Übungen zu Computergrundlagen WS 2019/2020

Übungsblatt 9: Zahlen im Computer

13. Dezember 2019

Allgemeine Hinweise

- Abgabetermin für die Lösungen ist **Freitag, 20.12.2019, 11:00 Uhr**
- Schickt die Lösungen bitte per Email an Euren Tutor:
  - Montag 14:00–15:30: Moritz Schumacher (`mschumacher@icp.uni-stuttgart.de`)
  - Dienstag 9:45–11:15: Samuel Tovey (`stovey@icp.uni-stuttgart.de`)
  - Dienstag 15:45–17:15: Philipp Stärk (`pstaerk@icp.uni-stuttgart.de`)
  - Mittwoch 15:45–17:15: Marco Brückner (`mbrueckner@icp.uni-stuttgart.de`)
  - Donnerstag 9:45–11:15: Ingo Tischler (`itischler@icp.uni-stuttgart.de`)
- Die Übungen sollen von Gruppen von jeweils *zwei* (nur in Ausnahmefällen drei) Leuten bearbeitet werden. Bitte gebt *nur eine Lösung pro Gruppe* ab und nennt in eurer Abgabe alle Mitglieder eurer Gruppe!
- Als Lösung der Aufgabe soll ein einziges Shell-Skript erstellt werden, welche ihr dann per E-Mail an euren Tutor schickt.

Aufgabe 9.1: Zahlensysteme (5 Punkte)

- **9.1.1** Berechne die folgenden Zahlen a bis e, indem Du zwischen verschiedenen Zahlensystemen umrechnest. Dabei steht 1234_7 für die Zahl 1234 im Zahlensystem zur Basis 7. $a_{10} = 1234_7$ bedeutet also, daß die Zahl 1234 im Zahlensystem zur Basis 7 ins Zahlensystem zur Basis 10 (Dezimalsystem) umgerechnet werden soll. Zeigt dabei eure Rechenschritte (Multiplikation oder schriftliche Division). (2 Punkte)

  Hinweis: In einigen Fällen dürfte die nebenstehende Tabelle nützlich sein.

- $a_{10} = 1234_7$
- $b_{10} = 10100101_2$
- $c_{16} = 1234_{10}$
- $d_2 = CD_{16}$
- $e_8 = 10000001_2$

2	**7**	**8**	**10**	**16**
0	0	0	0	0
1	1	1	1	1
10	2	2	2	2
11	3	3	3	3
100	4	4	4	4
101	5	5	5	5
110	6	6	6	6
111	10	7	7	7
1000	11	10	8	8
1001	12	11	9	9
1010	13	12	10	A
1011	14	13	11	B
1100	15	14	12	C
1101	16	15	13	D
1110	20	16	14	E
1111	21	17	15	F
10000	22	20	16	10

- **9.1.2** Schreibt nun ein `bash`-Skript, das die Umrechnung für euch vornimmt. (2 Punkte)

  Hinweise:

  - In `bash` kann ein String, der z. B. in der Variable `STRING` steht, mittels `echo $STRING | grep -o .` in ein Array von Zeichen umgewandelt werden.
  - Mit `rev` kann die Reihenfolge umgekehrt werden.
  - `bash` unterstützt nativ nur Integerarithmetik, besonders hilfreich sind dabei die *modulo* und die Divisions Operation `%` und `/`.
  - Auch der Kommandozeilen-Taschenrechner `bc` kann verwendet werden:
    `result=$(echo ‘‘1+2’’ | bc -l)`
  - Es kann hilfreich sein, schrittweise vorzugehen, also die Zahl zunächst beispielsweise im Dezimalsystem darzustellen um von dort aus in eine andere Basis zu wechseln.

- **9.1.3** Im Computerumfeld wird häufig das Hexadezimalsystem ($B = 16$) verwendet. Welche Vorteile bietet es gegenüber dem. . . (1 Punkt)
  - Dezimalsystem ($B = 10$)?
  - Oktalsystem ($B = 8$)?

Aufgabe 9.2: Fließkommazahlen (5 Punkte)

Ganzzahlen speichert der Computer wie oben im Zweiersystem. Da in der Physik viele Größen jedoch kontinuierliche Werte annehmen können, sind Gleitkomma- oder Fließkommazahlen für uns mindestens genau so wichtig. Während das Zweiersystem Ganzzahlen exakt abbildet, sind Fließkommazahlen nur aus zwei Ganzzahlen bestehende Näherungen für reelle Zahlen. Dies bringt in der Praxis einige Probleme mit sich, die wir hier anhand von Beispielen beleuchten.

Üblicherweise werden Fließkommazahlen nach dem IEEE 754-Standard gespeichert. Wir verwenden im Folgenden 32 Bit pro Zahl, was *single precision* genannt wird. Die 32 Bit werden wie folgt verwendet:

- 8 Bit für den Exponenten als Ganzzahl p, wobei vor der Speicherung sogenannte *Bias*wert $B = 127$ addiert wird;
- 23 Bit für die Mantisse als Ganzzahl m, wobei vor der Speicherung 1 abgezogen wird;
- 1 Bit s für das Vorzeichen der Mantisse.

Die so gespeicherte Zahl hat dann den Wert

$$x = (-1)^s(1 + m)2^{p-B}.$$

Der Exponent kann nur Werte aus $[-126{,}127]$ annehmen, weil $p - B = 0$ und $p - B = 255$ für Sonderfälle reserviert sind. Die Mantisse kann Werte aus $[1{,}2)$ annehmen. Genaue Erläuterungen finden sich z.B. bei `https://de.wikipedia.org/wiki/IEEE_754`.

- **9.2.1** Was sind die betragsmäßig und absolut kleinsten und größten Zahlen, die man so darstellen kann (abgesehen von der Null)? Gebt sie als Zweierpotenzen an, jeweils mit Begründung. Rechnet sie ins Dezimalsystem um, um zu bestimmen, wie viele signifikante Stellen eine so dargestellte Zahl hat. (1 Punkt)
- **9.2.2** Gebt die Bit-Darstellung der Zahl $-1{,}602176634 \cdot 10^{-19}$ an und zeigt, wie ihr diese bestimmt habt. (2 Punkte)
- **9.2.3** Berechnet $(1 + 2{,}1 \cdot 10^{-6}) - 1$, sowohl konventionell im Zehnersystem als auch in der IEEE 754-Darstellung. Woher kommt die Abweichung von mehreren Prozent zwischen diesen beiden Ergebnissen? (2 Punkte)