



## 1. Aufgabenblatt mit Lösungsvorschlag

18.04.2023

Die Aufgaben 1 bis 4 können gerne zur Wiederholung des Stoffes bearbeitet werden. Eine kompakte Zusammenstellung finden Sie im Foliensatz Informations- und Zahldarstellung.

### Aufgabe 1: ASCII/EBCDIC-Code

Die folgenden Dezimalzahlen stellen einen im ASCII-Code<sup>1</sup> kodierten Text dar.

65 115 115 101 109 98 108 101 114 32 109 97 99 104 116 32 83 112 97 115 115

- (a) Geben Sie den Text in Klarschrift an.
- (b) Kodieren Sie nun den Text im EBCDIC-Code<sup>2</sup>. Geben Sie die Kodierung in Hexadezimaldarstellung an. Eine EBCDIC-Tabelle lässt sich mit einer Suchmaschine Ihrer Wahl einfach finden.
- (c) Können im ASCII- bzw. EBCDIC-Code Dezimalziffern ausschließlich am oberen Halbbyte erkannt werden?

### Lösungsvorschlag:

- (a) Assembler macht Spass
- (b) Kodierung in Hexadezimaldarstellung: C1 A2 A2 85 94 82 93 85 99 40 94 81 83 88 A3 40 E2 97 81 A2 A2
- (c) ASCII: Nein, z. B. “?” = 3F - EBCDI: Ja, alle F

### Aufgabe 2: Binär- und 2-Komplement-Darstellung von Dezimalzahlen

Gegeben sind die folgenden Dezimalzahlen:  $x = 18$ ,  $y = 77$ ,  $z = -127$ ,  $w = 1$

- (a) Geben Sie die Binärdarstellung der Zahlen in Vorzeichen-Betragsform an.  
(1 Bit Vorzeichen, 7 Bit Betrag)
- (b) Führen Sie folgende Rechnung aus:  $\text{result} = y + z - x - w$
- (c) Wie lautet die 2K-Darstellung der Zahlen  $x$ ,  $y$ ,  $z$ ,  $w$ ? (Darstellung mit 8 Bit)
- (d) Wiederholen Sie die Rechnung aus (b) in 2K-Darstellung. Die Subtraktionen sollen auf Additionen des 2K-Komplements zurückgeführt werden.
- (e) Berechnen Sie folgende Ausdruck jeweils in Vorzeichen-Betrag- und 2K-Darstellung:  
 $\text{result} = z - w$  - Wo gibt es Probleme?

Hinweis: Führen Sie die Operationen nacheinander aus. Berechnen Sie die Zwischenergebnisse.

### Lösungsvorschlag:

(a)	VB	(c)	K2
x	= 0:0010010		= 00010010
y	= 0:1001101		= 01001101
z	= 1:1111111		= 10000001
w	= 0:0000001		= 00000001

<sup>1</sup> American Standard Code for Information Interchange

<sup>2</sup> Extended Binary Coded Decimal Interchange Code

(b)	VB	(d)	K2
y+z	1:1111111	-127	10000001
	0:1001101	+77	01001101
<hr/>			
	1:0110010	-50	11001110
x	1:0010010	-18	11101110
<hr/>			
	1:1000100	-68	11011100
w	1:0000001	-1	11111111
<hr/>			
	1:1000101	-69	11011101

(e)	VB	K2
z-w	1:1111111	-127
	1:0000001	-1
<hr/>		11000000
	11:0000000	-128

In der Vorzeichenbetragsform entsteht ein Überlauf. In der K2-Darstellung entsteht kein Überlauf.

### Aufgabe 3: BCD-Kodierung

Folgende BCD-Kodierung<sup>3</sup> einer Zahl ist gegeben: x = 0001 0100 0010 BCD

- Geben Sie die Zahl in Dezimaldarstellung an.
- Wieviele Bits werden zur Binärdarstellung der Zahl benötigt?
- Führen Sie die folgende Rechnung in BCD-Darstellung im Dualen aus:  
result = x + y (mit y = 687)

### Lösungsvorschlag:

- 142
- 10001110 B  $\Rightarrow$  8 Bit
- Hinweis: Beachten Sie die Pseudotetraden

x+y	0001 0100 0010	BCD
	0110 1000 0111	BCD
<hr/>		
	0111 1100 1001	
+6	0000 0110 0000	
<hr/>		
	1000 0010 1001	BCD

<sup>3</sup> Binary Coded Decimal, dualkodierte Dezimalziffer

## Aufgabe 4: Binär-, Oktal- und Hexadezimaldarstellung von Dezimalzahlen

- (a) Geben Sie die Binär-, Oktal- und Hexadezimaldarstellung der Dezimalzahl  $x = 67312$  an. Berechnen Sie zunächst die Binärzahl. Fassen Sie dann Bitgruppen zusammen und wandeln Sie diese in Hexadezimal- und Oktal- und Dual-Rechnung um.
- (b) Von der Zahl  $x$  soll nun die Zahl  $y = B15D$  H subtrahiert werden. Führen Sie die Rechnung in Hexadezimal-, Oktal- und Dual-Rechnung aus. Konvertieren Sie das Ergebnis in die Dezimaldarstellung.

### Lösungsvorschlag:

- (a) Die Konvertierung in das Hexadezimalsystem lässt sich vereinfachen, indem man die Bitfolge in 4-Bit große Blöcke aufteilt.

Analog dazu, kann die Konvertierung in das Oktalsystem vereinfacht werden, indem die Bitfolge in 3-Bit große Blöcke aufgeteilt wird.

$$\begin{aligned} x &= 1\ 0000\ 0110\ 1111\ 0000\ B &= 10F0\ H \\ &= 10\ 000\ 011\ 011\ 110\ 000\ B &= 203360\ O \end{aligned}$$

- (b) Binär-, Hexadezimal-, Oktal- Darstellung

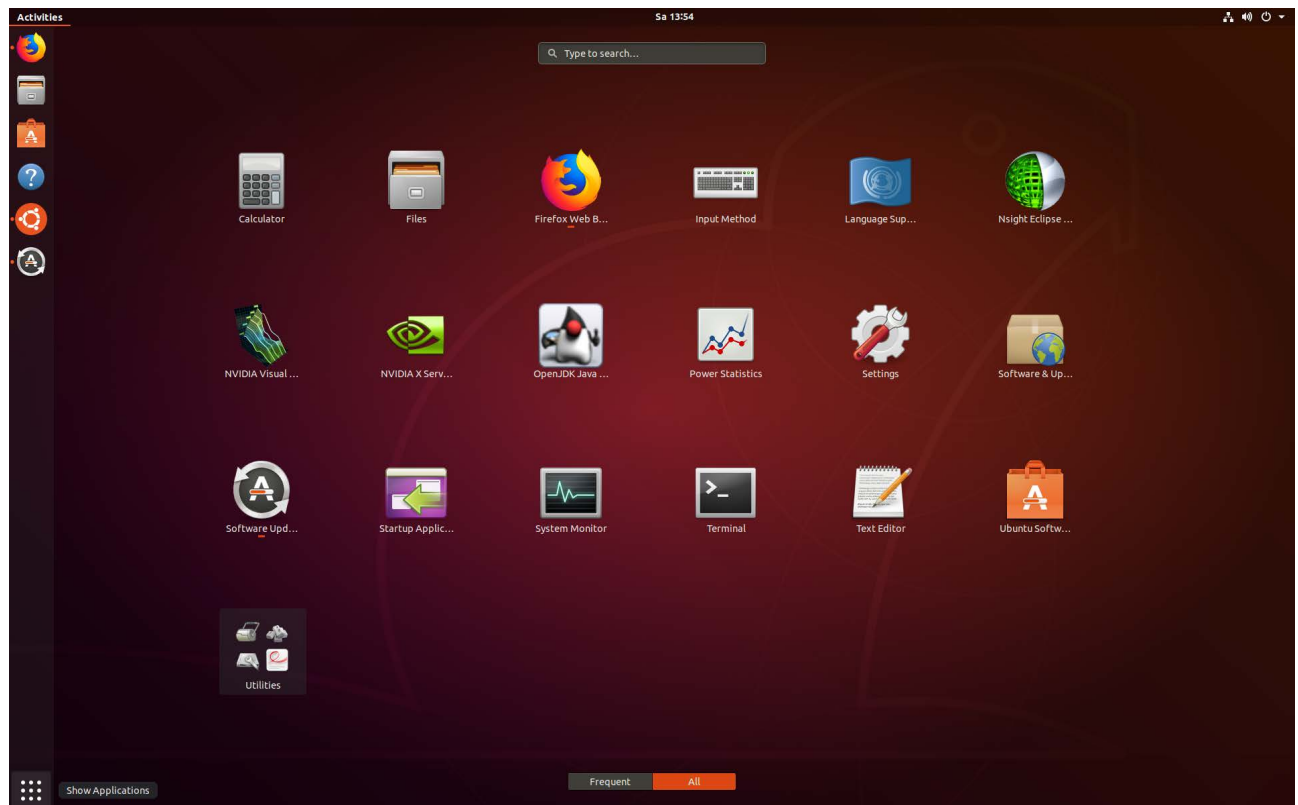
x	10000011011110000	106F0	203360
-y	1011000101011101	B15D	130535
z	101010110010011	5593	52623

$$z = 5 \cdot 16^3 + 5 \cdot 16^2 + 9 \cdot 16 + 3 = 21907$$

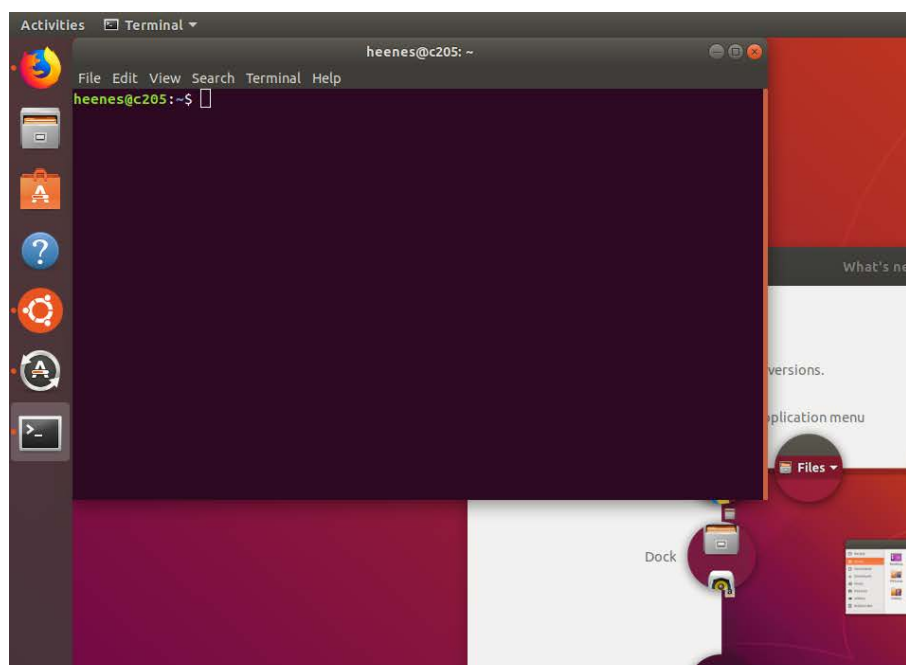
## Aufgabe 5: Kurze Einführung in Linux

Für dieses Modul wurde bewusst das Betriebssystem Linux ausgewählt. Dies liegt im Wesentlichen an dem Software-Entwicklungswerkzeug GCC (the GNU Compiler Collection, <https://gcc.gnu.org/>) Der Vorteil ist, dass man die Toolchain (Werkzeugkette), die in der Programmierung zur Anwendung kommt, an jeder Stelle aufbrechen kann. Hinzu kommt die hervorragende Dokumentation und Quelloffenheit der Werkzeuge.

Nach dem Login auf einem Linux Rechner sehen Sie ungefähr die folgende Oberfläche.



Klicken Sie mit der Maus auf das Programmsymbol Terminal.  
 Es geht dann ein sog. Terminalfenster (auch Shell genannt) auf.



Um in einem Terminalfenster ein Unterverzeichnis zu erzeugen, gibt es den Befehl `mkdir`. Zu jedem Befehl gibt es eine ausführliche Anleitung, die man durch die Eingabe `man mkdir` erhält.

Nach Eingabe des Befehls `mkdir Programme` und der Eingabe von `ls -l` sieht man die folgenden Verzeichnisse. `ls -l` listet alle Dateien und Verzeichnisse auf.

```
heenes@c205: ~  
File Edit View Search Terminal Help  
drwxr-xr-x 2 heenes heenes 4096 Okt 24 18:00 Videos  
heenes@c205:~$ ls -l  
total 32  
drwxr-xr-x 2 heenes heenes 4096 Okt 24 18:00 Desktop  
drwxr-xr-x 2 heenes heenes 4096 Okt 24 18:00 Documents  
drwxr-xr-x 2 heenes heenes 4096 Okt 24 18:00 Downloads  
drwxr-xr-x 2 heenes heenes 4096 Okt 24 18:00 Music  
drwxr-xr-x 2 heenes heenes 4096 Nov 10 13:58 Pictures  
drwxr-xr-x 2 heenes heenes 4096 Okt 24 18:00 Public  
drwxr-xr-x 2 heenes heenes 4096 Okt 24 18:00 Templates  
drwxr-xr-x 2 heenes heenes 4096 Okt 24 18:00 Videos  
heenes@c205:~$ mkdir Programme  
heenes@c205:~$ ls -l  
total 36  
drwxr-xr-x 2 heenes heenes 4096 Okt 24 18:00 Desktop  
drwxr-xr-x 2 heenes heenes 4096 Okt 24 18:00 Documents  
drwxr-xr-x 2 heenes heenes 4096 Okt 24 18:00 Downloads  
drwxr-xr-x 2 heenes heenes 4096 Okt 24 18:00 Music  
drwxr-xr-x 2 heenes heenes 4096 Nov 10 13:58 Pictures  
drwxr-xr-x 2 heenes heenes 4096 Nov 10 13:58 Programme  
drwxr-xr-x 2 heenes heenes 4096 Okt 24 18:00 Public  
drwxr-xr-x 2 heenes heenes 4096 Okt 24 18:00 Templates  
drwxr-xr-x 2 heenes heenes 4096 Okt 24 18:00 Videos  
heenes@c205:~$
```

Das Wechseln in das Unterverzeichnis „Programme“ geschieht durch die Eingabe von `cd Programme`. Möchte man nun den sog. Quellcode eines Programms eingeben (z. B. eines C Programms `hello.c`), so schreibt man `nano hello.c` in das Terminalfenster.

```
heenes@c205: ~/Programme  
File Edit View Search Terminal Help  
drwxr-xr-x 2 heenes heenes 4096 Okt 24 18:00 Desktop  
drwxr-xr-x 2 heenes heenes 4096 Okt 24 18:00 Documents  
drwxr-xr-x 2 heenes heenes 4096 Okt 24 18:00 Downloads  
drwxr-xr-x 2 heenes heenes 4096 Okt 24 18:00 Music  
drwxr-xr-x 2 heenes heenes 4096 Nov 10 13:58 Pictures  
drwxr-xr-x 2 heenes heenes 4096 Okt 24 18:00 Public  
drwxr-xr-x 2 heenes heenes 4096 Okt 24 18:00 Templates  
drwxr-xr-x 2 heenes heenes 4096 Okt 24 18:00 Videos  
heenes@c205:~$ mkdir Programme  
heenes@c205:~$ ls -l  
total 36  
drwxr-xr-x 2 heenes heenes 4096 Okt 24 18:00 Desktop  
drwxr-xr-x 2 heenes heenes 4096 Okt 24 18:00 Documents  
drwxr-xr-x 2 heenes heenes 4096 Okt 24 18:00 Downloads  
drwxr-xr-x 2 heenes heenes 4096 Okt 24 18:00 Music  
drwxr-xr-x 2 heenes heenes 4096 Nov 10 13:58 Pictures  
drwxr-xr-x 2 heenes heenes 4096 Nov 10 13:58 Programme  
drwxr-xr-x 2 heenes heenes 4096 Okt 24 18:00 Public  
drwxr-xr-x 2 heenes heenes 4096 Okt 24 18:00 Templates  
drwxr-xr-x 2 heenes heenes 4096 Okt 24 18:00 Videos  
heenes@c205:~$ cd Programme/  
heenes@c205:~/Programme$ ls -l  
total 0  
heenes@c205:~/Programme$ nano hello.c
```

Der Editor nano geht auf und man kann seinen Quellcode eingeben. Im Folgenden das Standardprogramm, welches in jeder Programmiersprache zum Testen verwendet wird.

```
#include<stdio.h>  
int main(){  
printf("Hello_World\n");  
return 0;  
}
```

Nach dem Speichern durch (Strg + O) und dem Beenden des Editors durch (Strg + X) sieht man durch die Eingabe von `ls -l` das erzeugte Programm. Die Übersetzung und Ausführung ist in Aufgabe 6) beschrieben. Die Programmiersprache

---

C wird im Rahmen des Moduls zur Beschreibung von Hochsprachenkonstrukten verwendet. Kleine Programme in C führen außerdem zu einem tieferen und praktischen Verständnis des Lernstoffs.

Weitere Befehle, die für Operationen mit Dateien und Verzeichnissen vorgesehen sind:

- cp - kopiert Dateien und Verzeichnisse (man cp)
- rm - löscht Dateien und Verzeichnisse (man rm)
- mv - benennt Dateien um (man mv)

Selbstverständlich können Sie Operationen mit Dateien und Verzeichnissen auch auf der graphischen Oberfläche durchführen.

Login auf den Rechnern der ISP mittels SSH

Die ISP betreibt sogenannte Login-Knoten. Der Login erfolgt über SSH<sup>4</sup>. Eine Einleitung zur Einrichtung finden Sie unter <https://support.rbg.informatik.tu-darmstadt.de/wiki/de/doku/computerhilfe/ssh>. Außerdem gibt ein HowTo aus dem Sommersemester 2020, welches die Einrichtung schrittweise erklärt.

Wünschenswert ist, dass sich jeder für die kommenden Übungen und Programmieraufgaben die notwendige Umgebung einrichtet und sich damit vertraut macht. Es lohnt sich!

## Aufgabe 6: Übersetzung und Ausführung von C Programmen

Sie können das obige C Programme auf Ihren Rechnern übersetzen und ausführen.

Zum Übersetzen eines C Programms soll im Folgenden der GCC Compiler benutzt werden. Die Übersetzung der Datei mit dem Namen hello.c wird durch folgende Befehle erreicht. Achtung: sh\$ muss nicht eingegeben werden.

```
sh$ gcc -o hello hello.c      (compilieren)
sh$ ./hello                  (Ausführen des Executables)
```

Der optionale Parameter -o mit der folgenden Zeichenkette hello gibt den Namen der übersetzten (und ausführbaren) Datei an. Der GCC Compiler ist für sehr viele Prozessorarchitekturen verfügbar. Eine Übersicht der möglichen Parameter beim GCC erhält man durch Eingabe man gcc in der Shell.

## Aufgabe 7: Plagiarismus und gute wissenschaftliche Praxis

Der Fachbereich Informatik misst der Einhaltung der Grundregeln der wissenschaftlichen Ethik großen Wert bei. Zu diesen gehört auch die strikte Verfolgung von Plagiarismus. Die Webseite informiert hierüber ausführlich.

Eine kompakte Einführung in wissenschaftliches Arbeiten und L<sup>A</sup>T<sub>E</sub>X finden Sie [hier](#).

---

<sup>4</sup> Secure Shell