Written examination

Operating Systems

July 22nd 2019

Last name: _____

First name: _____

Student number: _____

I confirm with my signature that I will process the written examination alone and that I feel healthy and capable to participate this examination.
I am aware, that from the moment, when I receive the written examination, I am a participant of this examination and I will be graded.

Signature: _____

- Use the provided sheets. Own paper must *not* be used.
- You are allowed to use a *self prepared*, *single sided DIN-A4 sheet* in the exam. Only *handwritten originals* are allowed, but no copies.
- You are allowed to use a non-programmable calculator.
- Do *not* use a red pen.
- The time limit ist *90 minutes*.
- Turn off your mobile phones!

Result:

Question:	1	2	3	4	5	6	7	8	9	10	11	**Σ**	Grade
Maximum points:	8	14	8	4	10	8	4	8	10	9	7	90	—
Achieved points:													

1.0: 90.0-85.5, **1.3**: 85.0-81.0, **1.7**: 80.5-76.5, **2.0**: 76.0-72.0, **2.3**: 71.5-67.5,
2.7: 67.0-63.0, **3.0**: 62.5-58.5, **3.3**: 58.0-54.0, **3.7**: 53.5-49.5, **4.0**: 49.0-45.0, **5.0**: <45

Last name: First name: Student number:

Question 1)

Points: .....

Maximum points: 1+1+2+2+2=8

a) At any given moment, only a single program can be executed. What is the technical term for this operation mode?

b) What is the name of the quasi-parallel program or process execution?

c) Describe the structure of a monolithic kernel.

d) Describe the structure of a microkernel.

e) Describe the structure of a hybrid kernel.

Last name: First name: Student number:

Question 2)

Points: .....

Maximum points: 1+2+2+3+6=14

a) Name the two groups of Input/Output devices for computer systems that are distinguished according to their minimum transfer unit.

b) Describe the different operating principles of the groups of subtask a).

c) Name two examples for each group from subtask a).

d) Name three possible ways for processes to read data from Input/Output devices.

e) Name a benefit and a drawback for each possible way from subtask d).

Last name: First name: Student number:

Question 3)

Points: .....

Maximum points: 2+2+2+2=8

A hard disk drive provides these information:

```
IBM Travelstar                                  MODEL: DBCA-204860 E182115 T
RATED: 5V 500mA                               MADE IN THAILAND BY IBM STORAGE
P/N: 21L9510   4090 MB                                                16NOV99
FRU: 22L0018   MLC:F41941                    (7944 CYL. 16 HEADS. 63 SEC/T)
```

a) Calculate the capacity of one side of one disk of the hard disk drive.
(Provide the calculation steps!)

Note: The number of cylinders (`CYL`) is equal to the number of tracks per disc. The size of the sectors (`SEC`) is 512 Byte.

b) Calculate the capacity of one track of the hard disk drive.
(Provide the calculation steps!)

c) Calculate the total capacity of the hard disk drive.
(Provide the calculation steps!)

d) How many disks does the hard disk drive have? *Note: Each disk has two sides.*
(Explain your answer!)

Last name: First name: Student number:

Question 4)

Points: .....

Maximum points: 4

Please mark for each one of the following statements, whether the statement is true or false.

a) Real mode is suited for multitasking systems.

☐ True ☐ False

b) In protected mode, each process is executed in its own copy of the physical address space, which is protected from other processes.

☐ True ☐ False

c) When static partitioning is used, internal fragmentation occurs.

☐ True ☐ False

d) When dynamic partitioning is used, external fragmentation cannot occur.

☐ True ☐ False

e) With paging, all pages have the same length.

☐ True ☐ False

f) One advantage of long pages is little internal fragmentation.

☐ True ☐ False

g) A drawback of short page page table can become huge.

☐ True ☐ False

h) When paging is used, the MMU translates the logical memory addresses into physical memory addresses.

☐ True ☐ False

Last name: First name: Student number:

Question 5)

Points: .....

Maximum points: 10

a) Describe which information inodes store.

b) Name three examples of metadata in the file system.

c) Describe what a cluster in the file system is.

d) Describe how a UNIX file system (e.g. ext2/3), which does not implement extents, can address more than 12 clusters.

e) Describe how directories in the Linux file systems are technically implemented.

f) Most operating systems operate according to the principle...

☐ write-back ☐ write-through

g) `/home/<username>/Mail/inbox/` is an/a...

☐ absolute path name ☐ relative path name

h) Describe what information the boot sector of a file system stores.

i) Describe what information the super block of a file system stores.

j) Explain why some file systems (e.g. ext2/3) do combine the clusters of the file system to block groups.

Last name: First name: Student number:

Question 6)

Points: .....

Maximum points: 2+1+1+3+1=8

a) Describe what the File Allocation Table (FAT) is and describe the information it stores.

b) Describe the objective of the journal in a journaling file system.

c) Describe a benefit of using a journaling file system compared with using a file system without a journal.

d) Name the three values that are required to store an extent.

e) Describe the benefit of using extents compared with direct addressing of the clusters.

Last name: First name: Student number:

Question 7)

Points: .....

Maximum points: 4

a) Describe the result of defragmenting a file system.

b) Describe the sort of data processing that is maximum accelerated by defragmenting.

c) Describe the scenario where defragmenting is useful.

d) Does defragmenting SSDs make sense? *(Explain your answer!)*

Last name: First name: Student number:

Question 8)

Points: .....

Maximum points: 8

a) Describe the effect of calling the system call `fork()`.

b) Describe the effect of calling the system call `exec()`.

c) Describe what `init` is and what its task is.

d) Name the differences of a child process from the parent process shortly after its creation.

e) Describe the effect, when a parent process is terminated before the child process.

f) Describe what data the Text Segment contains.

g) Describe what data the Heap contains.

h) Describe what data the Stack contains.

Last name: First name: Student number:

Question 9)

Points: .....

Maximum points: 6+2+2=10

a) Explain how multilevel feedback scheduling works.
(*An illustration can be useful here.*)

b) Name four scheduling strategies that are fair.

c) Name four scheduling strategies that do not need to know the *execution time* of the processes.
(Note: Only those scheduling procedures are searched, that can be used under realistic conditions.)

Last name: First name: Student number:

Question 10)

Points: .....

Maximum points: 2+7=9

a) Mark four conditions that must be fulfilled at the same time as precondition that a deadlock can occur.

- ☐ Recursive function calls
- ☐ Mutual exclusion
- ☐ Frequent function calls
- ☐ Nested `for` loops
- ☐ No preemption
- ☐ Hold and wait
- ☐ > 128 processes in `blocked` state
- ☐ Iterative programming
- ☐ Circular wait
- ☐ Queues

b) Does a deadlock occur?
Perform the deadlock detection with matrices.

$$\text{Existing resource vector} = \begin{pmatrix} 4 & 8 & 6 & 6 & 5 \end{pmatrix}$$

$$\text{Current allocation matrix} = \begin{bmatrix} 0 & 2 & 1 & 0 & 0 \\ 2 & 3 & 1 & 0 & 4 \\ 1 & 0 & 2 & 1 & 1 \end{bmatrix} \qquad \text{Request matrix} = \begin{bmatrix} 3 & 3 & 2 & 4 & 5 \\ 0 & 3 & 1 & 4 & 0 \\ 0 & 2 & 3 & 5 & 4 \end{bmatrix}$$

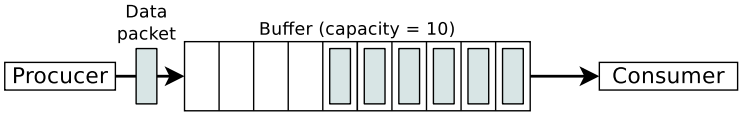
Last name: First name: Student number:

Question 11)

Points: .....

Maximum points: 7

- A producer writes data into a buffer and the consumer removes it.
- Mutual exclusion is necessary in order to avoid inconsistencies.
- If the buffer has no more free capacity, the producer must be blocked.
- If the buffer is empty, the consumer must be blocked.



For synchronizing the two processes, create the required semaphores, assign them initial values and insert semaphore operations.

```
typedef int semaphore;          // semaphores are of type integer



void producer (void) {
  int data;
  while (TRUE) {                // infinite loop
    createDatapacket(data);     // create data packet


    insertDatapacket(data);     // write data packet into the buffer


  }
}

void consumer (void) {
  int data;
  while (TRUE) {                // infinite loop


    removeDatapacket(data);     // pick data packet from the buffer


    consumeDatapacket(data);    // consume data packet
  }
}
```