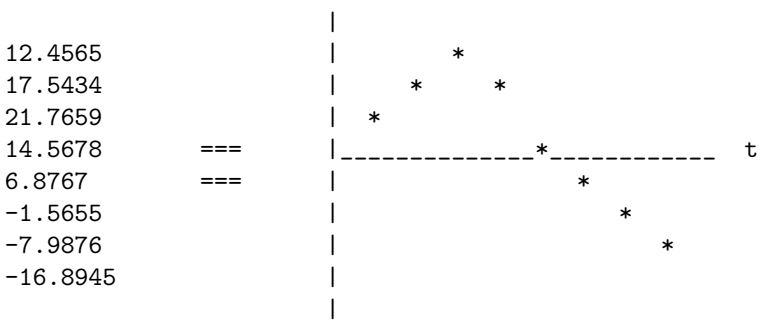
7 Wiederholung: Digitale Signale

In der praktischen wissenschaftlichen Arbeit geht es neben den Symbolen (Zeichen) hauptsächlich um gemessene und anschließend digitalisierte Signalverläufe. Ein *digitales Signal* ist zunächst einmal nur eine Reihe von diskreten (einzelnen) Werten, die als Zahlen hintereinander geschrieben werden (*Abtastwerte*).

Beispiel:



7.1 Abtastrate

Die *Abtastrate* oder *sampling rate* bestimmt, wieviele Werte pro Sekunde gemessen werden, und damit auch indirekt den Abstand zweier Abtastwerte auf der Zeitachse. Je schneller sich ein Signal mit der Zeit ändert, desto höher muß logischerweise auch die Abtastrate sein.

```
Sprache: 8000 - 48000 Hz
Musik:   44100 Hz
EMA:     200 Hz
```

7.2 Bitauflösung

Die *Wortbreite* oder *Bitauflösung* bestimmt, mit welcher *Genauigkeit* das Signal beschrieben wird. Allgemein üblich ist heutzutage eine Bitauflösung von 16 bit. D.h. das Signal kann Werte zwischen -32767 und +32767 annehmen. 16 Bit entspricht schon HiFi; für ein Telefon-Signal z.B. genügen 8 Bit (+/- 128). Ein Abtastwert wird auch häufig 'Wort' genannt; daher der Begriff 'Wortbreite'.

Ein Signal - meistens handelt es sich in unserem Institut um ein Sprachsignal - wird z.B. mit einem Mikrophon aufgenommen, d.h. die akustischen Schwingen in elektrische umgewandelt, und anschliessend in äquidistante Abtastwerte *digitalisiert*, die in einem File (Audio-, Sound-File) abgespeichert werden. Da es viel effektiver ist, Werte als sog. *binäre* Daten abzuspeichern, anstatt als lesbaren Text, können die meisten Audio-Files nicht mit dem Editor angeschaut werden. (Es gibt aber UNIX Befehle, die das machen, z.B. 'od'.)

In ihrem Kurs-Directory befinden sich mehrere Audio-Files mit einem Sprachsignal: `signal.raw signal.wav`

Es gibt im Prinzip drei Arten, digitale Signale zu betrachten:

- Als Folge von Abtastwerten (Zahlen)

  ```
  cip1 % od -w signal.raw | less
  ```

- Als ge-plottetes Signal auf dem Display

```
cip1 % wav2sfs signal.wav    (wandelt wav in sfs um)
cip1 % Es signal.sfs         (Öffnet Display und zeigt Signal)
```

- Als rekonstruiertes und damit hörbares Signal

```
cip1 % play signal.wav
```

Bei der Wiedergabe von Sprachsignalen sind sowohl die Abtastrate als auch das Format der einzelnen Abtastwerte entscheidend dafür, daß die Wiedergabe korrekt gelingt.

```
cip1 % hear_raw rate=8000 signal.raw
```

In diesem Befehl z.B. wird ein Signal, das mit 11025 Hz digitalisiert wurde, mit nur 8000 Hz wiedergegeben.

TIPP:
Falls Sie keine Audio-Ausgabe bekommen, gehen Sie in das 'K Menu' Ihrer Oberfläche und wählen Sie folgenden Befehl:

```
Applications / Multimedia / Volume Control / IPS audio preset
```

Prüfen Sie geg.falls auch, ob der Kopfhörer korrekt angeschlossen ist (an der mittleren (grünen) Buchse der Soundkarte (nicht beim eingebauten Sound-Anschluss!).

Übung:
Schliessen Sie ein Mikrophon an (Rosa Buchse der Soundkarte). Starten Sie ein einfaches Aufnahme-Tool:

```
cip1 % krecord &
```

Klicken Sie auf das Tool 'Input Level' und das Tool 'Freq Spectrum' in krecord. Regeln Sie im Mixer (kmix) mit dem Regler 'IGain' den Aufnahmepegel so, daß die Pegelbalken gerade nicht rot werden. (Wenn in Ihrem Mixer kein Regler 'IGain' vorhanden ist, reicht es auch in der Karte 'Switches' den Schalter 'Mic Boost' zu aktivieren.) Beobachten Sie das Spektrum des Signals für verschiedene Vokale des Deutschen. Nehmen Sie den Satz auf: 'The quick brown fox jumps over the lazy dog!' und speichern ihn im Soundfile 'dog.wav' ab.

7.3 Byteorder

Da Computer grundsätzlich ihre Daten in Blöcken von je 8 Bit (= 1 Byte) anordnen, gibt es bei Bitauflösungen grösser 8 Bit außerdem das Problem der *Byte-Reihenfolge* (*Byte-order*). Auf Maschinen mit Intel- und Digital-Prozessoren kommt das 'untere' Byte zuerst im File, danach das 'obere' Byte. Bei Motorola-, Sun- und vielen anderen Prozessoren ist es genau umgekehrt.

Beispiel:

```
                                          unteres  oberes    Byte

INTEL ('small endian', '01', 'low-high') :    10110100 00010110

MOTOROLA ('big endian', '10', high-low') :    00010110 10110100
```

Überträgt man nun ein Audio-File von einer Sun (10) auf einen Linux (01) und versucht es zu rekonstruieren, dann klingt das so (Vorsicht: Sehr laut!):

```
cip1 % hear_raw swap=yes test.raw
```

Das Vertauschen der Byte-Reihenfolge nennt man *swappen*. Daher haben viele Programme zum Abspielen von Audio-Files die Option, das Signal vor dem Abspielen zu swappen.

7.4 Technischer Umgang mit Audiodateien

7.4.1 Audio-Formate

Es gibt es eine Vielzahl verschiedener standardisierter Audio-File-Formate. Die Fileformat sind meistens - aber nicht immer - an ihren *Extensionen* (Teil des File-Namens nach dem '.') erkennbar. Die wichtigsten sind

.raw gar kein Format, d.h. die Abtastwerte stehen einfach hintereinander im File

.wav Microsoft-Format, Mono-Stereo-Mehrkanal, verschiedene Abtastraten möglich, byteorder nur '01'

.au SUN Audio, alles möglich

.aiff alles möglich

.nis NIST Format, lesbarer Header, alles möglich

.16 PhonDat, 16000 Hz (veraltet)

.al ALAW, europäisches Telefon-Format, 8 Bit, 8000 Hz, komprimiert

.ul ULAW, amerikanischen Telefon-Format, 8 Bit, 8000 Hz, komprimiert

.sfs SFS Archiv-File, kann beliebig viele synchnone Signale und Labels enthalten (s.u.)

7.4.2 Konvertierung von Formaten

Um ein Audiofile eines Formats in ein anderes umzuwandeln, verwendet man *Konvertierungs-Programme*. Das mächtigste dieser Programme ist 'sox' (siehe Man Page `man sox`). (TIPP: Das NIST format heißt in sox 'sph'.) Leider kennt sox kein SFS-Format. Daher gibt es noch die Programme `nist2sfs`, `wav2sfs` und `par2sfs` die oft gebraucht werden (siehe Man Pages).

Im folgenden Beispiel wird ein raw File mit 16kHz Abtastrate 2 bytes pro Abtastwert und der Kodierung 'signed integer' in ein Microsoft wav file gewandelt.

```
cip1 % sox -t raw -r 16000 -2 -s signal.raw signal.wav
cip1 % play signal.wav
```

Was stimmt hier nicht?

7.4.3 Programme zum Abhören

Voraussetzung für das Abhören ist natürlich, daß am Linux, an dem gearbeitet wird, entweder einen Kopfhörer oder eine Aktiv-Box angeschlossen ist. Wenn das nicht der Fall sein sollte: Kopfhörer oder Lautsprecher werden immer an der Rückseite der Audiokarte (erkennbar an drei oder vier Buchsen) an der Buchse für 'Speaker' angeschlossen.

Der Befehl `play` erkennt Audioformate an der Extension und versucht, diese abzuspielen. Wenn es nicht klappt, gibt es eine Diagnose ab, woran es liegen könnte. `play` ist eine Variante von `sox` und kennt die gleiche Syntax.

```
cip1 % play signal.wav
```

Als Header bezeichnet man einen Informationsblock zu Beginn eines Audio-Files. Die meisten Formate (mit Ausnahme von '.raw') haben solche Header, in welchen die Metadaten des Files verzeichnet sind, also z.B. die Abtastrate, die Bitbreite, das Sampleformat, etc.). Die meisten Header sind binär gespeichert und daher nur mit geeigneter Software lesbar. Einige wenige Formate haben lesbare Text-Header (z.B. NIST); um ein NIST file mit play abzuspielen, gibt man als Typ 'sph' an:

```
cip1 % play -t sph signal.nis
```

7.4.4 Programme zum Aufnehmen

Es gibt viele Programme. Sinnvoll sind 'krecord' (s.o.) und 'praat'. Ein reines Kommandozeilen-Tool ist 'rec'; dieses kann auch in Skripten verwendet werden.

Für beide Fälle (Abspielen und Abhören) gibt es ein *Mischpult-Programm* 'kmix' (oder 'kamix' oder 'alsamixer'), das es erlaubt, die Empfindlichkeit/Lautstärke sowie die Klangfarbe zu beeinflussen.

```
cip1 % kmix &
```

7.4.5 Programme zum Plotten

'Es' ist ein Programm des Pakets 'SFS'. Es kann mehrere Spuren, Sonagramm, andere Signal parallel auf dem Display anzeigen. Zuerst muss ein sog. SFS-Archiv-File (Extension '.sfs') angelegt werden, damit man damit arbeiten kann.

```
cip1 % wav2sfs signal.wav signal.sfs
cip1 % Es -i1.01 -g1.01 signal.sfs
```

'Praat' kann die gängigsten Standard-Soundformate lesen.