NTNU
Norwegian University of
Science and Technology

Department of Computer Science – IDI TDT4186 Operating Systems

https://folk.ntnu.no/michaeng/tdt4186_22/
michael.engel@ntnu.no

Theoretical exercises
Spring 2022

Theoretical Exercises 3

Deadlocks and Software Development Process

Please submit solutions on Blackboard by Friday, 18.2.2022 12:00h

3.1 Deadlocks in real life

We have seen the crossroads example to demonstrate the problem in the lecture. List three other examples of deadlocks that are *not* related to a computer system environment.

3.2 Resource allocation graphs

Suppose that there is a resource deadlock in a system. Give an example to show that the set of processes deadlocked can include processes that are not in the circular chain in the corresponding resource allocation graph.

3.3 Deadlock conditions

Two processes, A and B, each need three records, 1, 2, and 3, in a database.
If both A and B request the records in the order 1, 2, 3, deadlock is not possible. However, if B asks for the records in the order 3, 2, 1, then a deadlock can occur.
With three resources, there are 3! = 6 possible combinations each process can request resources.
What fraction of all combinations is guaranteed to be deadlock free?

3.4 Banker's algorithm

Consider a system that uses the banker's algorithm to avoid deadlocks. At some time a process P requests a resource R, but is denied even though R is currently available. Does it mean that if the system allocated R to P, the system would deadlock?

3.5 C preprocessor

You want to define a C preprocessor macro to calculate the square of a given number x as follows:

```
#define SQUARE(x) (x * x)
```

Explain what is problematic with this macro definition and give an example of the problematic behavior.

3.6 ELF segments

You are trying to analyze a binary program using the command `readelf -S prog` and obtain the following output (shortened):

```
Section Headers:
  [Nr] Name              Type             Address           Offset
       Size              EntSize          Flags  Link  Info  Align
  [25] .data             PROGBITS         0000000000004000  00003000
       0000000000000010  0000000000000000  WA       0     0     8
```

Assume that you know that there are only global `int` variables and each variable uses four bytes. Can you tell how many global `int` variables are declared in the program?

3.7 ELF symbols

Consider the following (very simple and useless) C program:

```
int foo;
int bar;

int main(int argc, char **argv) {
  int a, b;
}
```

- Which ELF segment will the variables `foo` and `bar` be located in?
- When running the `nm` command on the binary compiled from the program, variables `a` and `b` are not shown in the command's output. Explain why.