Solution of Exercise Sheet 3

Exercise 1 (Computer Architecture)

1. Name the three components the CPU contains.

   Arithmetic logic unit, control unit, memory.

2. Name the three digital bus systems each computer system contains according to the Von Neumann architecture.

   Control bus, address bus, data bus.

3. Name the tasks that are carried out by the three digital 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.

4. Explain what the Front Side Bus (FSB) is.

   It is the bus between CPU and chipset. It contains the address bus, data bus and control bus.

5. Give the names of the two components the chipset contains. *Northbridge und Southbridge.*

6. Name the tasks of the components of the chipset.

   The Northbridge is used for the connection of main memory and graphics card(s) with the CPU. It is located close to the CPU for rapid data transfer. The Southbridge is used for „slow“ connections like Ethernet, SATA and USB.

Exercise 2 (Input/Output Devices)

1. Give the names of the two groups of Input/Output devices for computer systems that can be distinguished according to their minimum transfer unit.

   Character devices and block devices.

2. Describe the different operating principles of the groups of subtask 1.

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.

3. Name two examples for each group from subtask 1.

   Character devices: Mouse, keyboard, printer, terminal, magnetic tape...

   Block devices: HDD, SSD, CD/DVD drive, floppy drive...

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

   - *Busy waiting*
   - *Interrupt-driven*
   - *Direct Memory Access (DMA)*

5. Name a benefit and a drawback for each possible way from subtask 4.

   - *Busy waiting*
     - *Benefits: No additional hardware required*
     - *Drawbacks: Causes CPU workload, slows down simultaneous processing of multiple processes*
   - *Interrupt-driven*
     - *Benefits: The CPU is not blocked, allows the simultaneous execution of multiple processes*
     - *Drawbacks: Additional hardware (interrupt controller) is required*
   - *Direct Memory Access (DMA)*
     - *Benefits: Reading data causes no CPU workload, simultaneous execution of multiple processes is not slowed down*
     - *Drawbacks: Additional hardware (DMA controller) is required*

Exercise 3 (Digital Data Storage)

1. Name one mechanic digital data storage.

   Punched tape, Punch card, mass-production of CDs/DVDs (pressing).

2. Name two rotating magnetic digital data storages.

   Hard Disk Drive, Drum memory, Floppy Disk.

3. Name two non-rotating magnetic digital data storages.

   Magnetic-core memory, Magnetic tape, Magnetic stripe card, Compact cassette (Datasette), Bubble memory.

4. Name four benefits of data storage without moving parts compared with data storage with moving parts.

   Lower power consumption, lesser waste heat, mechanical robustness, no noise generation.

5. Explain what random access is.

   The media does not need to be searched sequentially from the start – such as with magnetic tapes – to locate a specific location (file).

6. Name one non-persistent data storage.

   Main memory (DRAM).

7. The storage of computer systems is distinguished into the categories primary storage, secondary storage and tertiary storage. Give the name of the category or categories the CPU can access directly.

   Primary storage.

8. Give the name of the category or categories of subtask 7 the CPU can only access via a controller.

   Secondary storage and tertiary storage.

9. Name two examples for each category of subtask 7.

   Primary storage: Register, Cache, Main memory.

   Secondary storage: HDD, SSD, CF/SD flash storage card.

   Tertiary storage: CD/DVD drive, magnetic tape.

10. Name the two categories of tertiary storage.

    Near-line storage and off-line storage.

11. Describe the two categories of subtask 10.

    Near-line storage: Is automatically and without human intervention connected to the system (e.g. tape library).

    Off-line storage: Media are stored in cabinets or storage rooms and must be connected manually to the system.

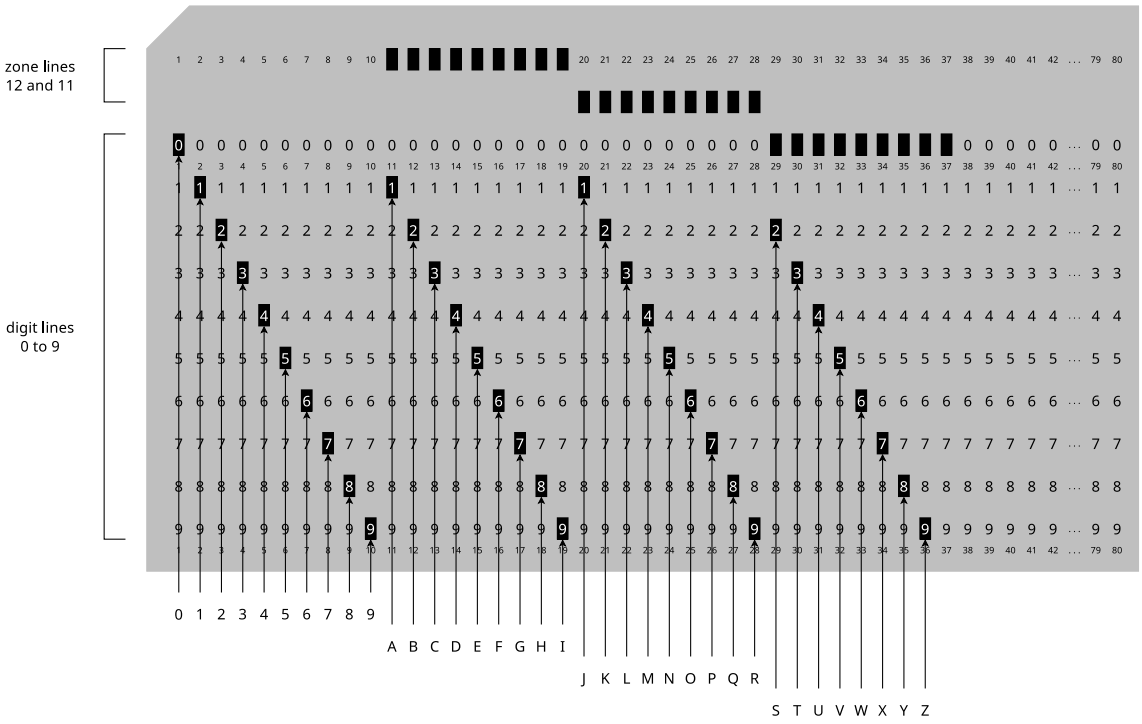
Exercise 4 (Encoding of data and data storage)

An exciting example of how digital data can be persistently stored is punched cards, which were used until the 1970s. Each punch card usually represents one line of

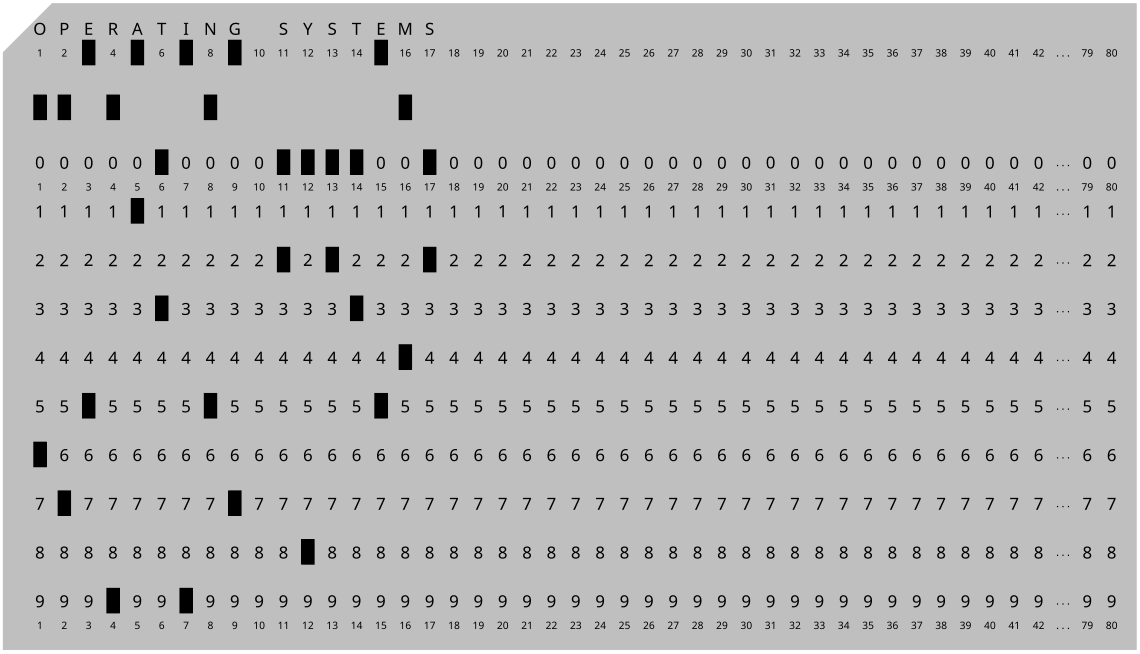
program code with 80 characters. Each column corresponds to one character. The numbers and letters are encoded with the digit lines 0 to 9 and with the so-called zone lines 11 and 12. The digit line 0 is sometimes called zone line 10 in the literature. The digits 0 to 9 are encoded directly in the corresponding lines. The coding of the letters, on the other hand, is as follows:

- A to I encoded by punching the zone line 12 and the digit line 1 to 9.
- J to R encoded by punching the zone line 11 and the digit line 1 to 9.
- S to Z encoded in digit line 0 (= zone line 10) and in zone lines 2 to 9.

Encoding lowercase letters, umlauts, negative numbers, and special characters is possible by various other hole combinations but is irrelevant here. The following figure shows the principle of encoding the digits 0 to 9 and the letters.



Record the character string `OPERATING SYSTEMS` on the punch card.



Note: There are punch card simulators such as `https://www.masswerk.at/keypunch`

Exercise 5 (Write policies)

1. Name the two basic cache write policies.

   Write-Through and Write-Back.

2. Name the cache write policy that may cause inconsistencies.

   Write-back.

3. Name the cache write policy that causes a lower system performance.

   Write-through.

4. Name the cache write policy that uses so called dirty bits.

   Write-Back.

5. Explain for what reason dirty bits are used.

   For each page inside the cache, a dirty bit specifies whether the page was modified.

Exercise 6 (Permissions and Links)

1. Which command can be used to specify that all new created files have this permissions: `-r--r--r--`

   ```
   $ umask 222
   ```

   Attention! If you executed the command, you should fix your permissions as a next step. Otherwise it will be not so comfortable for you to work in your home directory.

2. Create in your home directory a directory with the name `BTS_Links`. Navigate to the new directory and try to erase the entry „.“.

   ```
   $ mkdir ~/BTS_Links && cd ~/BTS_Links
   $ rm .
   ```

3. Create in the directory `BTS_Links`...

   - an empty file `Original`.
     `$ touch Original`
   - a hard link `HardLink`, which points to the file `Original`.
     `$ ln Original HardLink`
   - a symbolic link `SymbLink`, which points to the file `Original`.
     `$ ln -s Original SymbLink`

4. Check the inodes of the file `Original` and of both links via `ls -li`.

   ```
   $ ls -li Original SymbLink
   ```

5. Is it possible to copy hard links? Try to copy the link.

   It is possible to copy hard links inside a file system, but it is impossible to copy a hard link to another file system or partition because hard links are just file system entries that refer to an inode.

6. Is it possible to copy symbolic links? Try to copy the link.

   it is possible to copy symbolic links because they are just (text)files that contain a path information.

7. Check the result of your copying via `ls -li`. What are your conclusions?

   If a hard link is copied, a further directory entry is created that refers to an inode.

   If a symbolic link is copied, an new file (with a new inode number) is created.

8. The so called *link count* of files indicates the number of directory entries, which refer to an inode. What indicates the link count of directories and what influences the link count of directories?

   Each directory has two hard links from the moment when it is created. One hard link is the name of the directory itself, and the other hard link is the dot „.“, which refers to the directory one level above in the file system hierarchy. Therefore, the hard link counter of each directory has value 2 from the moment when it is created. For each additional subdirectory, another hard link is created, and the hard link counter is incremented by value 1. Thus, the number of subdirectories inside a directory is equal to the hard link counter -2.

Exercise 7 (Wildcards and Filters)

1. Create in your home directory a directory `DiverseDateien`. Navigate to this directory and create these files:

```
abcdefg.bat  cdata3.sav  cdata7.sav   datei3.txt  datei7.txt
abcxyz.bat   cdata4.sav  datei10.txt  datei4.txt  datei8.txt
cdata1.sav   cdata5.sav  datei1.txt   datei5.txt  datei9.txt
cdata2.sav   cdata6.sav  datei2.txt   datei6.txt  xyzabc.bat
```

```
$ mkdir ~/DiverseDateien && cd ~/DiverseDateien
$ touch abcdefg.bat
$ touch abcxyz.bat
$ for ((i=1; i < 8; i++)) ; do touch cdata$i.sav ; done
$ for ((i=1; i < 11; i++)) ; do touch datei$i.txt ; done
$ touch xyzabc.bat
```

2. Which command can be used to print out a list of all files in the directory, whose filenames start with the pattern `datei`? `$ ls -1 | grep -e "^datei"`

   Alternative solution:

   ```
   $ ls datei*
   ```

3. Which command can be used to print out a list of all files in the directory, whose filenames contain the pattern `cd`?

   ```
   $ ls -1 | grep -e "cd"
   ```

   Alternative solution:

   ```
   $ ls *cd*
   ```

4. Which command can be used to print out a list of the files `cdata2.sav`, ..., `cdata5.sav` in the directory?

   ```
   $ ls -1 | grep -e "cdata[2-5].sav"
   ```

Alternative solution:

```
$ ls cdata[2-5].sav
```

5. Which command can be used to print out a list of all files in the directory, whose filenames contain the characters `c` or `z` on position 3?

   ```
   $ ls -1 | grep -e "^.\{2\}[cz]"
   ```

   Alternative solution:

   ```
   $ ls ??[cz]*
   ```

6. Which command can be used to print out a list of all files in the directory, whose filenames start with the character `a` and end with the character `t` and also contain the character `c` on any position?

   ```
   $ ls -1 | grep -e "^a.*c.*t$"
   ```

   Alternative solution:

   ```
   $ ls [a]*c*t
   ```

7. Which command can be used to print out a list of the files `datei1.txt`, ..., `datei9.txt` in the directory but without the files `datei3.txt` and `datei4.txt`.

   ```
   $ ls datei[!34][!0]*
   ```

   Alternative solution:

   ```
   $ ls -1 | grep -e "datei[^3-4]\.txt"
   ```