Name	Signature	Student Id Nr	*Points*

Operating Systems, miniexam 2, 13.2.2017 (6p)

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

a) [2 p] How do kernel level threads (KLT) differ from user level threads (ULT)?

Give a situation where it would make sense to use KLTs and not ULTs. Explain.

Give a situation where it would make sense to use ULTs and not KLTs. Explain.

b) [1 p] The initial value for shared variable X is zero (0). The system has one processor (core). Threads A and B in multithreaded program P execute to following (machine language) code segments once, each their own segment:

```
Thread A                                Thread B
100:  ...                               250:  ...
101:  LOAD R1, X      ; R1 ← mem(X)     251:  LOAD R2, X
102:  ADD  R1, =1     ; R1++            252:  SUB  R2, =1     ; R1--
103:  STORE R1, X     ; mem(X) ← Ri     253:  STORE R2, X
104:  ...                               254:  ...
```

The intent is for A to increment the value of X by one, B to decrement it by one, and final value of X to be zero. However, program P is erroneous and does not work properly in all scenarios. Give a scenario that proves P to be erroneous and where final value of X is -1.

TURN

c) [1 p] What is wrong with program P in question (b)?

How do Dekker and Peterson algorithms relate to this problem?

d) [2 p] Running track and semaphores. Running track is 400m long. Ann and her friends Bill, Charlie, and Dave come there often and run 4000m. Ann is social and waits after every lap that all boys have caught up with her (equal number of laps). The boys are competitive and do not wait for anybody. Solve the resulting synchronization problem with semaphores. Give your solution by modifying the runner pseudocodes given below. Remember to define your semaphores, with their initial values.

```
Ann
for (i=1 to 10)
    <run lap>
    <synchronize>
```

```
Bill
for (i=1 to 10)
    <run lap>
    <synchronize>
```

```
Charlie
for (i=1 to 10)
    <run lap>
    <synchronize>
```

```
Dave
for (i=1 to 10)
    <run lap>
    <synchronize>
```