Rechnersysteme und Netze
Zusammenfassung

Andreas Rain

12. Februar 2011

Inhaltsverzeichnis

1 Zahlensysteme und Kodierung

1.1 Umrechnen

1.1.1 Binär - Oktal - Hex

Binärdarstellungen in Oktal und Hex lässt sich durch zusammenfassen von 3- bzw. 4 Bits realisieren. Umgekehrt entsprechen die Werte 3 bis 4 Bits. Von Oktal zu Hex sollte man zuerst die Binärdarstellung machen und dann in die jeweilige Darstellung konvertieren.

1.1.2 Dezimal - Beliebiges System

Man teilt die Zahl durch die Basis des Zielsystems mit Rest. Der Rest von unten nach oben gelesen ist die konvertierte Zahl.

Beispiel: $999_{10} = (X)_{16}$

$$\begin{aligned} 999_{10} : 16 &= 62R(7) \\ 62_{10} : 16 &= 14R(E) \\ 3_{10} : 16 &= 0R(3) \\ &= 3E7_{16} \end{aligned}$$

1.1.3 Beliebiges System - Dezimal

Mittels des Horner Schemas lässt sich diese Berechnung wie folgt durchführen: $(2555)_7 = ((2 \cdot 7 + 5) \cdot 7 + 5) \cdot 7 + 5 = 971_{10}$

1.2 Zweierkomplement (8-Bit)

Vom Dezimalsystem ins Binärsystem muss einfach die Zahl in das Binärsystem umgewandelt werden und falls sie negativ ist noch invertiert und inkrementiert werden.
Beispiele:
$127_{10} = 01111111_2 = 7F_{16}$
$4_{10} = 00000100_2 = 04_{16}$
$1_{10} = 00000001_2 = 01_{16}$
$0_{10} = 00000000_2 = 00_{16}$
$-1_{10} = 11111111_2 = FF_{16}$
$-4_{10} = 11111100_2 = FC_{16}$
$-127_{10} = 10000001_2 = 81_{16}$
$-128_{10} = 10000000_2 = 80_{16}$

1.3 Einerkomplement

Ähnlich wie das Zweierkomplement, allerdings ist die größte binär dargestellete Zahl -0 und nicht -128 (8-Bit).

1.3.1 Kodierung

1.3.2 ASCII

7-bit, nur Roman, Sonderzeichen, Steuerzeichen. Speicherineffizient aber keine Unterstützung anderer Sprachen

1.3.3 UTF-8

Ein in UTF-8 kodiertes Zeichen besteht aus 1 bis 4 Bytes:

- Bei einem Byte ist das erste Bit 0 und die restlichen Bits entsprechen ASCII
- Ab einer Länge von 2 Bytes beginnt das erste Byte z.B. mit 110xxxxx, wobei die Anzahl der Einsen vor der 0 angeben, aus wie vielen Bytes das kodierte Zeichen insgesamt besteht. Alle folgenden Bytes beginnen dann mit 10xxxxxx.
- Beispiel für 4 Byte Kodierung: 11110xxx 10xxxxxx 10xxxxxx 10xxxxxx

Die x aneinander gereiht ergeben den Unicode des Zeichens.

1.3.4 UTF-16

Ein in UTF-16 kodiertes Zeichen besteht auf 2 oder 4 Bytes:

- Bei zwei Bytes können alle Bits direkt für den Unicode genutzt werden. (bis 0xFFFF)
- Bei Werten über 0xFFFF, also 4 Bytes, beginnt das erste Byte mit 110110xxx... und das dritte mit 110111xxx....
- Die Bits für die x erhält man, indem 0x10000 von dem gewünschten Wert subtrahiert, da die CodePoints aufgeteilt werden müssen.

Ob bei der Kodierung nun das vordere oder das hintere Byte höherwertig ist, bestimmt die Angabe Little-, bzw. Big-Endian.
Unser „normales“ Alphabet, also hauptsächlich ASCII Zeichen, wird besser mit UTF-8 kodiert, da hier 1 Byte ausreicht. Weiter hinten liegende Zeichen wie z.B. chinesische benötigen dafür in UTF-16 nur 2 Bytes, wohingegen UTF-8 dann schon 3 oder 4 Bytes benötigt.

1.3.5 UTF-32

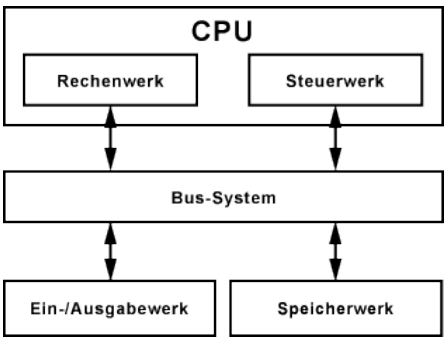
UTF-32 ist die einfachste Kodierung, da jedes der 32 Bits direkt für den Unicode genutzt werden kann. Dadurch ist auch die Umrechnung von UTF-8 oder 16 zu UTF-32 am einfachsten. Allerdings muss man von einer Umwandlung von UTF-16 aus eventuell die Rückrechnung beachten. (+ 0x10000)

1.3.6 BOM

Byte Order Mark. Gibt an welche endianess der code hat. Little Endian oder Big Endian. Nötig bei UTF-16 und 32, da man an Hand der Bytes nicht feststellen kann in welcher Reihenfolge sie gelesen werden müssen.

2 Rechnerarchitekturen

2.1 Von-Neumann Rechnerarchitektur



CPU Steuerwerk und Rechenwerk, Steuerung der Befehlsabfolge und Rechenoperationen.

CU speichert Daten und Text (Instruktionen).

I/O Ein-/Ausgabe von Daten und Schnittstellen.

Von-Neumann-Flaschenhals: sowohl Text als auch Daten müssen über einen Bus. CPU viel schneller als Busse. Rechenzeit verschenkt.

2.2 Speicherzugriffszeiten

1. (CPU) Register
2. (CPU) Cache
3. Arbeitsspeicher (RAM)
4. Solid State Disks
5. Magnetische Laufwerke
6. Optische Laufwerke
7. Magnetbänder

2.3 Betriebssysteme

Ein Betriebssystem ist unter anderem für das Management von Ressourcen und Abstraktion von Hardware zuständig. Hierzu gehören Speichermanagement, Prozessmanagement, Interruptmanagement, Dateisystemmanagement, Gerätetreiber, System Calls als Abstraktion usw. .

2.3.1 Virtuelle Maschine

Eine Virtuelle Maschine ist ein aus Software bestehender virtuelle Rechner. Als Betriebssystem abstrahiert sie die darunter liegende Hardware. Als Laufzeitumgebung simuliert sie einen virtuellen Prozessor.

2.3.2 Kernelspace und Userspace

Kernelspace ist ein besonderer Speicherbereich für den Kernel. Prozesse in diesem Bereich haben volle Sichtbarkeit und vollen Zugriff auf die Hardware. Kernel/privilegierter Modus.

Userspace ist der restliche Speicherbereich. Prozesse sind beschränkt auf einen zugewiesenen Speicherabschnitt und haben keinen direkten Zugriff auf Ressourcen. User/ unprivilegierter Modus.

2.3.3 Monolithischer Kernel

alle Komponenten als ein Prozess im Kernelspace. Volle Sichtbarkeit. Ein Fehler führt zu system failure. Sicherheitsbedenken.

2.3.4 Microkernel

Nur die wichtigsten Komponenten wie Interprocesskommunikation und Speicher-/Prozessmanagement laufen im kernelmode. Alle anderen Komponenten laufen als eigene Server im Usermode. Nicht kritische Fehler führen nur zum Absturz des betroffenen Servers und können behoben werden. Kritikpunkte sind vor allem Synchronisationsaufwand und zu einigschränkter Zugriff für Prozesse die privilegiert sein müssen.

2.3.5 Hybrid

Vereint die Ideen der beiden anderen. Prozesse die priviligiert sein müssen laufen im Kernelmode. Alles andere als Server oder Bibliothek im Usermode. Reduziert Synchronisationsaufwand, erhöht Fehlergefahr.

2.4 Dateisysteme

Abstrahiert Eingabe/Ausgabe Geräte. Dateien als Geräte unabhängige Entitäten. Organisiert Daten in Dateien und Verzeichnisse. Sichert integrität und Zugriff. Managed Speicher.

2.4.1 FAT

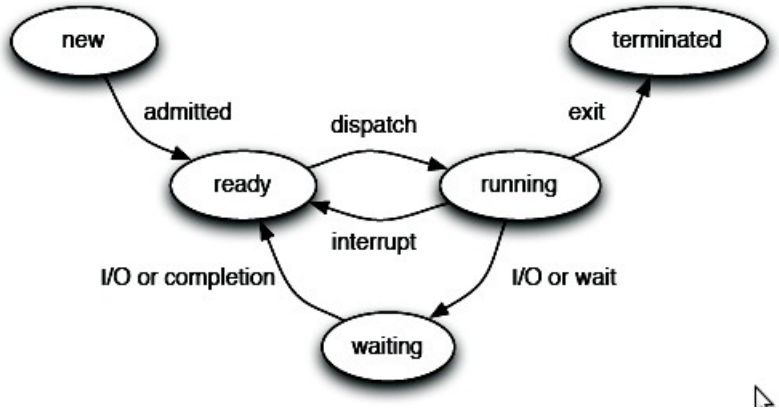
FAT teilt Partitionen in gleich grosse cluster. Die File Allocation Table ist eine Tabelle, die für jeden cluster einen Eintrag hält. Die Directory Table beinhaltet Namen und Metadaten von Dateien und Verzeichnissen sowie die Adresse des ersten assoziierten clusters.

2.4.2 Fragmentierung

Verstreute Speicherung von logisch zusammenhängenden Daten. Verhindern/ vermeiden durch: Defragmentierung (neu Positionieren), größere Blöcke, preallokation von Blöcken, spätes Festlegen von Blöcken.

2.5 Prozesse

2.5.1 Prozesslebenszyklus



2.5.2 Prozess Begriffe

Programm Ausführbare Binärdatei, Sequenz von Instruktionen.

Prozess Instanz eines Programms, Programm in Ausführung

Thread Kleinste Auszuführende Einheit. Im gleichen Speicherbereich wie Prozess.

Zombie Beendeter Prozess mit einem Eintrag in der Prozesstabelle.

Dämon Prozess im Hintergrund ohne Benutzerinteraktion.

Verhungern Prozess bekommt nie die Ressourcen die er für eine weitere Ausführung benötigt.

2.5.3 Process Control Block (PCB)

Ein Eintrag in der Prozesstabelle. Status, program counter, stack pointer, signals, scheduling info, ...

2.5.4 Prozess Scheduling

SJF Kürzester zuerst

FCFS Wer zuerst da ist

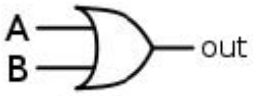
Round Robin jeder Prozess erhält immer wieder eine bestimmte Zeit

PSJF Preemtive Shortest Job First/aka SRTN: Shortest Remaining Time Next Wie SJF, aber laufende Prozesse können unterbrochen werden, wenn kürzere vorhanden sind.

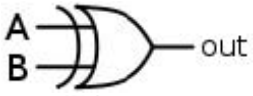
3 Schaltungen

3.1 Gatter

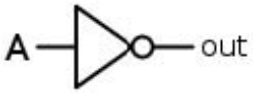
AND-Gatter



OR-Gatter



XOR-Gatter



NOT-Gatter

3.2 Karnaugh-Veitch - SOP - POS

KV → Trivial. Für SOP die einsen zusammen zählen. Für POS einfach SOP doppelt negieren und dann auflösen.

4 Assembler

4.1 Begriffe

Instruction Set Menge aller von der Hardware bereitgestellten Instruktionen, Datentypen, Datenstrukturen.

Mnemonic Für den Menschen lesbare Repräsentation eines Opcodes.

Opcode operation code. Spezifiziert eine Operation in Maschinensprache.

Pseudo-Opcode Ein opcode, der kein Äquivalent in der Maschinensprache hat, sondern in mehrere Operationen expandiert wird.

Heap Gesamter reservierter Speicherbereich eines Prozesses. Benötigt zur ausführung

Stack Speicherbereich in dem Daten LIFO (last-in-first-out) eingefügt oder entfernt werden. Nötig, um Funktionen/Subroutinen zu erlauben; Rücksprungadresse, Lokale Daten, Parameter, Rückgabewerte

4.2 Variablen deklarieren

```
1 dseg segment
2     varName slong       // byte: 8bit, sword: 16bit, slong: 32bit
3 dseg ends
```

4.3 Arithmetische Operatoren

add r,a,b: r← a+b	**sub** r,a,b: r←a-b	**mul** r,a,b: r←a*b
div r,a,b: r← a/b	**mod** r,a,b: r←a mod b	**neg** r,a: r←-a
sign r,a: r← {-1, 0, +1}	**abs** r,a,b: r←$\mid a \mid$	

4.4 Logische Operatoren

mov r,a: r← a	**and** r,a,b: r← $a \wedge b$	**or** r,a,b: r← $a \vee b$
xor r,a,b: r← ...	**not** r,a: r← ...	**cmnt** r,a: r← $\neg a$
shr r,a,x: r← a $\gg$ x	**shl** r,a,x: r← a $\ll$ x	

4.5 Bedingungen und Jumps

jmp label: Sprung zu..	**brcmp** O,label,a,b: Sprung zu ..., wenn a O b
cmp O,r,a,b: r← a O b	

5 Netzwerke

5.1 Ent-to-End Prinzip

Kommunikations-Operationen sollten so nah wie möglich an den Endpunkten stattfinden, dumme Netzwerke, intelligente Endsysteme.

5.2 5-Layer-Referenzmodell

Layer	Aufgabe	Protokolle	Geräte
Application	Kommunikation der Programme untereinander	(s)FTP, HTTP, SMTP, ..	
Transport	Sorgt für die Verbindung zwischen den Anwendungen (Host-2-Host über Ports)	TCP, UDP, ..	Firewall
Network	Routing und Übermittlung der Datenpakete	IP, IPX, ... Routing Protokolle, ..	Router
Link	1:1 Direktverbindung zwischen den Geräten	PPP, Ethernet, MAC	Bridge/Switch
Physical	Physisches Übertragungsmedium	Glasfaser, Wlan, Kabel, ...	Repeater, Hub

5.2.1 Unterschied zu OSI

Beim OSI Modell gibt es noch Presentation und Session zwischen Application und Transport Layer.

5.3 Zwei Generäle

Kurze Antwort: Nein. Keiner der beiden kann sich sicher sein, dass der andere die Nachrichten erhalten hat. Daher werden sie zögern und es kommt kein Angriff zu Stande. Problem: Es fehlt ein definiertes Kommunikationsprotokoll!

5.4 Protokolle

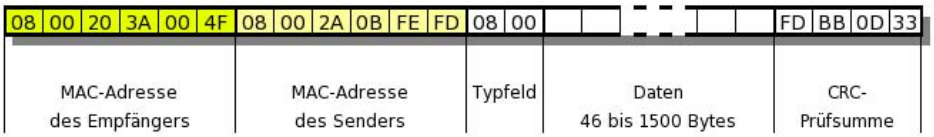
5.4.1 DNS

Über die URL wird die eigentliche IP Adresse des gewünschten Services erfragt. Der DNS Server besitzt eine Tabelle, um jede URL der IP Adressen zuweisen zu können und diese dem Clienten mitzuteilen. (UDP, erlaubt auch TCP)

5.4.2 DHCP

Automatischer Bezug von IP Adressen. Rechner haben also keine statische Adresse, sondern beziehen diese jedes Mal vom DHCP Server, der die Adressen verteilt. (Benutzt UDP)

5.4.3 Ethernet Header



5.4.4 IP

- IPv4
  - Besteht aus 4 Bytes
  - 127.0.0.1 Local Host
  - 192.168.x.x private Network
- IPv6
  - Besteht aus 16 Bytes! 2^{128} Möglichkeiten

5.4.5 TCP

Verbindungsaufbau:
$SYN \rightarrow$
$\leftarrow SYNACK$
$ACK \rightarrow$

5.4.6 HTTP

Das HTTP Protokoll ist Zustandslos, das bedeutet, dass Informationen aus früheren Anforderungen verloren gehen. Dies kann allerdings durchs das Speichern der Information in Cookies umgangen werden. HTTP ist für alle im Netzwerk lesbar, eine Verschlüsselung ist nur mit HTTPS möglich.

5.5 Netzwerkgeräte

Hub Sendet "dumm"Daten an das gesamte Netzwerk

Switch Sendet die Daten ïntelligentän einen bestimmten Empfänger addressiert. (MAC)

Router Routet.

Firewall Regelt Verkehr auf Transportebene

Bridge Wie Switch, jedoch zwischen unabhängigen Netzwerken

Repeater Verstärkt.

5.6 Delay

Es gibt ein „End-to-End“-Delay. Dies ist die Gesamtverzögerung zwischen Quelle und Ziel:
Berechnung: N * (dprocessing + dtransmission+ dpropagation), wobei N = Anzahl der Teilstrecken → Anzahl der Router + 1.

5.6.1 Processing Delay

Verarbeitungsverzögerung entsteht durch die Überprüfung der Packetheader, Entscheidung über den weiteren Weg des Paketes (output link), Bitfehler prüfen. → Bei Hochgeschwindigkeitsroutern liegt Verarbeitungsverzögerung bei wenigen Mikrosekunden.

5.6.2 queuing Delay

Warteschlangenverzögerung entsteht durch das Warten der Pakete auf der entsprechenden Leitung versendet zu werden. Die Dauer hängt von der Anzahl der bereits wartenden Pakete ab, d.h. Ist die Leitung leer, gibt es keine Warteschlangenverzögerung. → In der Praxis liegt sie meist bei Mikro- bis Millisekunden.
Berechnung: Länge des Paketes * Durchschnittliche Ankommensrate der Pakete / Bandbreite der Leitung

5.6.3 Transmission Delay

Übertragungsverzögerung ist die Zeitdauer, die der Router benötigt, um das Paket abzuschicken. → Liegt im Bereich von Mikro- und Millisekunden.
Berechnung: Länge des Paketes (bit) / Übertragungsgeschwindigkeit (bit/s)

5.6.4 Propagation Delay

Ausbreitungsgeschwindigkeit ist die Zeit, die ein Bit von einem Router zum nächsten benötigt. Sie ist eine Funktion der Entfernung zwischen den beiden Routern, hängt aber in keiner Weise von der Paketlänge und der Übertragungsrate der Leitung ab.
Berechnung: Entfernung (m) / Ausbreitungsgeschwindigkeit $(2 * 10^8 m/s)$

5.7 Packet-Switching, Circuit-Switching

Beim Cirguit-Switching werden die Daten an einem Stück gesendet, während sie beim Packet-Switching in kleinere Pakete geteilt werden, um einen kurzen Delay, bzw. eine bessere Auslastung zu erhalten.

Circuit-Switching:

\+ – keine Zwischenspeicherung nötig
– Packetübermittlung in richtiger Reihenfolge
– Information an einem Stick

\- – lange Warteschlangen
– Übertragungskanal die ganze Zeit über belegt

Packet-Switching:

\+ – kurzer Delay - gute Auslastung
– Übertragungsfehler werden schnell erkannt

\- – evtl. Überlastungserscheinungen
– alle Teilnehmer brauchen gleichen Netzwerkprotokoll
– Pakete können durcheinander ankommen

5.8 Multiplexing

Multiplexing wird bei konstanten Bandbreiten (z.B. circuit switching) und Statical Multiplexing bei variablen Bandbreiten (z.B. packet switching) verwendet.

- Beim Multiplexing (FDMA) erhält jeder Nutzer die ganze Zeit über nur einen Teil der Bandbreite (Frequenzen).
- Beim Statical Multiplexing (TDMA) erhält jeder Nutzer zwar die gesamte Breitbreite, allerdings immer nur für eine kurze Zeit.