

Aufgaben zu Basiswissen IT

1. Der (nicht so) geheime Code

(a) Was mag dies bedeuten:

001001101 001111101 001111111 001111111 001111111 001111111
 001111111 001111111 001111111 001111111

Hinweis: ASCII-Code

(b) Wie lautet der String in Hexadezimalzahlen, in Dezimalzahlen, in Oktalzahlen?

(c) Wie groß ist die einfache 1-byte Check-Summe? Welchem ASCII-Zeichen entspricht dies? (Hinweis: Summe aller Binärzahlen modulo 256)

2. Na logisch logisch

In der (binären) Logik benutzt man oft sogenannte Wahrheitstafeln. Die elementaren logischen Operationen lassen sich damit übersichtlich darstellen. Schreibt man 1 für wahr und 0 für falsch, dann sehen die Wahrheitstafeln für NOT, OR und AND so aus:

	1	0
NOT	0	1

OR	1	0
1	1	1
0	1	0

AND	1	0
1	1	0
0	0	0

(a) Drücken Sie folgende Wahrheitstafel mit Hilfe der elementaren Operationen NOT, OR, AND aus.

XOR	1	0
1	0	1
0	1	0

(b) Beweisen Sie (mit Hilfe von Wahrheitstafeln) die *de Morganschen* Gesetze (NOT hat Vorrang vor AND/OR) :

a AND b = NOT(NOT a OR NOT b)

a OR b = NOT (NOT a AND NOT b)

(Hier zeigt sich, dass man für die binäre Logik im Prinzip nur zwei Operationen braucht.)

(c) Wieviel verschiedene Wahrheitstafeln gibt es?

3. Little Big Endian

Eine lange Ganzzahl (`long`, `int`) belegt normalerweise 4 Bytes im Speicher. Es gibt zwei Konventionen dies zu tun: Little Endian und Big

Endian. Bei Big Endian ist das höchstwertige Byte (Most Significant Byte) an der Adresse `ADR` gespeichert, das nächste Byte an der Adressen `ADR+1` usw., bei Little Endian ist es genau umgekehrt. Intel Rechner gehorchen der Little Endian Konvention. Eine IP-Adresse besteht auch aus 4 Byte, aber im Big Endian-Format laut Protokollspezifikation. Auf Intel-Rechner besteht also das Problem die Byte-Reihenfolge umzukehren. Nehmen wir an, wir hätten einen primitiven Prozessor, der ein internes Register (Arbeitsspeicher) besitzt und einen Stapelspeicher, den sogenannten Stack. Ein Stapelspeicher, auch Kellerspeicher genannt, funktioniert nach dem LIFO-Prinzip (Last-In-First-Out), d.h. der letzte Wert der auf den Stapel gelegt wurde (PUSH), kommt als erstes wieder heraus (POP). Desweiteren kann der Prozessor Werte aus dem Hauptspeicher in sein Register laden und wieder zurückspeichern. Der Prozessor versteht also folgende Befehle:

Befehl	Operation
<code>MOV adr</code>	Holt das Byte von der Adresse <code>adr</code> in das Register
<code>STO adr</code>	Speichert den Inhalt des Registers bei Adresse <code>adr</code>
<code>PUSH</code>	Schiebt den Inhalt des Registers auf den Stack
<code>POP</code>	Holt den obersten Wert des Stacks ins Register

Schreiben Sie ein möglichst kurzes Programm, das die Reihenfolge der Bytes an den Speicherstellen `0x01`, `0x02`, `0x03`, `0x04` umkehrt.