

1. Aufgabe

Daniel Düsentrieb arbeitet gerade an einer neuen Lichtorgel. Sie besteht zunächst aus 3 Glühbirnen, die einzeln ein- und ausgeschaltet werden können.

- (a) Wieviele Zustände kann die Lichtorgel annehmen?
- (b) Wieviele Zustände sind es im allgemeinen Fall, wenn die Anzahl der Glühbirnen n beträgt, wobei n eine beliebige natürliche Zahl sein kann?
- (c) Und wieviele Glühbirnen braucht Düsentrieb mindestens, wenn er 1000 Zustände ermöglichen will,
- (d) Und wieviel für 65356 Zustände?

2. Aufgabe

- (a) Im Gegensatz zur Kommunikationsinformatik wird in der Informationsverarbeitung die Darstellung in der 2er-Potenz bevorzugt. Zur Angabe von Bitraten und ähnlichen wird die gebräuchliche Dezimaldarstellung herangezogen. Untersuchen Sie die entsprechenden Abweichungen zwischen den beiden Darstellungen. Werden die absoluten Abweichungen kleiner, größer oder bleiben sie konstant? Wie verhält es sich mit den relativen Abweichungen?
- (b) Wie groß ist die maximale Größe des Arbeitsspeichers, den ein Prozess unter einem 32-Bit-Betriebssystem benutzen darf, falls wie üblich jedes einzelne Byte adressiert werden kann? Wie groß wäre der maximale Arbeitsspeicher, falls nur jedes einzelne Wort (2 Byte) adressiert werden muß?
- (c) Die Deutschen sollen durch eine Bitfolge eindeutig „codiert“ werden. Wie lang muß diese Bitfolge sein, unter der Annahme, dass es aktuell ca. 80 Millionen Bundesbürger gibt?

3. Aufgabe

Daniel Düsentrieb arbeitet gerade an einem Computer, der intern alle Zahlen zur Basis 13 darstellt. Er verwendet dazu die Symbole 0 bis 9 und A, B und C.

- (a) Das erste Ergebnis, das ihm der Rechner liefert, lautet ABC_{13} . Können Sie ihm sagen, was dies im Dezimalsystem bedeutet? Und was bedeutet 13_{13} ?
- (b) Welches Ergebnis müsste sein Computer liefern, wenn das Ergebnis wie allgemein bekannt 42_{10} lautet? Und wie lautet seine Ausgabe für 4711_{10} ?

4. Aufgabe

Daniel Düsentrieb entscheidet sich doch für einen profanen Computer und beschäftigt sich deshalb mit Dual-, Hexal- und dem Oktaldezimalsystem. Wie sieht die oktale, die hexadezimale und die dezimale Darstellung der Zahl 1100011001_2 aus? Wie sieht die oktale, die hexadezimale und die binäre Darstellung der Zahl 313_{10} aus?

5. Aufgabe

Wir betrachten nun einen Rechner, welcher nur Daten bis zu 8 Bit verarbeiten kann. Der Rechner soll die folgende Berechnung für Natürliche Zahlen ausführen: $160_{10} + 210_{10}$. Da es sich nur um positive Zahlen handelt, kann in der binären Entsprechung die direkte binäre Darstellung verwendet werden.

- (a) Wie lautet das entsprechende Ergebnis? Rechnen Sie sowohl binär als auch hexadezimal.
- (b) Zeigen Sie am Beispiel, dass das Ergebnis dem tatsächlichen Ergebnis modulo 256 entspricht.
- (c) Wie sieht die Codierung des ASCII-Textes „GDI“ aus, und zwar binär und hexadezimal?

6. Aufgabe

Wieviele Stellen hat die Zahl 2^{123} in oktaler und in hexadezimaler Darstellung? Wieviele Stellen sind für die dezimale Darstellung notwendig? Bevor Sie diese Zahl exakt berechnen, machen Sie eine sinnvolle Schätzung!

7. Aufgabe

Eine ganze Zahl hat im Zweierkomplement die Darstellung 1100101.

- (a) Wie lautet die entsprechende ganze Zahl, wenn man diese so wie angegeben interpretiert? Wie sieht das Ergebnis aus, wenn man die Zahl als 8-Bit Zahl interpretiert?
- (b) Wir betrachten nun das Bitkomplement (Vertauschen von 0 und 1). Welches Ergebnis liefert die Addition mit dem Bitkomplement und warum?

8. Aufgabe

Wie sehen die Zweierkomplement-Darstellungen der Zahlen $z = -13$, $u = -111$ und $v = -313$ mit minimaler Länge aus?

9. Aufgabe

In dieser Aufgabe sollen konkrete Zahlendarstellung für Java-Datentypen ermittelt werden. Allerdings wird statt der Single- bzw. Double Precision Float-Variante für diesen Zweck eine einfache Simple Precision Variante eingeführt.

- (a) Betrachtet wird der Java-Datentyp **short**. Zunächst soll der Zahlenbereich angegeben werden, der mit Hilfe dieses Typs dargestellt werden kann. Anschließend sollen die Zahlen **45** und **- 58** als Java-short binär ausgedrückt werden.
- (b) In dieser Teilaufgabe soll die **Gleitkommazahl-Darstellung** angewendet werden. Zur Vereinfachung wird die folgende Simple-Precision Variante eingeführt, die sich an den realen Single- und Double Precision IEEE 754 Standard orientiert und statt 4 bzw. 8 Byte nun mit 1 Byte zur Darstellung einer Gleitkommazahl auskommt. Von den insgesamt zur Verfügung stehenden 8 Bits wird das erste für das Vorzeichen verwendet, die nächsten drei Bits zur Darstellung des Exponenten und die restlichen zur Darstellung der Mantisse in der wie im IEEE 754 Standard umgesetzten normalisierten Form. Für die Zahlen **-11,4** und **0,90** sind die entsprechenden Binärmuster zu ermitteln!

10. Aufgabe

Im Kapitel 2.4 wurde kurz das Betriebssystem eingeführt, welches dem Benutzer den Zugang zum Rechner erleichtern soll. Gegenstand dieser Aufgabe ist es, sich mit der „klassischen“ DOS-Shell vertraut zu machen, um einen minimalistischen Umgang mit dem Rechner zu üben. Auf diese Weise sollen die relevanten Betriebssystem-Befehle im Rechnerbetrieb selbständig erfahren werden.

Bitte beachten Sie, dass DOS im Gegensatz zu modernen Betriebssystemen wie UNIX bzw. unixoiden Systemen keinen Mehrbenutzer-Betrieb unterstützt und im Grunde auch keine Prozessverwaltung anbietet, also auch keinen Mehrprozess-Betrieb erlaubt. Aus diesem Grund ist die Komplexität der DOS-Befehle deutlich geringer als die der entsprechenden UNIX-Befehle. Bitte fassen Sie DOS nur als ersten Einstieg in die Systemwelt auf. Eine komplexe und elegante Systemprogrammierung auf hohem Niveau ist damit nicht möglich.

- (a) Welches Betriebssystem ist auf dem Rechner installiert? Welche logischen Plattenlaufwerke sind vorhanden? Wie ist das Dateisystem organisiert? Wie sieht die rudimentäre Verwaltung der Zugriffsrechte aus?
- (b) Verwenden Sie im folgenden nur noch die **DOS-Shell**, um mit dem Betriebssystem zu kommunizieren. Starten Sie dazu ein entsprechendes DOS-Fenster (Eingabeaufforderung). Mit Hilfe des DOS-Befehls **dir** können Sie sich die Dateien in einem Verzeichnis anzeigen lassen. Welche Angaben werden sonst noch angezeigt?
- (c) Mit Hilfe des DOS-Befehls **attrib -x +y datei.txt** können die Attribute (Archive Bit **a**, Schreibgeschützt **r**, Versteckt **h**, Systemdatei **s**) manipuliert werden. Mit den DOS-Befehlen **attrib *** bzw. **attrib datei.txt** können die Attribute angezeigt werden.

- (d) Auf einer Festplatte mit hoher Kapazität können im einfachsten Fall viele Dateien mit Hilfe einer flachen Struktur gespeichert werden. Zur Verbesserung der Übersicht bzw. auch zur Beschleunigung der Zugriffszeiten können **Verzeichnisse** (Directories) bzw. Unterverzeichnisse verwendet werden. Was bedeutet in diesem Zusammenhang der Begriff **Zugriffspfad**? Stellen Sie die Verzeichnis-Struktur Ihres Home-Verzeichnisses grafisch dar und zwar in Form eines Baumes! Das **aktuelle Verzeichnis** ist das Verzeichnis, in dem gerade gearbeitet wird. DOS merkt sich auch das letzte Verzeichnis auf einem Laufwerk, auf dem zuletzt gearbeitet wurde.
- (e) Erstellen Sie auf Ihrem Netzlaufwerk (Home-Verzeichnis) ein Unterverzeichnis mit dem Namen **gdi** und darunter zwei Verzeichnisse mit Namen **kapitel1** und **kapitel2**. Zum Navigieren in der Verzeichnis-Struktur verwenden Sie bitte den DOS-Befehl **cd** (change directory). Mit Hilfe von **cd ..** wechseln Sie in ein übergeordnetes Verzeichnis, mit **cd <ordnername>** wechseln Sie in das entsprechende Unterverzeichnis. Mit Hilfe des DOS-Befehls **md** (make directory) wird ein entsprechendes Unterverzeichnis erstellt, also in unserem Fall: **md gdi**.
- (f) Falls kein Zugriffspfad angegeben wird, sucht DOS beim Zugriff auf eine Datei (Programm) nur im aktuellen Verzeichnis des Standardlaufwerks nach dieser Datei. Um die ständige, mühsame Verwendung des Zugriffspfads (bestehend aus Laufwerk, Verzeichnisname und Dateiname) besonders bei Aufrufen von Programmen zu vermeiden, bietet sich der DOS-Befehl **path** an. Die spezielle Umgebungsvariable **PATH** enthält eine Menge der Zugriffspfade (durch Semikolon getrennt), die DOS ermöglicht, ausführbare Dateien aufzurufen, die sich nicht im aktuellen Verzeichnis befinden. Bitte beachten Sie, dass DOS keine Gross-/Klein-Schreibweise unterscheidet. Die unterschiedliche Schreibweise soll helfen, zwischen den beiden Elementen Umgebungsvariable **PATH** und DOS-Befehl **path** zu unterscheiden.
- (g) Verwenden Sie den DOS-Befehl **path**, um sich den aktuellen Wert der **speziellen Umgebungsvariablen PATH** anzusehen. Was genau sehen Sie? Erkennen Sie in diesem Zusammenhang etwas JAVA-relevantes? Bitte verwenden Sie diesen Befehl mit Vorsicht. Wird nur der Befehl **path** angegeben, so wird der aktuelle Pfad angezeigt. Wird **path** mit einem Semikolon eingegeben (**path ;**), wird der Suchpfad auf Null zurückgesetzt, d.h. es ist kein erweiterter Suchpfad mehr definiert.
- (h) **Umgebungsvariablen** können innerhalb von Stapelverarbeitungsprogrammen (Batch-Betrieb) verwendet werden wie variable Parameter. Vor und nach jeder Umgebungsvariablen muß ein Prozentzeichen (%) eingegeben werden. Um z.B. den Wert für **PATH** in DOS anzupassen, nachdem die aktuelle Java-Version installiert wurde, muss z.B. der folgende DOS-Befehl verwendet werden: **set PATH=%PATH%;c:\Programme\jdk1.6.0._17\bin** verwendet werden.
- (i) Mit Hilfe des DOS-Befehls **set** kann man sich alle Umgebungsvariablen anzeigen lassen. Mit dem DOS-Befehl **set name** kann man sich den aktuellen Wert der Umgebungsvariablen **name** anzeigen lassen. Um die Umgebungsvariable **name** zurückzusetzen, wird der DOS-Befehl **set name=** verwendet. Zum Beispiel wird mit Hilfe von **set classpath=** die Umgebungsvariable **CLASSPATH** zurückgesetzt. Bitte beachten Sie, dass DOS im Gegensatz zu Java nicht zwischen Gross- und Kleinbuchstaben unterscheidet.

- (j) Mit Hilfe der in Abschnitt (i) eingeführten DOS-Befehle können die Werte der Umgebungsvariablen nur temporär (bzgl. der entsprechenden DOS-shell) gesetzt werden. Um die entsprechenden Werte permanent zu setzen, kann in DOS die spezielle Batch-Routine (Shell-Programm, ab **Version 64-Bit Windows 7 nicht** mehr verfügbar) **autoexec.bat** verwendet werden, welche standardmäßig immer zu Beginn aufgerufen wird. Eine (optionale) **autoexec.bat-Datei** muss im Stammverzeichnis des Laufwerks (Platte) erstellt werden, von dem aus DOS gestartet wird.
- (k) Alternativ können Umgebungsvariablen komfortabler unter **Systemsteuerung/Erweitert/Umgebungsvariablen** gesetzt bzw. ermittelt werden.
- (l) Erstellen Sie zunächst eine einfache Text-Datei mit dem Namen „Hello.txt“ und dem Inhalt: „Hello World!“. Verwenden Sie dazu z.B. den Editor **notepad** bzw. den älteren DOS-Editor **edit** (ebenfalls in neueren Versionen nicht mehr verfügbar) den sie einfach aus der DOS-shell aufrufen.
- (m) Lassen Sie sich den Inhalt der gerade erstellten Datei auf dem Bildschirm ausgeben. Verwenden Sie dazu den DOS-Befehl **type**. Lassen Sie sich vorher alle Dateien im aktuellen Verzeichnis anzeigen. Alternativ kann der Befehl **more** verwendet werden. Erkennen Sie einen Unterschied? Was ist eine **Pipe**?
- (n) Verwenden Sie den DOS-Befehl **copy**, um die Datei Hello.txt zu kopieren mit einem Namen Ihrer Wahl. Testen Sie mit Hilfe des type-Befehls, ob der Kopiervorgang erfolgreich war.
- (o) Nennen Sie die Ursprungsdatei Hello.txt um in Abfall.txt, verwenden Sie dazu den DOS-Befehl **ren** (rename.) Löschen Sie anschließend diese Datei mit Hilfe des DOS-Befehls **del**. Bitte beachten Sie, dass Unterverzeichnisse nur mit dem DOS-Befehl **rmdir** (remove directory) gelöscht werden können.
- (p) Machen Sie sich mit der **Wildcard** vertraut *. Sie können sich beispielsweise mit Hilfe des Befehls **dir t*** alle Dateien im aktuellen Verzeichnis anzeigen lassen, die mit t beginnen. Versuchen Sie die wildcard-Option auch in anderen Befehlskontexten.