

OS 2021 Problem Sheet #5

Problem 5.1: bank transfers

(2+2 = 4 points)

You are implementing account management software and a fragment of your source code looks like this:

```
#include <pthread.h>

typedef struct account {
    unsigned int number;
    unsigned int money;
    pthread_mutex_t lock;
} account_t;

void transfer(unsigned int money, account_t *from, account_t *to)
{
    pthread_mutex_lock(&from->lock);
    pthread_mutex_lock(&to->lock);

    from->money -= money;
    to->money += money;

    pthread_mutex_unlock(&to->lock);
    pthread_mutex_unlock(&from->lock);
}
```

- Describe a situation in which this code can deadlock and provide a proof why this code can deadlock.
- Is it possible to change the code such that deadlocks can be prevented? Show how or why this is not possible.

Problem 5.2: safe states

(3 points)

A system has $n = 5$ processes, $m = 5$ resource types, and the number of resources for each resource type is given by $t = (6, 17, 9, 10, 7)$. The system is in the following state:

$$M = \begin{bmatrix} 2 & 5 & 3 & 3 & 2 \\ 3 & 5 & 8 & 10 & 1 \\ 4 & 12 & 4 & 9 & 2 \\ 6 & 1 & 4 & 5 & 5 \\ 1 & 2 & 3 & 4 & 5 \end{bmatrix} \quad A = \begin{bmatrix} 0 & 5 & 3 & 1 & 1 \\ 0 & 2 & 1 & 1 & 1 \\ 0 & 7 & 1 & 2 & 1 \\ 3 & 1 & 1 & 1 & 0 \\ 1 & 2 & 3 & 2 & 1 \end{bmatrix}$$

Is the system in a safe state? Provide a calculation to justify your answer.

Problem 5.3: deadlock detection

(1+2 = 3 points)

A system has $n = 3$ processes, $m = 4$ resource types, and the number of resources for each resource type is given by $t = (2, 2, 1, 3)$. The system is in the following state:

$$A = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad N = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

- a) Draw the corresponding resource allocation graph.
- b) Is the system deadlocked? Provide a calculation to justify your answer.