

## Lösung von Übungsblatt 3

### Aufgabe 1 (Rechnerarchitektur)

1. Nennen Sie die drei Komponenten des Hauptprozessors.

*Rechenwerk, Steuerwerk und Speicher.*

2. Nennen Sie die drei digitalen Busse eines Rechnersystems nach der Von-Neumann-Architektur.

*Steuerbus, Adressbus und Datenbus.*

3. Nennen Sie die Aufgaben, der drei digitalen Busse aus Teilaufgabe 2.

*Steuerbus: Überträgt Kommandos (z.B. Lese- und Schreibweisungen) von der CPU und Statusmeldungen von den Peripheriegeräten. Adressbus: Überträgt Speicheradressen. Datenbus: Überträgt Daten zwischen CPU, Arbeitsspeicher und Peripherie.*

4. Beschreiben Sie was der Systembus oder Front Side Bus (FSB) ist.

*Es ist der Bus zwischen CPU und Chipsatz. Er enthält den Adressbus, Datenbus und Steuerbus.*

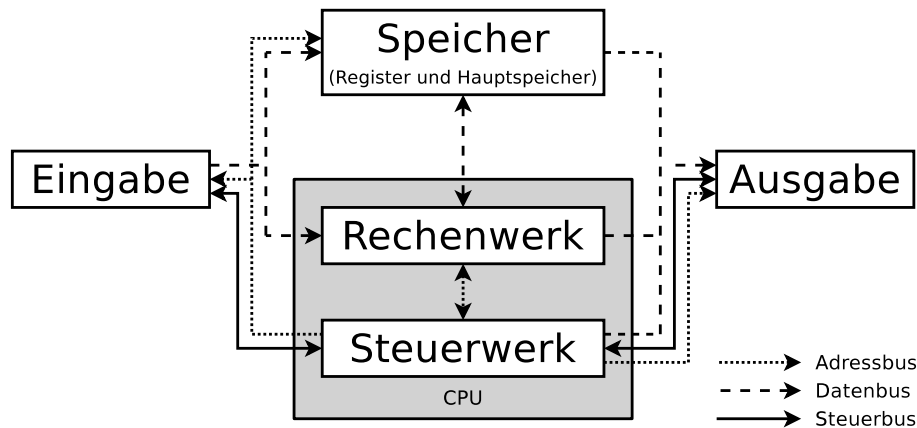
5. Nennen Sie die beiden Komponenten des Chipsatzes.

*Der Chipsatz besteht aus Northbridge und Southbridge.*

6. Geben Sie für jede Komponente des Chipsatzes an, welche Aufgabe sie hat.

*Die Northbridge ist Zuständig für die Anbindung des Hauptspeichers und der Grafikkarte(n) an die CPU. Sie liegt dicht an der CPU, um Daten schnell übertragen zu können. Die Southbridge ist für „langsamere“ Verbindungen wie zum Beispiel Ethernet, SATA und USB zuständig.*

7. The diagram shows the Von Neumann Architecture. Fill into the boxes the missing components.



8. Explain briefly how the Von Neumann Cycle works (explain the six steps).
- FETCH:** Abzuarbeitenden Befehl aus dem Speicher in das Befehlsregister (Instruction Register) kopieren
  - DECODE:** Steuerwerk löst den Befehl in Schaltinstruktionen für das Rechenwerk auf
  - FETCH OPERANDS:** Eventuell verfügbare Parameter (Operanden) für den Befehl aus dem Speicher holen
  - EXECUTE:** Rechenwerk führt den Befehl aus
  - UPDATE PROGRAM COUNTER:** Befehlszähler (Program Counter) wird auf den nächsten Befehl gesetzt
  - WRITE BACK:** Das Ergebnis des Befehls wird in einem Register oder im Hauptspeicher gespeichert oder zu einem Ausgabegerät gesendet

## Aufgabe 2 (Ein-/Ausgabegeräte)

1. Nennen Sie die Namen der zwei Gruppen von Ein- und Ausgabegeräten bezüglich der kleinsten Übertragungseinheit.

*Zeichenorientierte Geräte und blockorientierte Geräte.*

2. Vergleichen Sie die Arbeitsweise der Gruppen aus Teilaufgabe 1.

*Zeichenorientierte Geräte: Bei Ankunft/Anforderung jedes einzelnes Zeichens wird immer mit der CPU kommuniziert.*

*Blockorientierte Geräte: Datenübertragung findet erst statt, wenn ein kompletter Block (z.B. 1-4 kB) vorliegt.*

3. Nennen Sie für jede Gruppe aus Teilaufgabe 1 zwei Beispiele.

*Zeichenorientierte Geräte: Maus, Tastatur, Drucker, Terminal, Magnetband. . .*

*Blockorientierte Geräte: Festplatte, SSD, CD-/DVD-Laufwerk, Disketten-Laufwerk. . .*

4. Nennen Sie drei Möglichkeiten, wie Prozesse Daten von Ein- und Ausgabege-  
räten lesen können.
  - *Busy Waiting (geschäftiges bzw. aktives Warten)*
  - *Interrupt-gesteuert*
  - *Direct Memory Access (DMA)*
  
5. Nennen Sie für jede Möglichkeit aus Teilaufgabe 4 jeweils einen Vorteil und  
einen Nachteil.
  - *Busy Waiting (geschäftiges bzw. aktives Warten)*
    - *Vorteile:*
      - \* *Keine zusätzliche Hardware nötig*
    - *Nachteile:*
      - \* *Belastet die CPU*
      - \* *Verlangsamt die gleichzeitige Abarbeitung mehrerer Prozesse*
  - *Interrupt-gesteuert*
    - *Vorteile:*
      - \* *Die CPU wird nicht blockiert*
      - \* *Gleichzeitige Abarbeitung mehrerer Prozesse wird nicht verlang-*  
*samt*
    - *Nachteile:*
      - \* *Zusätzliche Hardware (Interrupt-Controller) ist nötig*
  - *Direct Memory Access (DMA)*
    - *Vorteile:*
      - \* *Vollständige Entlastung der CPU*
      - \* *Gleichzeitige Abarbeitung mehrerer Prozesse wird nicht verlang-*  
*samt*
    - *Nachteile:*
      - \* *Zusätzliche Hardware (DMA-Controller) ist nötig*

## **Aufgabe 3 (Digitale Datenspeicher)**

1. Nennen Sie einen digitalen Datenspeicher, der mechanisch arbeitet.  
*Lochstreifen, Lochkarte, CD/DVD beim Pressen.*
2. Nennen Sie zwei rotierende magnetische digitale Datenspeicher.  
*Festplatte, Trommelspeicher, Diskette.*
3. Nennen Sie zwei nichtrotierende magnetische digitale Datenspeicher.

*Kernspeicher, Magnetband, Magnetstreifen, Magnetkarte, Compact Cassette (Datasette), Magnetblasenspeicher.*

4. Nennen Sie vier Vorteile von Datenspeicher ohne bewegliche Teile gegenüber Datenspeichern mit beweglichen Teilen.

*Weniger Energieverbrauch, weniger Abnutzung, weniger Abwärme, unempfindlichkeit gegen Stöße, keine Laufgeräusche.*

5. Beschreiben Sie was wahlfreier Zugriff ist.

*Wahlfreier Zugriff heißt, dass das Medium nicht - wie z.B. bei Bandlaufwerken - von Beginn an sequentiell durchsucht werden muss, um eine bestimmte Stelle (Datei) zu finden.*

6. Nennen Sie einen nicht-persistenten Datenspeicher.

*Hauptspeicher (DRAM).*

7. Der Speicher eines Computersystems wird in die Kategorien Primärspeicher, Sekundärspeicher und Tertiärspeicher unterschieden. Nennen Sie die Kategorie(n), auf die der Prozessor direkt zugreifen kann.

*Nur auf den Primärspeicher.*

8. Nennen Sie die Kategorie(n) aus Teilaufgabe 7, auf die der Prozessor nur über einen Controller zugreifen kann.

*Auf den Sekundärspeicher und den Tertiärspeicher.*

9. Nennen Sie für jede Kategorie aus Teilaufgabe 7 zwei Beispiele.

*Primärspeicher: Register, Cache, Hauptspeicher.*

*Sekundärspeicher: Festplatte, SSD, CF-Karte.*

*Tertiärspeicher: CD/DVD-Laufwerk, MO-Laufwerk, Magnetband.*

10. Nennen Sie die beiden Kategorien, in die der Tertiärspeicher unterschieden wird.

*Nearlinespeicher und Offlinespeicher.*

11. Beschreiben Sie die beiden Kategorien aus Teilaufgabe 10.

*Nearlinespeicher: Werden automatisch und ohne menschliches Zutun dem System bereitgestellt (z.B. Band-Library).*

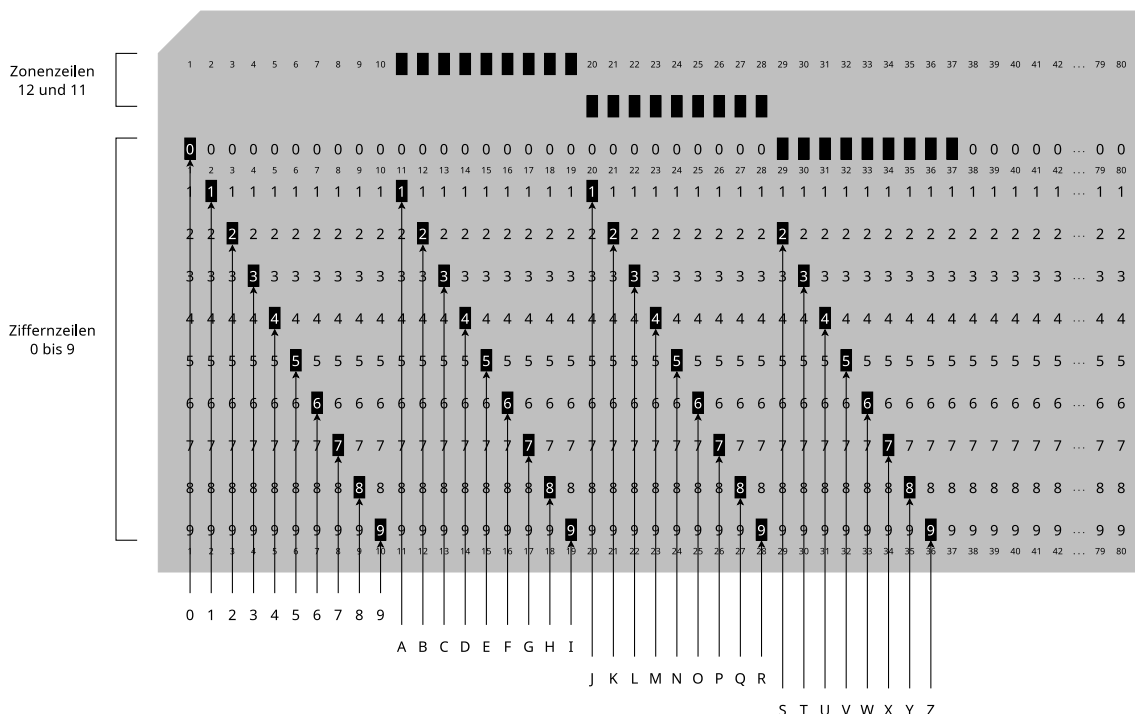
*Offlinespeicher: Medien werden in Schränken oder Lagerräumen aufbewahrt und müssen von Hand in das System integriert werden.*

## Aufgabe 4 (Kodierung von Daten und Datenspeicher)

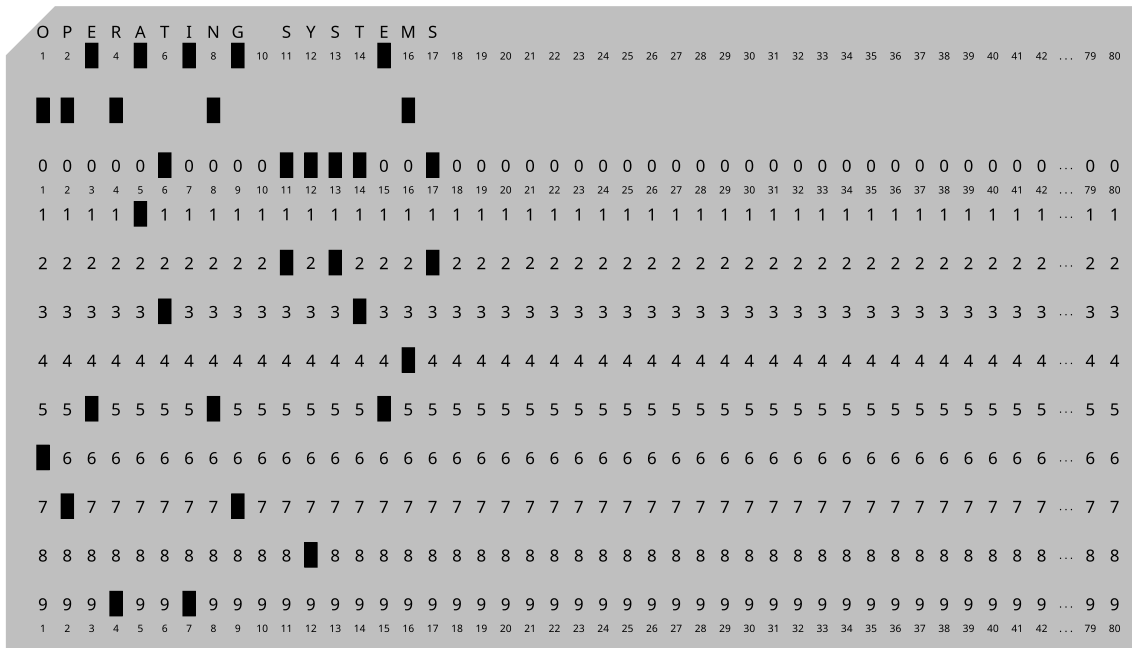
Ein interessantes Beispiel für die vielfältigen Möglichkeiten, digitale Daten persistent zu speichern, sind Lochkarten, die bis in die 1970er Jahre verwendet wurden. Jede Lochkarte stellt üblicherweise eine Zeile Programmtext mit 80 Zeichen dar. Jede Spalte entspricht einem Zeichen. Die Kodierung der Zahlen und Buchstaben erfolgt mit den Ziffernzeilen 0 bis 9 und mit den sogenannten Zonenzeilen 11 und 12. In der Literatur heißt die Ziffernzeilen 0 auch manchmal Zonenzeile 10. Die Ziffern 0 bis 9 werden direkt in den entsprechenden Zeilen kodiert. Die Kodierung der Buchstaben ist hingegen wie folgt:

- A bis I erfolgt durch Lochung der Zonenzeile 12 und der Ziffernzeile 1 bis 9
- J bis R erfolgt durch Lochung der Zonenzeile 11 und der Ziffernzeile 1 bis 9
- S bis Z erfolgt durch Lochung der Ziffernzeile 0 (= Zonenzeile 10) und der Ziffernzeile 2 bis 9

Die Kodierung von Kleinbuchstaben, Umlauten, negativen Zahlen und Sonderzeichen ist durch verschiedene weiteren Lochkombinationen möglich, spielt hier aber keine Rolle. Die folgende Abbildung zeigt das Prinzip der Kodierung der Ziffern 0 bis 9 und der Buchstaben.



Speichern Sie auf der Lochkarte die Zeichenkette **OPERATING SYSTEMS**



Hinweis: Es Lochkarten-Simulatoren wie z.B. <https://www.masswerk.at/keypunch>

## Aufgabe 5 (Cache-Schreibstrategien)

1. Nennen Sie die beiden grundsätzlichen Cache-Schreibstrategien.

*Write-Through und Write-Back.*

2. Nennen Sie die Cache-Schreibstrategie bei der es Inkonsistenzen kommen kann.

*Write-Back.*

3. Nennen Sie die Cache-Schreibstrategie bei der die System-Geschwindigkeit geringer ist.

*Write-Through.*

4. Nennen Sie die Cache-Schreibstrategie bei der sogenannte „Dirty Bits“ zum Einsatz kommen.

*Write-Back.*

5. Beschreiben Sie die Aufgabe der „Dirty Bits“.

*Für jede Seite im Cache wird ein Dirty Bit im Cache gespeichert, das angibt, ob die Seite geändert wurde.*

## Aufgabe 6 (Zugriffsrechte und Links)

1. Sie haben versucht `script.sh` auszuführen und es kam zu folgender Ausgabe:

```
$ ./script.sh
permission denied: ./script.sh
```

Nennen Sie einen Lösungsweg für die Eingabeaufforderung, der die Ausführung von `script.sh` ermöglicht.

*Einige Beispiele (jedes davon führt zum Ziel)...*

```
$ chmod 777 script.sh
$ chmod 700 script.sh
$ chmod 500 script.sh
$ chmod u+rwx script.sh
$ chmod u+rx script.sh
```

2. Geben Sie das Kommando an, mit dem Sie festlegen können, dass alle neu erzeugten Dateien die folgenden Rechte erhalten: `-r--r--r--`

```
$ umask 222
```

*Achtung! Wenn Sie das Kommando ausgeführt haben, sollten sie es später auch wieder rückgängig machen. Ansonsten wird zukünftiges Arbeiten schwierig.*

3. Erzeugen Sie in Ihrem Home-Verzeichnis ein Verzeichnis mit dem Namen `BTS_Links`. Wechseln Sie in das Verzeichnis und versuchen Sie `.` zu löschen.

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

*Ergebnis: Das Entfernen von „.“ ist nicht möglich.*

4. Erzeugen Sie im Verzeichnis `BTS_Links`...

- eine leere Datei `Original`.  
`$ touch Original`
- einen Hard Link `HardLink` auf die Datei `Original`.  
`$ ln Original HardLink`
- einen Symbolischen Link `SymbLink` auf die Datei `Original`.  
`$ ln -s Original SymbLink`

5. Kontrollieren Sie die Inodes der Datei `Original` und der beiden Links mit `ls -li`.

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

6. Können Hard Links kopiert werden? Versuchen Sie den Link zu kopieren.

*Ja, aber nicht außerhalb des Dateisystems bzw. der Partition, weil Hard Links nur Verzeichniseinträge sind, die auf einen Inode verweisen.*

7. Können Symbolische Links kopiert werden? Versuchen Sie den Link zu kopieren.

*Ja, weil symbolische Links (Text-)Dateien mit einer Pfadangabe sind.*

8. Betrachten Sie das Ergebnis des Kopierens mit `ls -li`. Welche Schlüsse ziehen Sie daraus?

*Wird ein Hard Link kopiert, wird nur ein weiterer Verzeichniseintrag angelegt, der auf einen Inode zeigt.*

*Wird ein symbolischer Link kopiert, wird eine neue Datei (mit einer eigenen Inode-Nummer) erzeugt.*

9. Der sog. *Link-Count* bei Dateien gibt an, wie viele Verzeichniseinträge auf einen Inode zeigen. Was gibt der Link-Count bei Verzeichnissen an und wie wird er beeinflusst?

*Jedes Verzeichnis hat bereits beim Anlegen zwei Hard Links, Einmal der Name des Verzeichnisses und der Punkt „.“, der auf das Verzeichnis zeigt. Darum hat direkt nach der Erzeugung jedes Verzeichnis zwei Hard Links-Counts. Für jedes weitere Verzeichnis wird ein neuer HardLink angelegt und hinzugezählt. Es liegen also mindestens zwei Verzeichnisse weniger im Verzeichnis, als der Link-Counter angibt.*

## Aufgabe 7 (Wildcards und Filter)

1. Erstellen Sie in Ihrem Home-Verzeichnis ein Verzeichnis `DiverseDateien`. Wechseln Sie in das Verzeichnis und erzeugen Sie die folgenden Dateien:

```
abcdefg.bat  cdata3.sav  cdata7.sav  datei3.txt  datei7.txt
abcxyz.bat   cdata4.sav  datei10.txt datei4.txt  datei8.txt
cdata1.sav   cdata5.sav  datei11.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.tyt ; done
$ touch xyzabc.bat
```

2. Wie können Sie die Liste aller Dateien im Verzeichnis ausgeben lassen, deren Dateiname mit `datei` beginnt?



```
$ ls -l | grep -e "^datei"
```

*oder alternativ:*

```
$ ls datei*
```

3. Wie können Sie die Liste aller Dateien im Verzeichnis ausgeben lassen, deren Dateiname `cd` enthält?

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

*oder alternativ:*

```
$ ls *cd*
```

4. Wie können Sie die Liste der Dateien `cdata2.sav`, ..., `cdata5.sav` im Verzeichnis ausgeben lassen?

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

*oder alternativ:*

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

5. Wie können Sie die Liste aller Dateien im Verzeichnis ausgeben lassen, die im Dateinamen an dritter Stelle ein `c` oder ein `z` enthalten?

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

*oder alternativ:*

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

6. Wie können Sie die Liste aller Dateien im Verzeichnis ausgeben lassen, deren Dateiname mit `a` anfängt, an beliebiger Stelle ein `c` enthält und mit `t` endet?

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

*oder alternativ:*

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

7. Wie können Sie die Liste der Dateien `datei1.txt`, ..., `datei9.txt`, ausgenommen `datei3.txt` und `datei4.txt` im Verzeichnis ausgeben lassen?

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

*oder alternativ:*

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

*oder alternativ:*

```
$ ls -l | grep -e "datei[1-25-9]\.txt"
```

*oder alternativ:*

```
$ ls -l | grep -e '.*datei[1-9].txt' | grep -e '.*datei[^34].txt'
```