



Übungen zur Vorlesung Grundlagen von Informatiksystemen  
Wintersemester 2022/23

Übungsblatt 11

---

**Aufgabe 1 Assembler-Programmierung**

Schreiben Sie für die CPU aus der Vorlesung ein Assembler-Programm, das die ersten  $n$  natürlichen Zahlen in den Datenspeicher  $S$  schreibt. Dabei soll die jeweils  $i$ -te natürliche Zahl in die Speicherzelle  $S(i)$  geschrieben werden ( $i = 1, 2, \dots, n$ ). Vor Ausführung des Programms enthalte die Speicherzelle  $S(0)$  die Anzahl  $n$  der zu schreibenden Zahlen.

Beispiel:

Speicher vor Ausführung des Programms:	$S(0)$	$S(1)$	$S(2)$	$S(3)$	$S(4)$	$S(5)$
Speicher nach Ausführung des Programms:	4	142	22	0	86	110
	?	1	2	3	4	110

Hier wurden also vier (siehe  $S(0)$  vor Ausführung des Programms) natürliche Zahlen an die Speicherzellen  $S(1), S(2), S(3)$  und  $S(4)$  geschrieben. Der Wert der Speicherzelle  $S(0)$  könnte ebenfalls durch das Programm verändert worden sein und z.B. nun 0 betragen, je nachdem, wie das Programm vorgeht. Das kommt darauf an, wie ihr konkretes Programm vorgeht.

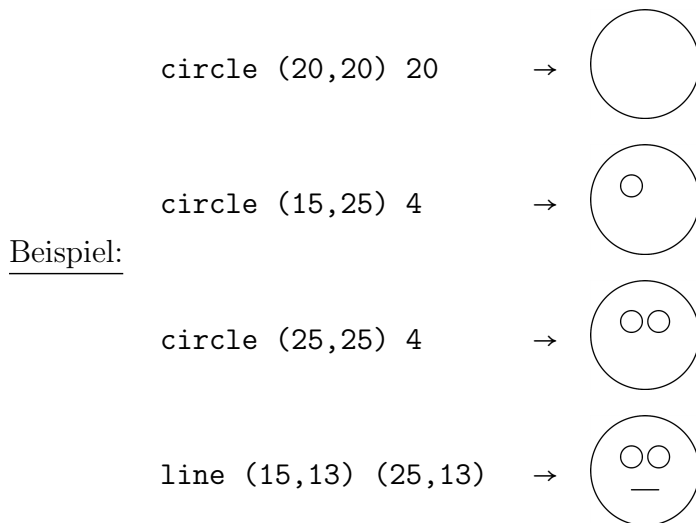
*Tipp:* Es könnte nützlich sein, die Zahlen in umgekehrter Reihenfolge schreiben zu lassen, also nicht zuerst  $1, 2, 3, \dots$ , sondern zuerst  $n, n-1, n-2, \dots$ .

**Aufgabe 2 Codierung**

- a) Welchem Wort entspricht die folgende Folge von Dezimalzahlen, wenn Sie davon ausgehen, dass es sich bei dieser um eine ASCII-Codierung handelt?

71 111 111 100 98 121 101 46

- b) Bilder, die wie in der Vorlesung durch Unterteilung in einzelne Pixel diskretisiert werden, nennt man auch *Rastergrafiken*. Eine ganz andere Art von Bildern stellen dagegen *Vektorgrafiken* dar: statt das Bild direkt zu speichern, werden Zeichenoperationen (Linien, Punkte, Kreise, etc.) codiert. Soll das zugehörige Bild angezeigt werden, führt der Rechner diese Operationen nacheinander aus, und zeichnet somit quasi jedes Mal das gesamte Bild neu.



Dieses Beispiel besteht aus vier Zeichenoperationen, die sich jeweils auf ein kartesisches Koordinatensystem beziehen: die **circle**-Operation spezifiziert den Mittelpunkt und den Radius eines Kreises, während die **line**-Operation den Start- und Endpunkt einer Linie spezifiziert.

Entwerfen Sie eine binäre Codierung, mit denen man Vektorgraphiken, die aus den folgenden Zeichenoperationen bestehen, codieren kann:

- **dot** (x, y) - zeichnet einen einzelnen Punkt an die Position (x, y) im Koordinatensystem.
- **line** (x1, y1) (x2, y2) - zeichnet eine gerade Linie von Punkt (x1, y1) im Koordinatensystem zum Punkt (x2, y2).
- **circle** (x, y) r - zeichnet einen Kreis mit Mittelpunkt (x, y) und Radius r.
- **rect** (x, y) w h - zeichnet ein Rechteck, dessen Linien parallel zu den beiden Achsen des Koordinatensystems sind. Die linke, obere Ecke des Rechtecks ist (x, y), seine Breite ist w und seine Höhe ist h.

Gehen Sie dabei davon aus, dass sämtliche Zahlenwerte natürliche Zahlen sind.

Codieren Sie anschließend mit dieser Codierung die Vektorgrafik aus dem obigen Beispiel.

### Aufgabe 3 Komprimierung

a) Wir betrachten die folgende Bitfolge:

0 0 0 1 1 0 0 0 0 1 1 1 1

Wie lauten die Lauflängencodierungen für diese Folge unter der Annahme, dass die Längen der einzelnen jeweils aus demselben Bit bestehenden Teilsequenzen als (i) zweistellige, (ii) dreistellige oder (iii) vierstellige Binärzahlen kodiert werden?

Welche Kompressionsraten werden in den Fällen (i), (ii) und (iii) jeweils erzielt?

b) In der Genforschung spielt die sog. DNA-Sequenzierung eine wichtige Rolle, bei der die Nukleotidsequenz einer DNA ermittelt wird, welche durch die Abfolge ihrer vier Nukleinbasen angegeben wird: Adenin, Cytosin Guanin und Thymin. Da diese üblicherweise mit ihrem

Anfangsbuchstaben abkürzt werden, ist eine Basensequenz also ein Wort der Sprache  $A^*$  über dem Alphabet  $A = \{A, C, G, T\}$ .

Bestimmen Sie für die folgende Basensequenz eine Huffman-Codierung:

G A T A T C A A A T G A C G A A T C A A T A C A A

#### **Aufgabe 4 Fehlerkorrigierende Codes**

Bei einer Datenübertragung hat der Empfänger folgende Bitfolge erhalten:

0101101 0001111 1011101 1100011 1111000

Da der Empfänger weiß, dass die Daten mit einem (7,4)-Hamming-Code codiert waren, geht er wie folgt vor:

- Er unterteilt die Bitfolge zunächst in Blöcke von je sieben Bits.
- Jede dieser Folgen unterteilt er wiederum in je 4 Daten- und 3 Paritätsbits.
- Er überprüft für jeden Block, ob die Paritätsbits korrekt sind oder ob es bei der Übertragung womöglich zu Fehlern (gekippte Bits) gekommen sein könnte.
- Je nach dem Ergebnis des vorherigen Schritts dekodiert er die vier Datenbits jedes einzelnen Blocks.

Das Ergebnis sind 20 dekodierte Datenbits. Wie lauten diese?

#### **Aufgabe 5 European Article Number**

Wählen Sie ein Produkt in Ihrem Umfeld, auf dem ein EAN-Strichcode abgedruckt ist. Notieren Sie sich die Nummer und legen Sie das Produkt dann zunächst zur Seite. Bestimmen Sie die Bitfolge, die nach dem EAN-Verfahren der notierten Nummer entspricht. Zeichnen Sie dann mit einem Lineal und einem schwarzen Stift den entsprechenden Strichcode.

Stimmt Ihre Zeichnung visuell mit dem Strichcode auf dem von Ihnen gewählten Produkt überein?

#### **Aufgabe 6 HTML + CSS**

In der Vorlesung wurde auf Folie 328 ein einfaches Beispiel für die Kombination eines HTML- und eines CSS-Dokuments vorgestellt, die hier noch einmal abgedruckt sind.

index.html

```
<!DOCTYPE html>
<html lang="de">
<head>
  <meta charset="utf-8">
  <title>Beispiel Dokument</title>
  <link rel="stylesheet" href="mystyle.css">
</head>
<body>
  <h1>Beispiel</h1>
  Dies ist ein <b>kurzes</b> Dokument.<p>
  Auf Wiedersehen.
</body>
</html>
```

mystyle.css

```
body {
  padding: 10px;
}

h1 {
  font-family: TimesNewRoman;
  color: black;
  margin-bottom : 20px;
}
```

- a) Kopieren Sie die beiden Dateien `index.html` und `mystyle.css` in einen Texteditor<sup>1</sup> Ihrer Wahl, z.B. Notepad. Korrigieren Sie sämtliche Kopierfehler (insbesondere bei den Sonderzeichen) und speichern sie beide Dateien unter diesen Namen in demselben Ordner ab. Öffnen Sie dann die Datei `index.html` in Ihrem Webbrowser.
- b) Ändern Sie in der Datei `index.html` die Überschrift (zwischen `<h1>` und `</h1>`) und den darunter stehenden Text ab. Speichern Sie die Datei erneut und refreshen Sie die zuvor im Webbrowser geöffnete Seite.
- c) Finden Sie durch eine Webrecherche heraus, wie man in einem HTML-Dokument einen Zeilenumbruch realisieren kann.
- d) Ändern Sie die Farbe der Überschrift des HTML-Dokuments, indem sie die entsprechende Stelle in der CSS-Datei editieren:
  - i. von schwarz zu rot.
  - ii. von schwarz zu einer RGB-Farbe, bei der die Beiträge der Grundfarben rot, grün und blau den Werten 128 : 170 : 32 entsprechen. Dazu werden die Dezimalzahlen 128, 170 und 32 zunächst in zweistellige Hexadezimalzahlen umgewandelt, und dann auf ein # folgend direkt hintereinander geschrieben. Beispiel: die RGB-Werte 255, 16, 0 entsprechen den Hexadezimalzahlen FF, 10, 00, sodass der zugehörige Farbton durch folgende Zeichenfolge beschrieben wird: `#FF1000`.

<sup>1</sup>Nicht zu verwechseln mit einem *Textverarbeitungsprogramm* wie z.B. MS Word oder LibreOffice Writer, s.a. <https://de.wikipedia.org/wiki/Texteditor>

- e) Ein Hyperlink zu einem anderen Dokument wird in HTML mit dem sog. Anchor-Tag realisiert, das folgende Form hat: `<a href="index2.html">Link Text</a>`. Kopieren Sie diesen Text in den Body-Teil Ihres HTML-Dokuments - was passiert, wenn Sie im Webbrowser auf „Link Text“ klicken? Legen Sie eine zweite HTML-Datei namens `index2.html` in demselben Ordner an, und klicken Sie erneut auf den Link.
- f) Welche anderen möglichen Änderungen an den Beispieldokumenten fallen Ihnen ein? Recherchieren Sie, wie Sie diese in HTML und CSS umsetzen können.

## Aufgabe 7 Routing

Ein *Uniform Resource Locator* (URL), z.B. `http://www.example.com/foo/bar/index.html`, besteht im einfachsten Fall aus folgenden Teilen:

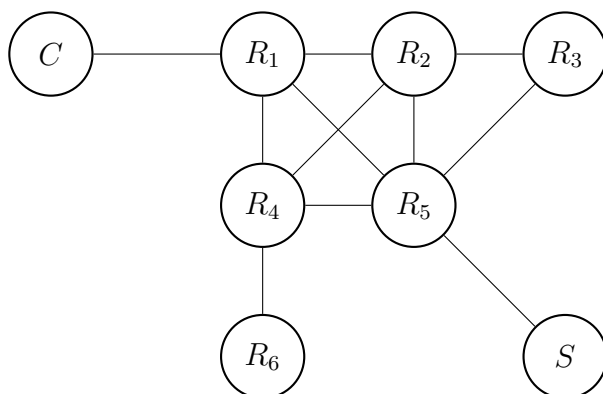
- dem sog. Schema (im Beispiel `http`)
- der Zeichenfolge `://`
- einem Hostnamen (im Beispiel `www.example.com`)
- einem Dateipfad (im Beispiel `/foo/bar/index.html`)

Klickt der Nutzer eine Webseite auf einen Link, der im HTML der Seite als `<a href="http://www.example.com/foo/bar/index.html">` kodiert ist, müssen folgende Schritte durchgeführt werden:

1. Anfrage an einen DNS-Server, um die dem Hostnamen zugeordnete IP-Adresse zu finden (`www.example.com` → `93.184.216.34`)
2. Da sich das Schema `http` auf das HTTP-Protokoll bezieht, Herstellen einer Netzwerkverbindung (ein sog. *Socket*) zu dem unter dieser IP-Adresse erreichbaren Webserver.
3. Senden einer HTTP-GET Anfrage nach dem Dateipfad der URL über diese Socket-Verbindung:

```
GET /foo/bar/index.html HTTP/1.1
```

Die Kommunikation zwischen dem anfragenden Rechner (Client) und dem anbietenden Rechner (Server) geschieht wie beim HTTP-Protokoll üblich mittels einzelner IP-Pakete. Nehmen wir an, der relevante Teil des Internets hätte folgenden Aufbau:



Host	IP-Adresse
<i>C</i>	92.146.18.1
<i>R</i> <sub>1</sub>	62.214.32.82
<i>R</i> <sub>2</sub>	149.11.96.43
<i>R</i> <sub>3</sub>	154.54.56.181
<i>R</i> <sub>4</sub>	130.117.0.129
<i>R</i> <sub>5</sub>	154.54.56.99
<i>R</i> <sub>6</sub>	152.195.233.122
<i>S</i>	93.184.216.34

Welchen Weg würde ein IP-Paket von dem Client-Rechner *C* zu dem Server-Rechner *S* nehmen, wenn die folgenden Routing-Tabellen zugrunde liegen?

	Network destination	Netmask	Gateway	Interface	Metric
$R_1$	91.0.0.0	255.0.0.0	130.117.0.129	eth0	1
	92.0.0.0	255.0.0.0	92.146.18.1	eth1	1
	93.0.0.0	255.0.0.0	149.11.96.43	eth2	1
	default	255.255.255.255	154.54.56.99	eth3	1

	Network destination	Netmask	Gateway	Interface	Metric
$R_2$	92.146.0.0	255.255.0.0	62.214.32.82	eth0	1
	92.172.0.0	255.255.0.0	154.54.56.181	eth1	1
	93.184.0.0	255.255.0.0	130.117.0.129	eth2	1
	default	255.255.255.255	154.54.56.99	eth3	1

	Network destination	Netmask	Gateway	Interface	Metric
$R_3$	92.146.18.0	255.255.255.0	149.11.96.43	eth0	1
	default	255.255.255.255	154.54.56.99	eth1	1

	Network destination	Netmask	Gateway	Interface	Metric
$R_4$	92.0.0.0	255.0.0.0	62.214.32.82	eth0	1
	93.1.0.0	255.255.0.0	149.11.96.43	eth1	1
	94.172.1.0	255.255.255.0	152.195.233.122	eth2	1
	default	255.255.255.255	154.54.56.99	eth3	1

	Network destination	Netmask	Gateway	Interface	Metric
$R_5$	92.146.18.0	255.255.255.0	62.214.32.82	eth0	1
	92.172.54.0	255.255.255.0	149.11.96.43	eth1	1
	93.28.216.0	255.255.255.0	154.54.56.181	eth2	1
	93.101.2.0	255.255.255.0	130.117.0.129	eth3	1
	93.184.216.34	255.255.255.255	93.184.216.34	eth4	1

	Network destination	Netmask	Gateway	Interface	Metric
$R_6$	default	255.255.255.255	130.117.0.129	eth0	1

Welchen Weg würde bei diesen Routing-Tabellen ein IP-Paket vom Server zurück zum Client nehmen?

## Aufgabe 8 Suchmaschinen

Beim Indizieren einer Dokumentensammlung unterscheidet man zwischen *Stoppwörtern* und *Inhaltswörtern*. Stoppwörter sind Worte, die so häufig in der jeweiligen Sprache vorkommen, dass sie wenig inhaltlich beitragen und daher nicht in den Index aufgenommen werden, z.B. *bin/bist/ist/...*, *der/die/das*, *und/oder/...*, etc. Dagegen nehmen wir an, dass die Dokumente einer Dokumentensammlung folgende Inhaltswörter in den angegebenen Häufigkeiten enthalten:

	$d_1$	$d_2$	$d_3$	$d_4$	$d_5$	$d_6$	$d_7$	$d_8$	$d_9$	$d_{10}$
animal	2			4			2		3	
car		5				7				8
cat	8			4			1		1	
jaguar				9		5	2		6	
leopard		1		1			3		7	
luxury	1					2		1		
vehicle						4		3		

Die Gesamtzahl aller Terme in den Dokumenten  $d_1-d_{10}$  sei durch folgende Tabelle gegeben:

$d_1$	$d_2$	$d_3$	$d_4$	$d_5$	$d_6$	$d_7$	$d_8$	$d_9$	$d_{10}$
52	103	85	66	158	114	72	44	87	23

- a) Berechnen Sie alle  $tf$  (Term-Frequency) und  $idf$  (Inverse Document Frequency)-Werte für die Dokumentensammlung.
- b) Besteht eine Suchanfrage  $s$  aus mehreren Termen  $t_1, \dots, t_k$ , kann man die Relevanz eines Dokuments  $d$  bezüglich  $s$  als Summe der Relevanzwerte von  $d$  für jeden einzelnen Term berechnen:  $\sum_{i=1}^k tfidf(t_i, d)$ .

Welches Dokument der Dokumentensammlung hat die höchste Relevanz für die Suchanfrage „jaguar car“ und welches für die Suchanfrage „jaguar animal“?