Sample solution of the written examination in

Operating Systems

February 17th 2025

Last name: _____

First name: _____

Student number: _____

Mit dem Bearbeiten dieser schriftlichen Prüfung (Klausur) bestätigen Sie, dass Sie diese alleine bearbeiten und dass Sie sich gesund und prüfungsfähig fühlen. Mit dem Erhalt der Aufgabenstellung gilt die Klausur als angetreten und wird bewertet.

By attending this written exam, you confirm that you are working on it alone and feel healthy and capable to participate. Once you have received the examination paper, you are considered to have participated in the exam, and it will be graded.

- Use the provided sheets. Do *not* use own paper.
- You are allowed to use a *self prepared*, *single sided DIN-A4 sheet* in the exam. Only *hand-written originals* are allowed, but no copies.
- Do *not* use a red pen.
- Time limit: *90 minutes*
- Turn off your mobile phones!

Grade: _____

Questions:	1	2	3	4	5	6	7	8	9	10	11	**Σ**
Maximum Points:	6	6	10	10	16	7	9	6	5	7	8	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

Question 1)

Points: ..... of 6

1 Point (1) Describe how memory protection works.
The memory is split and running programs are separated from each other.

1 Point (2) Name one singletasking operating system.
MS-DOS, Palm OS

1 Point (3) Name one multitasking operating system.
Linux/UNIX, MacOS X, Server editions of the Windows NT family, MacOS 8x/9x, AmigaOS, Risc OS, OS/2, Windows 3x/95/98, BeOS

1 Point (4) Name one single-user operating system.
MS-DOS, Palm OS

1 Point (5) Name one multi-user operating system.
Linux/UNIX, MacOS X, Server editions of the Windows NT family

1 Point (6) Name one real-time operating system
RIOT OS, VxWorks, QNX, FreeRTOS, RTLinux

Question 2)

Points: ..... of 6

½ Point (1) GNU HURD implements a...

☐ monolithic kernel ☒ microkernel ☐ hybrid kernel

½ Point (2) Linux implements a...

☒ monolithic kernel ☐ microkernel ☐ hybrid kernel

½ Point (3) MacOS X implements a...

☐ monolithic kernel ☐ microkernel ☒ hybrid kernel

½ Point (4) Windows NT4/Vista/XP/7/8/10/11 implements a...

☐ monolithic kernel ☐ microkernel ☒ hybrid kernel

2 Points (5) Name <u>one</u> advantage and <u>one</u> drawback of microkernels.

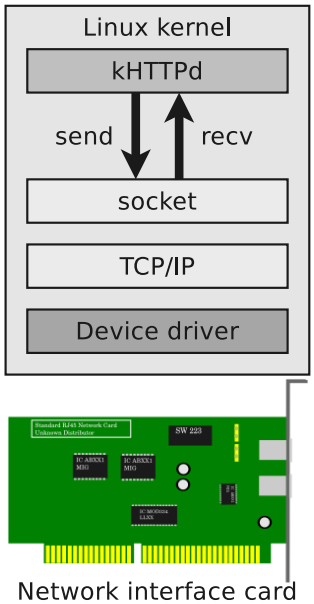
Advantages:

- *Components can be exchanged easily*
- *Best stability and security in theory, because fewer functions run in kernel mode*

Drawbacks:

- *Slower because of more context switches*
- *Development of a new (micro)kernel is a complex task*

2 Points (6) In class we discussed the concept of the kernel-based web server kHTTPd (see image). Explain one benefit and one drawback of this concept.



Advantage: Faster delivery of static(!) web pages. Less switching between user mode and kernel mode is required.

Drawback: Security risk. Complex software like a web server should not run in kernel mode. Bugs in the web server could cause system crashes or enable an attacker to takeover system control.

Question 3)

Points: ..... of 10

4 Points (1) Name and explain one advantage and one drawback of the autonomous subsystems (e.g. Intel Management Engine or AMD Platform Security Processor) in modern PCs.

Advantage: They allow a computer to be monitored and woken up over the network (Wake-on-LAN) and enable remote administration (remote management).

Drawback: These subsystems are not fully documented (quasi-secret). These subsystems always run when sufficient energy is available. They can access all hardware resources, including main memory, I/O interfaces, interfaces and bus systems, and network interfaces. It is an uncontrollable computer within the computer whose exact range of functions is unclear. Such a system is a major potential security risk.

2 Points (2) Describe the purpose of the firmware in the computer.

The firmware performs the Power-On Self-Test (POST). Among other things, this checks the correct functioning of the processor, the buffer memory (cache), and the main memory. After the computer has started and the self-test has been completed, the firmware searches for the first boot device (boot drive) and starts the boot loader.

1 Point (3) Give the name of the firmware in classic computers from the early 1980s to the late 2000s.

BIOS (Basic Input/Output System)

1 Point (4) Give the name of the firmware in modern computers.

UEFI (Unified Extensible Firmware Interface)

1 Point (5) Explain what the boot loader is.

The boot loader is a program that loads the operating system kernel into the main memory when the operating system is started. It also loads the initial RAM disk (`initrd`) or the initial RAM file system (`initramfs`).

1 Point (6) Explain where the boot loader is stored.

When using a classic PC partition table, the boot loader is stored in the 512-byte large master boot record (MBR) at the very beginning of the drive. When using a GUID partition table (GPT), the boot loader is stored in the ESP (EFI System Partition).

Question 4)

Points: ..... of 10

2 Points (1) Explain why an initial RAM disk (`initrd`) or than initial RAM file system (`initramfs`) are used.

The temporary root file system loaded by `initrd` or `initramfs` implements a minimum Linux environment in the main memory. Its primary purpose is to provide the kernel with additional device drivers, file system drivers, and programs to load the operating system's real root file system into memory.

1 Point (2) Describe the task of a `getty` process.

A `getty` process allows text-based user login via a (virtual) console.

1 Point (3) Specify how many `getty` processes the operating system starts.

The operating system starts a separate instance of the `getty` process for each virtual console (TTY1 to TTY6).

1½ Points (4) Name the three sorts of process context information the operating system stores.

User context, hardware context and system context.

½ Point (5) Explain why the process control block (PCB) does not store all process context information.

Depending on the architecture, the virtual memory of each process may be several GB in size. Therefore, the user context is just too big in size to store it twice.

1 Point (6) Explain the task of the dispatcher.

It carries out the state transitions of the processes.

1 Point (7) Explain the task of the scheduler.

It specifies the execution order of the processes.

1 Point (8) Name one drawback of preemptive scheduling.

Higher overhead compared with non-preemptive scheduling because of the frequent process switches.

1 Point (9) Name one drawback of non-preemptive scheduling.

A process may occupy the CPU for as long as it wants and other (maybe more important) processes need to wait.

Question 5)

Points: ..... of 16

1½ Points (1) Name the three main components the CPU contains.
Arithmetic logic unit, control unit, memory.

1½ Points (2) Name the three digital bus systems each computer system contains according to the Von Neumann architecture.
Control bus, address bus, data bus.

3 Points (3) Explain the tasks that are carried out by the three bus systems of subtask (2).
Control bus. Transmits commands (e.g. read and write requests) from the CPU and returns status signals from the I/O devices
Address bus: Transmits memory addresses.
Data bus: Transmits data between CPU, main memory and I/O devices.

2 Points (4) Name the two groups of Input/Output devices for computer systems that are distinguished according to their minimum transfer unit.
Character devices and block devices.

2 Points (5) Describe the different operating principles of the two groups of subtask (4).
Character devices: On arrival/request of each single character, communication with the CPU always takes place.
Block devices: Data transfer takes place only when an entire block (z.B. 1-4 kB) exists.

2 Points (6) Name two examples for each group from subtask (4).
Character devices: Mouse, keyboard, printer, terminal, magnetic tape...
Block devices: HDD, SSD, CD/DVD drive, floppy drive...

1 Point (7) Mark the concept where the CPU must check periodically whether data is available
☐ Direct Memory Access ☐ Interrupt driven ☒ Busy waiting

1 Point (8) Mark the concept where reading data causes no CPU workload
☒ Direct Memory Access ☒ Interrupt driven ☐ Busy waiting

1 Point (9) Name the cache write policy that uses so called dirty bits.
Write-Back.

1 Point (10) Explain for what reason dirty bits are used.
For each page inside the cache, a dirty bit specifies whether the page was modified.

Question 6)

Points: ..... of 7

1 Point (1) Explain why it is wrong to call SSDs Solid State Disks.

They do not contain moving parts. Consequently, they do not include disks.

1 Point (2) Name two advantages of SSDs over HDDs.

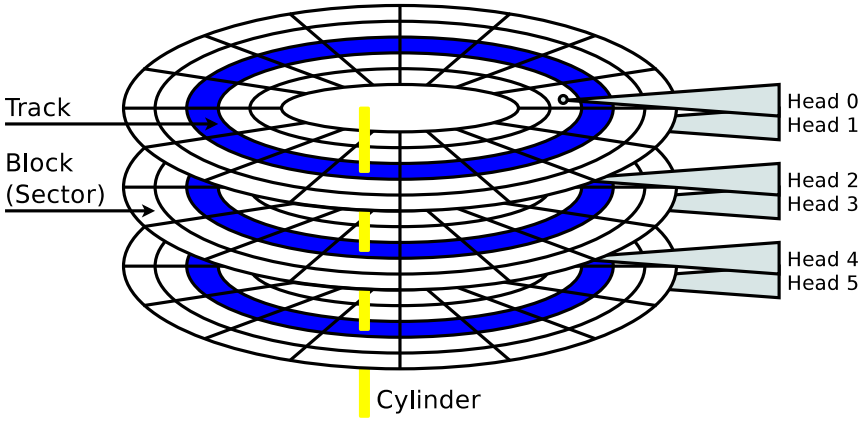
Fast access time, low power consumption, no noise generation, mechanical robustness, low weight, the location of data does not matter $\Longrightarrow$ defragmenting makes no sense.

1 Point (3) Explain why erase operations on flash memory are more complex than read operations.

The memory cells are arranged in groups of pages and blocks. Depending on the structure of a flash memory, a block always contains a fixed number of pages. Write and erase operations can only be carried out for complete pages or blocks.

4 Points (4) Draw the structure of a hard disk drive schematically. Explain with your drawing(s) the meaning of the following terms:

- Sector (= Block)
- Track
- Cylinder
- Cluster



Question 7)

Points: ..... of 9

1 Point (1) Describe the information inodes store.
An inode stores a file's metadata, except the file name.

1 Point (2) Name two examples of metadata in the file system.
Metadata are among others the size, UID/GID, permissions and date.

1 Point (3) Describe what a cluster in the file system is.
File systems address clusters and not blocks of the storage device. Each file occupies an integer number of clusters.

1 Point (4) Describe how directories in the Linux file systems are technically implemented.
Directories are just text files, which contain the names and inodes of files.

1 Point (5) Explain why moving a large file within a file system is always faster than copying it.
Directories are just text files containing the names and inode numbers of files. Moving a file within a file system implies removing a line in the old directory and adding a line in the new directory. On the other hand, copying a file creates a new inode and duplicates the file contents.

½ Point (6) `Documents/MasterThesis/thesis.tex` is an/a...
☐ absolute path name ☒ relative path name

½ Point (7) `/home/<username>/Mail/inbox/` is an/a...
☒ absolute path name ☐ relative path name

1 Point (8) Describe what the File Allocation Table (FAT) is and the information it stores.
The FAT (File Allocation Table) is a table of fixed size. For each cluster in the file system, an entry exists in the FAT with the following information about the cluster:

- *Cluster is free or the storage medium is damaged at this point.*
- *Cluster is occupied by a file and stores the address of the next cluster, which belongs to the file or it is the last cluster of the file.*

1 Point (9) Describe the objective of the journal in a journaling file system.
In the journal, write operations are collected before being committed to the file system.

1 Point (10) Describe a benefit of using a journaling file system compared with using a file system without a journal.
After a crash, only the files (clusters) and metadata must be checked, for which a record exists in the journal.

Question 8)

Points: ..... of 6

1½ Points (1) Name the three values that are required to store an extent.
Start (cluster number) of the area (extent) in the file.
Size of the area in the file (in clusters).
Number of the first cluster on the storage device.

1 Point (2) Describe the benefit of using extents compared with direct addressing of the clusters.
Instead of multiple individual clusters numbers, only 3 values are required: Lesser overhead.

½ Point (3) Name one Linux file system that implements block addressing.
Minix, ext2, ext3

½ Point (4) Name one Linux file system that implements journaling.
ext3, ext4, ReiserFS, XFS, JFS

½ Point (5) Name one Linux file system that implements extents.
ext4, JFS, XFS, btrfs

½ Point (6) Name one Windows file system that implements the file allocation table.
FAT12, FAT16, FAT32, exFAT

½ Point (7) Name one Windows file system that implements journaling.
NTFS

½ Point (8) Name one Windows file system that implements extents.
NTFS

½ Point (9) Name one file system that implements copy-on-write.
ZFS, btrfs, ReFS

Question 9)

Points: ..... of 5

5 Points (1) Perform the deadlock detection with matrices and check if a deadlock occurs.

$$\text{Existing resource vector} = \begin{pmatrix} 9 & 6 & 8 & 7 & 6 & 7 \end{pmatrix}$$

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

The existing resource vector and the current allocation matrix are used to calculate the available resource vector.

$$\textit{Available resource vector} = \begin{pmatrix} 1 & 1 & 2 & 2 & 3 & 2 \end{pmatrix}$$

Only process 1 can run with this available resource vector. The following available resource vector results when process 1 has finished execution and deallocates its resources.

$$\textit{Available resource vector} = \begin{pmatrix} 3 & 1 & 4 & 5 & 5 & 2 \end{pmatrix}$$

Only process 3 can run with this available resource vector. The following available resource vector results when process 3 has finished execution and deallocates its resources.

$$\textit{Available resource vector} = \begin{pmatrix} 4 & 4 & 6 & 6 & 5 & 3 \end{pmatrix}$$

Only process 4 can run with this available resource vector. The following available resource vector results when process 4 has finished execution and deallocates its resources.

$$\textit{Available resource vector} = \begin{pmatrix} 7 & 5 & 6 & 7 & 6 & 4 \end{pmatrix}$$

Process 2 is not blocked.
No deadlock occurs.

Question 10)

Points: ..... of 7

7 Points

(1) Develop a pub simulation software. Glasses are filled by a bartender, and a guest consumes their content.

- The number of available glasses is limited. The bar has only 20 glasses.
- Process `bartender` fills glasses and places them on the bar.
- Process `guest` removes glasses from the bar and consumes their content.
- Mutual exclusion when accessing shared resources (taking a glass) is necessary to avoid inconsistencies.
- If all glasses are filled, the process `bartender` must be blocked.
- If all glasses are empty, the process `guest` must be blocked.

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

```
typedef int semaphore;
semaphore emptyglass = 20;
semaphore fillglass = 0;
semaphore mutex  = 1;


void bartender (void) {

  while (TRUE) {

    P(emptyglass);
    P(mutex);

    fillGlass;
    placeGlassOnBar;

    V(mutex);
    V(fullglass);

  }
}

void guest (void) {

  while (TRUE) {

    P(fullglass);
    P(mutex);

    removeGlassFromBar;
    emptyGlass;

    V(mutex);
    V(emptyglass);

  }
}
```

Question 11)

Points: ..... of 8

The output of the `ps` command contains helpful information about the processes in the operating system.

```
$ ps -eFw
UID      PID   PPID  C      SZ    RSS PSR STIME TTY        TIME CMD
root       1      0  0   42090  12820   0 Aug29 ?      00:00:03 /sbin/initroot
root       2      0  0       0      0   4 Aug29 ?      00:00:00 [kthreadd]
...
bnc     2149   1782  1  258958 133484   7 Aug29 ?      00:11:20 xfwm4 --display :0.0  ...
bnc     2474   1782  0  137013  54512   8 Aug29 ?      00:03:28 xfce4-panel --display :0.0 ...
bnc     2478   1782  0  166034 138652  15 Aug29 ?      00:00:20 xfdesktop --display :0.0 ...
bnc     3252   2474  3 8590107 577484   9 Aug29 ?      00:51:07 /opt/google/chrome/chrome
bnc     3530   1721  0  157125  62824   0 Aug29 ?      00:00:44 /usr/libexec/gnome-terminal-server
bnc     3568   3530  0    3271   9556  15 Aug29 pts/0  00:00:01 bash
root    6706      1  0    7087  10556   3 Aug29 ?      00:00:00 /usr/sbin/cupsd -l
root    6737      1  0   44549  18680  12 Aug30 ?      00:00:00 /usr/sbin/cups-browsed
bnc    72577  72539  0    2773   7224   4 Aug31 pts/1  00:00:00 /bin/bash
bnc    90775  72577  1  279130 187352   9 09:39 pts/1  00:00:04 okular thesis.pdf
bnc    94414   3568  0    2861   4952   6 11:19 pts/0  00:00:00 ps -eFw
```

1 Point (1) Explain the information in the column `UID`.

User ID of the owner of the process.

1 Point (2) Explain the information in the column `PID`.

The unique process ID.

1 Point (3) Explain the information in the column `PPID`.

The unique process ID of the parent process.

1 Point (4) Explain the information in the column `SZ`.

virtual process size = text segment, heap and stack.

1 Point (5) Explain the information in the column `RSS`.

Resident Set Size = occupied physical memory (without swap) in kB.

1 Point (6) Explain the information in the column `TTY`.

Teletypewriter = control terminal. Usually a virtual device: pts (pseudo terminal slave)

1 Point (7) Explain the information in the column `TIME`.

Consumed CPU time of the process (HH:MM:SS).

1 Point (8) Name the parent process of the process that has printed this overview of the processes in the command-line interface.

The `bash` process with PID 3568 is the parent process of the `ps` process with PID 94414.