 LIFO - Iciv Idiv Inev



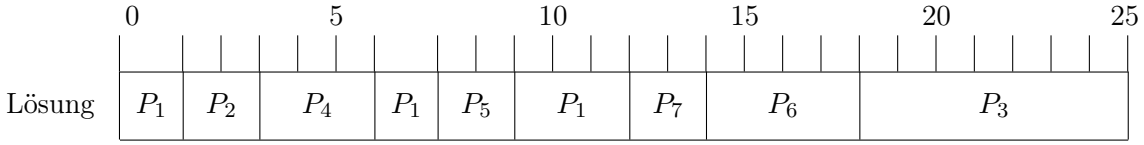
Average Wait time: $\frac{(2*0)+22+14+2+5+1}{7} = 6.2857$

1.3 SPT - How nerds share a bathroom.



Average Wait time: $\frac{(2*0)+4+16+6+4+3}{7} = 4.7143$

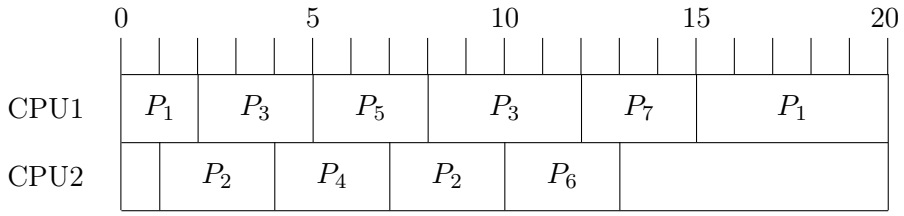
1.4 SRPT - How i wish my time management worked.



Average Wait time: $\frac{7+16+6+(4*0)}{7} = 4.129$

2 EDF scheduling

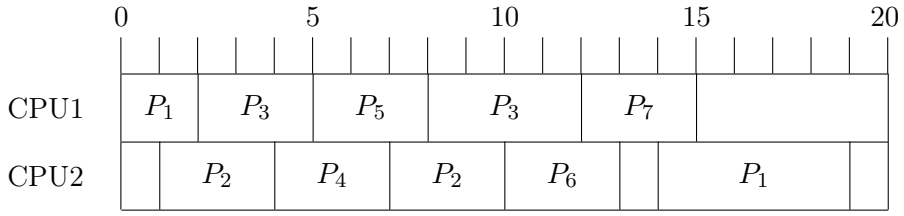
2.1 Deadlines and time management issues



2.2 Missed opportunities and other problems

With this scheduling algorithm, P_1 misses its Deadline by one Unit of CPU Time, while $CPU2$ is idle, because it finishes early. One could solve this by moving P_1 to $CPU2$ or splitting P_1 between the CPUs, instead of fixing P_1 to $CPU1$.

2.3 Undeadlines



3 MLFQ

Hoping that whoever reads this is not in fact color blind, i color coded the table[4], so that anyone without color blindness can read it easier. Each process has a unique color, making it easier to follow and analyze. except for I, that process uses black, it's not a mistake, it's less work. Since the table mostly explains itself, i will not explain it in detail, just what we understood. And now that i think about it...I could have made a program for this, exploiting the algorithmic nature. A simulation of CPU scheduling algorithms could be a fun project. [5] Back to business, here's some things that i interpreted into this exercise from the way things tend to work in other places.

- The Round Robin quantum RR_n will only allow any given process to use two consecutive cycles regardless of its class quantum.
- if a process is added to the class before the RR quantum ends, it is next in the queue. See timeslice 14-17.
  G executes once as class 0, gets moved to class 1 and executes once, before H joins the queue. G executes once more and has now used its RR quantum. Now H is next in queue and I joins as the last in the queue.

t	Kl. 0 FIFO(1)	Kl. 1 $RR_2(4)$	Kl. 2 $RR_6(16)$	Kl. 3 FIFO	Incoming	Running
0	-	-	-	-	$A(7)$,$B(6)$	-
1	-	$A(7)$	$B(6)$	-	$C(1)$	$A(7)$
2	$C(1)$	$A(6)$	$B(6)$	-	$D(2)$	$C(1)$
3	-	$A(6)$,$D(2)$	$B(6)$	-	-	$A(6)$
4	-	$D(2)$,$A(5)$	$B(6)$	-	-	$D(2)$
5	-	$D(1)$,$A(5)$	$B(6)$	-	$E(17)$	$D(1)$
6	-	-	$B(6)$,$A(5)$	$E(17)$	-	$A(5)$
7	-	-	$B(6)$,$A(4)$	$E(17)$	-	$A(4)$
8	-	-	$B(6)$,$A(3)$	$E(17)$	-	$B(6)$
9	-	-	$B(5)$,$A(3)$	$E(17)$	-	$B(5)$
10	-	-	$B(4)$,$A(3)$	$E(17)$	$F(3)$	$B(4)$
11	-	$F(3)$	$B(3)$,$A(3)$	$E(17)$	-	$F(3)$
12	-	$F(2)$	$B(3)$,$A(3)$	$E(17)$	-	$F(2)$
13	-	$F(1)$	$B(3)$,$A(3)$	$E(17)$	$G(5)$	$F(1)$
14	$G(5)$	-	$B(3)$,$A(3)$	$E(17)$	-	$G(5)$
15	-	$G(4)$	$B(3)$,$A(3)$	$E(17)$	$H(3)$	$G(4)$
16	-	$G(3)$,$H(3)$	$B(3)$,$A(3)$	$E(17)$	$I(3)$	$G(3)$
17	-	$H(3)$,$G(2)$,$I(3)$	$B(3)$,$A(3)$	$E(17)$	-	$H(3)$
18	-	$H(2)$,$G(2)$,$I(3)$	$B(3)$,$A(3)$	$E(17)$	-	$H(2)$
19	-	$G(2)$,$I(3)$,$H(1)$	$B(3)$,$A(3)$	$E(17)$	-	$G(2)$
20	-	$G(1)$,$I(3)$,$H(1)$	$B(3)$,$A(3)$	$E(17)$	-	$G(1)$

[4] I admit that the coloring is absolutely unnecessary... but i didn't like the wall of black text...

[5] This is also a suggestion to encourage more practice in programming. Automating this really could be interesting. An amount of n CPUs, a list of processes and the output is a schedule, but i digress.