BUS Exercise 6
Group 23

Thilo Metzlaff	Mats Frenk	Emma van Emelen
406247	393702	406008

June 24, 2020

Contents

Introduction

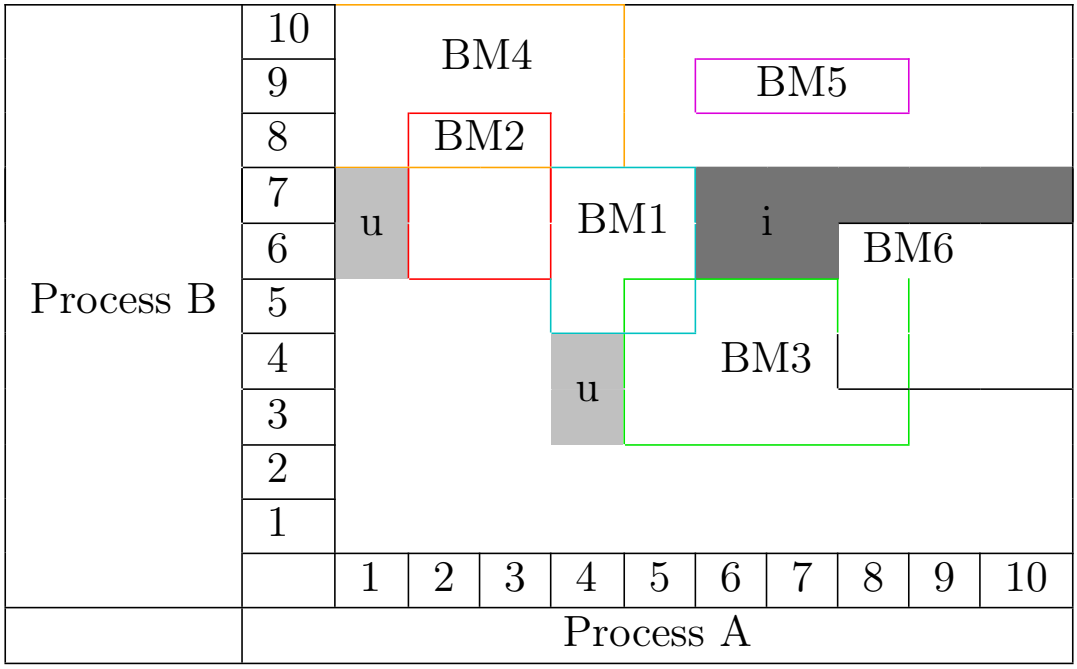
And `A G A I N`, welcome to this document of absolute bullshittery, written in LaTeX. This will be one of my Low-Effort documents, with low-effort answers. Why? idk, i just feel like it. Having absolutely no energy REALLY doesn't make it better... but i shall prevail. With the help of premade solutions by a friend, because i truly have no time or energy to even try. Humor will be limited.

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, please do back up your data. And try to keep track of your periods... they seem to be notoriously hard to find

1 Things i can't process

1.1 The infinity Square



u = unsafe; n = impossible

1.2 The no effort name for the no effort answer

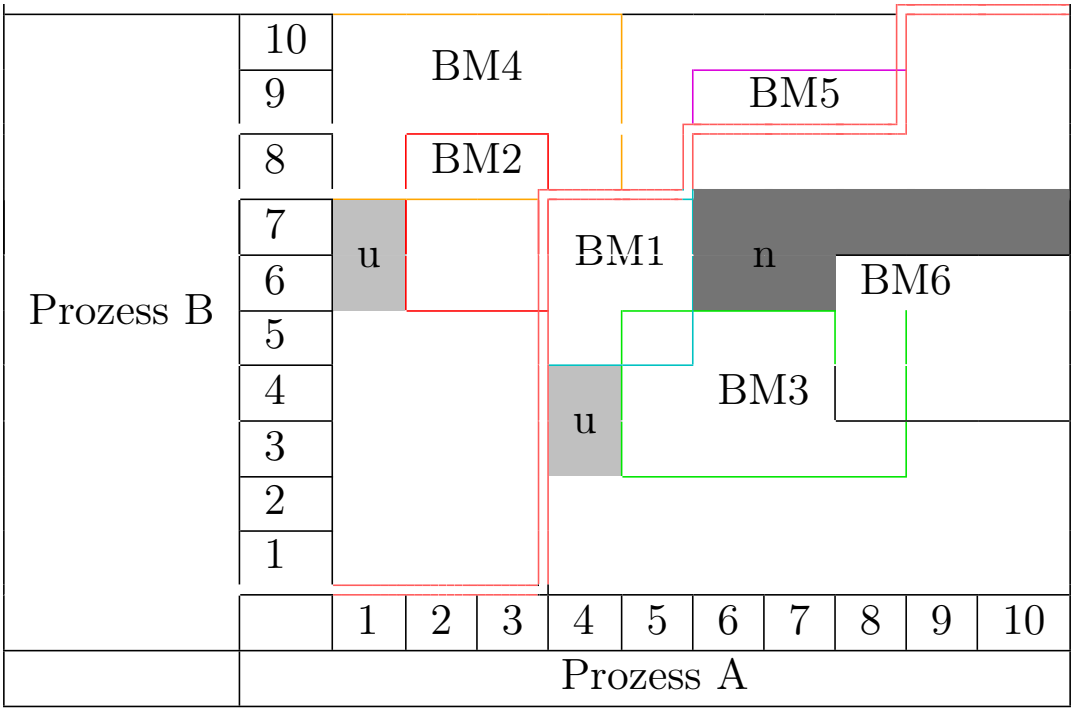
$(6, 4)$ - unmöglich

$(4, 3)$ - unsicher

$(9, 8)$ - sicher

$(6, 6)$ - unmöglich

1.3 Pathfinding and walking around processes



Schedule: AAABBBBBBBAABAAABBAA

2 Banker Algorithm

2.1 Dreadlock prevention

1. P_1 is unmarked
    1. Test: $Q_1^{max}(k) = (2,4,2) \leq V(k) = (3,3,4) \rightarrow nicht$
2. P_2 is unmarked
    1. test: $Q_2^{max}(k) = (4,1,1) \leq V(k) = (3,3,4) \rightarrow nicht$
3. P_3 is unmarked
    1. test: $Q_3^{max}(k) = (2,1,4) \leq V(k) = (3,3,4)$
    2. $V(k) = V(k) + H_3(k) = (5,3,4)$
    3. Markiere P_3
4. P_1 is unmarked
    1. test: $Q_1^{max}(k) = (2,4,2) \leq V(k) = (5,3,4)$
    2. $V(k) = V(k) + H_2(k) = (6,5,4)$
    3. Markiere P_1
5. P_2 is unmarked
    1. test: $Q_2^{max}(k) = (4,1,1) \leq V(k) = (6,5,4)$
    2. $V(k) = V(k) + H_2(k) = (6,6,4)$
    3. Markiere P_2

$\Rightarrow$ All processes are marked, therefore the state is safe.

2.2 Evil Dreadlocks

1 .)

1.1 $Q_1^{akt}(k) = (0,0,1) \leq (3,3,4) = V(k)$

1.2 Trying:

$$V(k) = V(k) - Q_1^{akt}(k) = (3,3,3)$$
$$H_1(k) = H_1(k) + Q_1^{akt}(k) = (1,2,1)$$
$$Q_1^{max}(k) = Q_1^{max}(k) - Q_1^{akt}(k) = (2,4,1)$$

1.3 Security test:
P_3 cannot be executed anymore: $Q_3^{max}(k) \lneq V(k) = (3,3,3)$
P_2 cannot be executed anymore: $Q_2^{max}(k) \lneq V(k) = (3,3,3)$
P_1 cannot be executed anymore: $Q_1^{max}(k) \lneq V(k) = (3,3,3)$

$\Rightarrow$ unsafe state

2 .)

2.1 $Q_1^{akt}(k) = (0,4,0) \lneq (3,3,4) = V(k)$
Q_1^{akt} cannot be executed $\Rightarrow$ unsafe/deadlock

3 .)

3.1 $Q_2^{akt}(k) = (1,0,0) \leq (3,3,4) = V(k)$

3.2 Trying:

$$V(k) = V(k) - Q_2^{akt}(k) = (2,3,4)$$
$$H_2(k) = H_2(k) + Q_2^{akt}(k) = (1,1,0)$$
$$Q_2^{max}(k) = Q_2^{max}(k) - Q_2^{akt}(k) = (3,1,1)$$

3.3 Security test:
P_3 cannot be executed anymore:
$Q_3^{max}(k) \lneq V(k) = (2,3,4) \rightarrow V(k) = (4,3,4)$
P_2 cannot be executed anymore:
$Q_2^{max}(k) \lneq V(k) = (4,3,4) \rightarrow V(k) = (5,5,4)$
P_1 cannot be executed anymore:
$Q_1^{max}(k) \lneq V(k) = (5,5,4) \rightarrow V(k) = (6,6,4)$
$\Rightarrow$ safe state

2.3

Assumption P_1 can run BM_2 at most twice at time k.
Because $V(k) - (0,2,0) = (3,1,4)$
And $Q_3^{max} \leq (3,1,4) \Rightarrow$ Banker algoritm is executable like in *subsection 2.1*.
And $V(k) - (0,3,0) = (3,0,4)$
And Q_3^{max} ! $\leq (3,0,4) \Rightarrow$ Banker algoritm is not executable like in *subsection 2.1*

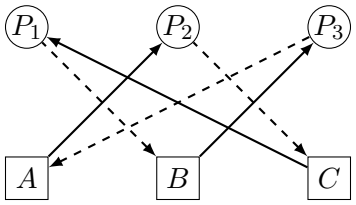
3 Resource Allocation Graphs

3.1 The fun RAG

	1	2	3	4	5	6	7	8	9	10
P_1	A+	A, c	A, c	A, C+	A, C	A-, C	C, b	C, b	C, b	C, b
P_2		a	a	a	a	a	A+	A	A, c	A, c
P_3	C+	C	C-	B+	B, a	B, a	B, a	B,a	B, a	B,a

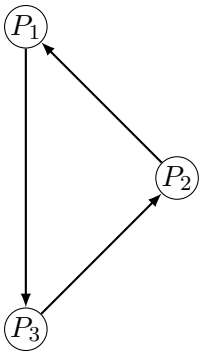
3.2 Graphs to the people

3.2.1 Resource Allocation Graph



Dotted lines are requests, whereas solid lines are allocations.

3.2.2 LEGEN- ... Wait-For it...

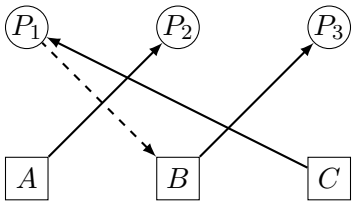


- circular wait: see "Wait for graph" ⇒ therefore we have a deadlock.
- Non-Preemption: as defined by Exercise
- Hold and wait: as defined by Exerciseg
- Exclusive Use: as defined by Exercise

3.3 Resource unavailable.

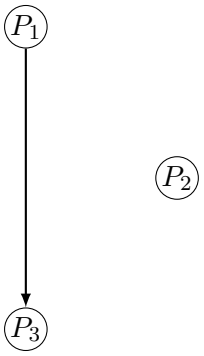
	1	2	3	4	5	6	7	8	9	10
P_1	A+	A, c	A, c	A, C+	A, C	A-, C	C, b	C, b	C, b	C, b
P_2		a	a	a	a	a	a	a	a	A+
P_3	C+	C	C-	B+	B, a	B, a	B, A+	B,A	B, A-	B-

3.3.1 RAG time



Dotted lines are requests, whereas solid lines are allocations.

3.3.2 wait for it... -DARY



- circular wait: doesn't exist, see "Wait for graph"

$\Rightarrow$ therefore we have a deadlock.

3.4 The end

Due to the fact that P_3 terminates in step 10, B is free which lets all processs Terminate. Therefore P_1 doesn't have to wait for B and can rest in piece, after finishing. Now P_3 can run without interruptions.