Name	Signature	Student Id Nr	*Points*

Operating Systems, miniexam 4, 23.2.2015 (6p)

Write your answer on this exam paper in the space given. Please notice, that the exam paper is 2-sided.

a) [2 p] Dijkstra's Deadlock Detection Algorithm (DDA).
What kind of environment is DDA used in? When is DDA started and who starts it?

What do you do, if DDA finds a deadlock?
(Note: There is no need to explain how DDA works!)

b) [1 p] Think about a solution to the Dining Philosophers Problem, where forks (nr 1,2,3,4,5) are always reserved in ascending fork number order. Can this solution cause a deadlock? Give a scenario leading to a deadlock, or prove why deadlock cannot occur.

TURN

Editor and keyboard driver. Text editor (TE) reads character buffer (B) one character at a time, and then makes the needed changes to the file being edited. Keyboard device driver (DD) reads the pressed keys (one at a time) and then writes the corresponding characters to the buffer B. Buffer B contains 200 characters.

Routines Put(buf, c) and c=Get(buf) are used to move data into the buffer and from it. They must not be executed concurrently, if they access the same buffer.

c) [2 p] Give the solution to this synchronization and communication problem for TE and DD using monitor (M).

The pseudocodes for DD and TE in this monitor solution are

DD

```
while (true)
    c = "next character pressed"
    M.BufWrite (c);
```

TE

```
while (true)
    c = M.BufRead();
    "use c, edit file"
```

Present the pseudocode for monitor M. You can invoke routines Put() and Get() from your monitor methods BufWrite() and BufRead(). Monitors in your system use Hoare signaling semantics (signal and wait). Declare clearly all your condition variables and other data structures (needed for synchronization) with their initial values.

d) [1 p] Assume now, that monitors in your system use Lampson & Redell signalling semantics (signal and continue). How does that affect your monitor solution in previous case (c)? Explain.