stellt bis zu 7 gültige Ziffern im Bereich $\mathcal{F} = [\min \mathcal{F}, \max \mathcal{F}]$ zur Verfügung; der Fehler für jede in $\mathcal{F}$ liegende reelle Zahl $x$ durch ihre Approximation trunc($x$) bzw. durch ihre Rundung round($x$) ist absolut

$$\begin{aligned} |x - \mathrm{trunc}(x)| &\leq 10^{-3} \\ |x - \mathrm{round}(x)| &\leq \frac{1}{2} \cdot 10^{-3}, \end{aligned}$$

also z.B. $|\frac{2}{3} - 0.666|$ bzw. $|\frac{2}{3} - 0.667|$. Dagegen kann der relative Fehler

$$\left| \frac{x - \mathrm{trunc}(x)}{x} \right| \text{ bzw. } \left| \frac{x - \mathrm{trunc}(x)}{\mathrm{trunc}(x)} \right|$$

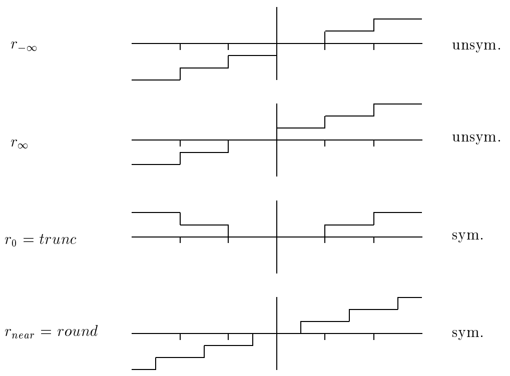
beliebig groß werden, es kann sogar passieren, daß keine Ziffer der Näherung stimmt! Für viele naturwissenschaftlichen Rechnungen reichen etwa obige 7 Stellen (7 *richtige* führende Stellen) aus, aber der abgedeckte Zahlenbereich erstreckt sich über viele Größenordnungen wie $\mu$V,...,kV, also $10^{-6}$V,...,$10^3$V, obwohl viele analoge Voltmeter nur $\pm 1\%$ Ablesegenauigkeit haben.

**Definition 1.3** *Eine Abbildung* $\circ : \mathbb{R} \to \mathcal{F}_n^\nu$ *mit den Eigenschaften*

$$\begin{aligned} \textit{Projektion :} \quad & \circ|_{\mathcal{F}_n^\nu} = id \\ \textit{Monotonie :} \quad & x \leq y \Leftarrow \circ x \leq \circ y \end{aligned}$$

*heißt* Rundung .

IEEE stellt 4 Rundungen zur Verfügung:






Sei $\tilde{x} \in \mathcal{F}_n^\nu$ eine Näherung von $x$, z.B. $\tilde{x} = \circ x$ mit $\circ \in \{r_{-\infty}, r_\infty, r_0, r_{near}\}$, so bezeichnet man

$$\Delta\tilde{x} := |\tilde{x} - x|$$

als *absoluten Fehler* des Näherungswertes $\tilde{x}$,

$$\bar{\varepsilon}_{\tilde{x}} := \left| \frac{\tilde{x} - x}{x} \right|$$

für $x \neq 0$ als *relativen Fehler* von $\tilde{x}$. Da $x$ häufig nicht bekannt ist, man also nur weiß, daß $\tilde{x}$ maximal $\Delta\tilde{x}$ von $x$ entfernt ist, benutzt man

$$\varepsilon_{\tilde{x}} := \left| \frac{\tilde{x} - x}{\tilde{x}} \right|$$

für $\tilde{x} \neq 0$ als relativen Fehler. Weiß man (z.B. bei $\tilde{x} = \circ x$ gerundet), daß $\tilde{x}$ genau zwischen zwei Rasterpunkten von $\mathcal{F}_3^4$ liegt, so kann man grob abschätzen:

$$\begin{aligned} \Delta\tilde{x} &\leq 10^{-3} \\ \bar{\varepsilon}_{\tilde{x}} &\leq \frac{10^{-3}}{x} \quad \forall x \neq 0 \\ \varepsilon_{\tilde{x}} &\leq \frac{10^{-3}}{\tilde{x}} \quad \forall \tilde{x} \neq 0 \end{aligned}$$

Ist die Art der Rundung nicht bekannt, ergibt sich also:

$$\begin{aligned} \tilde{x} = 0 \quad &\Rightarrow x \in (-0.001, 0.001) \Rightarrow \quad \left|\frac{\Delta\tilde{x}}{x}\right| \leq \infty \\ \tilde{x} = 0.001 \quad &\Rightarrow x \in (0.000, 0.002) \Rightarrow \quad \left|\frac{\Delta\tilde{x}}{x}\right| \leq \infty \\ \tilde{x} = 0.002 \quad &\Rightarrow x \in (0.001, 0.003) \Rightarrow \quad \left|\frac{\Delta\tilde{x}}{x}\right| \leq 1 \\ \tilde{x} = 0.003 \quad &\Rightarrow x \in (0.002, 0.004) \Rightarrow \quad \left|\frac{\Delta\tilde{x}}{x}\right| \leq 0.5 \\ &\vdots \\ \tilde{x} = 9999.998 \quad &\Rightarrow x \in (9999.997, 9999.999) \Rightarrow \quad \left|\frac{\Delta\tilde{x}}{x}\right| \leq 10^{-7} \\ \tilde{x} = 9999.999 \quad &\Rightarrow \ ??? \end{aligned}$$

Kennt man die Art der Rundung, so läßt sich genauer feststellen

1. Rundung zur Null:

$$\tilde{x} = 0 \quad \Rightarrow x \in (-0.001, 0.001) \Rightarrow \quad \left|\frac{\Delta\tilde{x}}{x}\right| \leq \infty$$



$$\begin{aligned} \tilde{x} = 0.001 \quad &\Rightarrow x \in [0.001, 0.002) \Rightarrow \quad \left|\frac{\Delta\tilde{x}}{x}\right| \leq 1 \\ \tilde{x} = 0.002 \quad &\Rightarrow x \in [0.002, 0.003) \Rightarrow \quad \left|\frac{\Delta\tilde{x}}{x}\right| \leq 0.5 \\ &\vdots \end{aligned}$$

Ein analoges Ergebnis gilt für $r_{+\infty}$ und $r_{-\infty}$.

2. $r_{near}$ : $\quad \Delta\tilde{x} = |x - \tilde{x}| \leq \frac{1}{2} \cdot 10^{-3}$

$$\begin{aligned} \tilde{x} = 0 \quad &\Rightarrow x \in [-0.0005, 0.0005) \Rightarrow \quad \left|\frac{\Delta\tilde{x}}{x}\right| \leq \infty \\ \tilde{x} = 0.001 \quad &\Rightarrow x \in [0.0005, 0.0015) \Rightarrow \quad \left|\frac{\Delta\tilde{x}}{x}\right| \leq 1 \\ \tilde{x} = 0.002 \quad &\Rightarrow x \in [0.0015, 0.0025) \Rightarrow \quad \left|\frac{\Delta\tilde{x}}{x}\right| \leq 0.33 \\ \tilde{x} = 0.003 \quad &\Rightarrow x \in [0.0025, 0.0035) \Rightarrow \quad \left|\frac{\Delta\tilde{x}}{x}\right| \leq 0.20 \\ &\vdots \\ \tilde{x} = 9999.998 \quad &\Rightarrow x \in [9999.9975, 9999.9985) \Rightarrow \quad \left|\frac{\Delta\tilde{x}}{x}\right| \leq \frac{1}{2} \cdot 10^{-7} \end{aligned}$$

Für naturwissenschaftliche/technische Anwendungen sind also etwa nur zwei führende Ziffern sinnvoll. Im Beispiel heißt das

$$(-1)^s \cdot \underbrace{(z_1 10^{-1} + z_2 10^{-2})}_{\text{norm. Mantisse}} \cdot \underbrace{10^e}_{\text{Basis}}$$

mit $s \in \{0,1\}, z_{1/2} \in \{0, \ldots, 9\}, z_1 \neq 0$ und einem Exponenten $e \in \{-6, -5, \ldots, 5\}$. Allgemein ergibt sich

$$(-1)^s \cdot \left( \sum_{i=1}^{N} z_i b^{-i} \right) \cdot b^e, \quad z_1 \neq 0 \qquad (\circ)$$

oder

$$(-1)^s \cdot \left( \sum_{i=0}^{N} z_i b^{-i} \right) \cdot b^e, \quad z_0 \neq 0$$

Zur Untersuchung des Zahlenbereichs der durch ($\circ$) gegebenen Zahlen sei

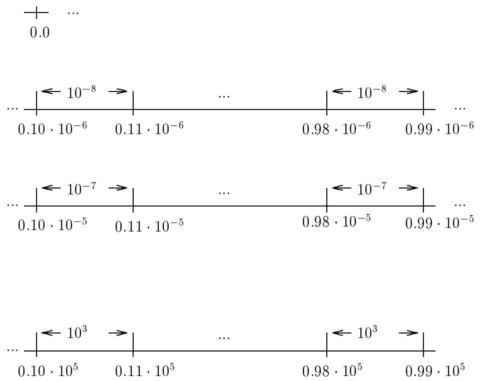
$$\tilde{\mathcal{G}} := \left\{ (-1)^s \cdot \left( \sum_{i=1}^{N} z_i b^{-i} \right) \cdot b^e | s \in \{0,1\}, z_i \in \{0, 1, \ldots, b-1\}, z_1 \neq 0, e \in \{e_{min}, \ldots, e_{max}\} \right\}$$



Man beachte, daß $\tilde{\mathcal{G}}$ <u>nicht</u> die 0 enthält! Man setzt daher $\mathcal{G} := \tilde{\mathcal{G}} \cup \{0\}$, $\max \mathcal{G} := \max \tilde{\mathcal{G}}$ und $\min \mathcal{G} := \min \tilde{\mathcal{G}}$. Für $\tilde{\mathcal{G}}$ erhält man z.B. mit $b = 10, N = 2$ und $e \in \{-6, \ldots, 5\}$:

$$\begin{aligned} \{-0.99 \cdot 10^5, -0.98 \cdot 10^5, \ \ldots \ &, -0.11 \cdot 10^5, -0.10 \cdot 10^5, \\ -0.99 \cdot 10^4, -0.98 \cdot 10^4, \ \ldots \ &, -0.11 \cdot 10^4, -0.10 \cdot 10^4, \\ \ldots \ & \\ -0.99 \cdot 10^{-6}, -0.98 \cdot 10^{-6}, \ \ldots \ &, -0.11 \cdot 10^{-6}, -0.10 \cdot 10^{-6}, \\ 0.10 \cdot 10^{-6}, 0.11 \cdot 10^{-6}, \ \ldots \ &, 0.98 \cdot 10^{-6}, 0.99 \cdot 10^{-6}, \\ 0.10 \cdot 10^{-5}, 0.11 \cdot 10^{-5}, \ \ldots \ &, 0.98 \cdot 10^{-5}, 0.99 \cdot 10^{-5}, \\ \ldots \ & \\ 0.10 \cdot 10^4, 0.11 \cdot 10^4, \ \ldots \ & 0.98 \cdot 10^4, 0.99 \cdot 10^4, \\ 0.10 \cdot 10^5, 0.11 \cdot 10^5, \ \ldots \ & 0.98 \cdot 10^5, 0.99 \cdot 10^5\} \end{aligned}$$

Für die Rundungsfehler ergibt sich im Beispiel mit $r_0$ folgendes Bild:



Eine Rechnung liefert für die relativen Fehler:

$$\tilde{x} = 0.0 \quad \Rightarrow x \in (-0.10 \cdot 10^{-6}, 0.10 \cdot 10^{-6}) \Rightarrow \quad \left|\frac{\tilde{x} - x}{x}\right| \leq \infty$$

$$\tilde{x} = 0.10 \cdot 10^{-6} \quad \Rightarrow x \in [0.10 \cdot 10^{-6}, 0.11 \cdot 10^{-6}) \Rightarrow \quad \left|\frac{\tilde{x} - x}{x}\right| \leq \frac{10^{-8}}{10^{-7}} = 0.1$$

$$\tilde{x} = 0.11 \cdot 10^{-6} \quad \Rightarrow x \in [0.11 \cdot 10^{-6}, 0.12 \cdot 10^{-6}) \Rightarrow \quad \left|\frac{\tilde{x} - x}{x}\right| \leq \frac{10^{-8}}{0.11 \cdot 10^{-6}} \leq 0.\overline{09}$$

$$\vdots$$

$$\tilde{x} = 0.99 \cdot 10^{-6} \quad \Rightarrow x \in [0.99 \cdot 10^{-6}, 1.00 \cdot 10^{-6}) \Rightarrow \quad \left|\frac{\tilde{x} - x}{x}\right| \leq \frac{10^{-8}}{0.99 \cdot 10^{-6}} \leq 0.\overline{01}\ldots$$

Also mit $p = N$:

$$\boxed{\Delta\tilde{x} \leq b^{exponent(\tilde{x})-p}}$$

Ist nun $\tilde{x} \in S(b, p, emin, emax) \setminus \{0.0\}$ und dabei ohne Einschränkung $\tilde{x} > 0$, so gilt bei der Rundung $r_0$ für $x \in [\tilde{x}, \tilde{x} + b^{-p+exponent(\tilde{x})})$

$$\begin{aligned}
\left|\frac{x - \tilde{x}}{x}\right| &\leq \left|\frac{\tilde{x}}{x} - 1\right| \\
&\leq \left|\frac{\tilde{x}}{\tilde{x} + b^{-p+exponent(\tilde{x})}} - 1\right| \\
&= \left|\frac{mantisse(\tilde{x})}{mantisse(\tilde{x}) + b^{-p}} - 1\right| \\
&= \left|\frac{b^{-p}}{mantisse(\tilde{x}) + b^{-p}}\right| \\
&= \left(\frac{mantisse(\tilde{x})}{b^{-p}} + 1\right)^{-1} \\
&\approx \frac{b^{-p}}{mantisse(\tilde{x})} \\
&\in \left(b^{-p}, b^{1-p}\right],
\end{aligned}$$

wobei benutzt wurde, daß für große $\alpha$ gilt: $\frac{1}{1+\alpha} \approx \frac{1}{\alpha}$. Bei der Rundungsart $r_{near}$ erhält man folgende Abschätzungen:

$$\tilde{x} = 0.0 \qquad \Rightarrow \quad ?$$

$$\tilde{x} = 0.10 \cdot 10^{-6} \qquad \Rightarrow \quad ?$$

$$\tilde{x} = 0.11 \cdot 10^{-6} \quad \Rightarrow x \in [0.105 \cdot 10^{-6}, 0.115 \cdot 10^{-6}) \Rightarrow \quad \left|\frac{\tilde{x} - x}{x}\right| \leq \frac{\frac{1}{2} \cdot 10^{-8}}{0.105 \cdot 10^{-6}} \leq 0.0476...$$

$$\vdots$$



$$\tilde{x} = 0.99 \cdot 10^{-6} \quad \Rightarrow x \in [0.985 \cdot 10^{-6}, 0.995 \cdot 10^{-6}) \Rightarrow \quad \left|\frac{\tilde{x} - x}{x}\right| \leq \frac{\frac{1}{2} \cdot 10^{-8}}{0.985 \cdot 10^{-6}} \leq 0.005076...$$

$$\vdots$$

$$\tilde{x} = 0.99 \cdot 10^{5} \qquad \Rightarrow \quad ?$$

Für $r_{near}$ gilt also:

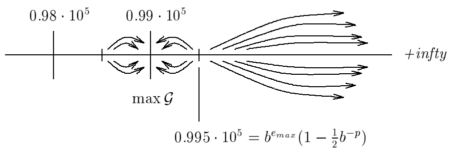
$$\boxed{\Delta\tilde{x} \leq \frac{1}{2} \cdot 10^{exponent(\tilde{x})-p}}$$

Damit die Formel von $r_{near}$ auch für $\tilde{x} = 0.99 \cdot 10^5$ gilt, muß

$$x \in [0.985 \cdot 10^5, 0.995 \cdot 10^5)$$

auf $\tilde{x}$ gerundet werden; alle $x \geq 0.995 \cdot 10^5$ werden zu $+infty$.

$$\mathcal{G}(b, p, e_{min}, e_{max}) = \mathcal{G}(10, 2, -6, 5)$$




Nahe 0 arbeiten *alte* Arithmetiken mit

$$\tilde{x} = 0.10 \cdot 10^{-6} \quad \Rightarrow x \in [0.10 \cdot 10^{-6}, 0.15 \cdot 10^{-6}),$$

d.h. die Formel gilt auch hier, könnte jedoch noch verschärft werden, und

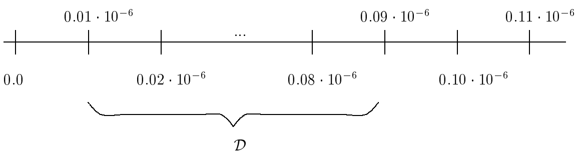
$$\tilde{x} = 0.0 \quad \Rightarrow x \in (-0.10 \cdot 10^{-6}, +0.10 \cdot 10^{-6}) \Rightarrow \left\{ \begin{array}{lcl} \Delta\tilde{x} & = & 2 \cdot 0.10 \cdot 10^{-6} \\ \left|\frac{\tilde{x}-x}{x}\right| & < & \infty \end{array} \right.$$

Es ergibt sich folgendes **Problem**: Die Äquivalenz

$$\boxed{x - y > 0 \quad \Leftrightarrow \quad x > y}$$

gilt im allgemeinen hier nicht mehr. So wird z.B. die Differenz $0.11 \cdot 10^{-6} - 0.10 \cdot 10^{-6}$ nach 0 gerundet, weil $10^{-8} < 0.10 \cdot 10^{-6} = minReal$ ist, aber natürlich ist $0.11 \cdot 10^{-6} > 0.10 \cdot 10^{-6}$. Da Festpunktsysteme die Addition exakt ausführen, ergänzen wir $\mathcal{G}(b, p, e_{min}, e_{max})$ um Null herum um die Rasterpunkte des Festkommasystems mit $\Delta = b^{e_{min}-p}$, d.h. im Beispiel $\Delta = b^{-8}$.






Damit ergeben sich in $\mathcal{D}$ folgende Fehlerformeln für $\tilde{x} \in \mathcal{D} \setminus \{0.0\}$: Bei der Rundungsart $r_0$ ist $\Delta\tilde{x} = b^{minexp}$ und somit mit $\tilde{mant}(\tilde{x}) = \frac{\tilde{x}}{b^{minexp}}$

$$\begin{aligned}
\left|\frac{\tilde{x} - x}{\tilde{x}}\right| &\leq \frac{b^{minexp-p}}{\tilde{mant}(\tilde{x}) \cdot b^{minexp}} = \frac{b^{-p}}{\tilde{mant}(\tilde{x})} \\
\left|\frac{\tilde{x} - x}{x}\right| &\leq \left(\frac{\tilde{mant}(\tilde{x})}{b^{-p}} + 1\right)^{-1}.
\end{aligned}$$

Für $r_{near}$ ist $\Delta\tilde{x} = \frac{1}{2}b^{minexp-p}$ und daher

$$\left|\frac{\tilde{x} - x}{x}\right| \leq \frac{1}{2}\frac{b^{-p}}{\tilde{mant}(\tilde{x}) - \frac{1}{2}b^{-p}}$$

**Aufgabe 1.5** *Berechne für* $0.01 \cdot 10^{-6}$ *die Abschätzungen für* $r_0$ *und* $r_{near}$.

Hochgenaue Standardfunktionen können i.a. nur auf drei benachbarte Gitterpunkte genau berechnet werden:

$$\frac{1}{3} \cdot 3 \in [0.99...9, 1.00...01]$$

Für ein (unbekanntes) Funktionsergebnis $x$ und die gelieferte Näherung $\tilde{x}$ gilt dann also z.B. bei der Rundung $r_0$:

$$\begin{aligned}
\tilde{x} = 0.15 \cdot 10^{-6} \quad &\Rightarrow \quad x \in [0.15 \cdot 10^{-6}, 0.17 \cdot 10^{-6}] \\
&\Rightarrow \quad \left|\frac{x - \tilde{x}}{x}\right| \leq \left|\frac{mant(\tilde{x})}{mant(\tilde{x}) + 2 \cdot 10^{-2}} - 1\right| = \left(\frac{mant(\tilde{x})}{2 \cdot 10^{-2}} + 1\right)^{-1} \\
&\in \quad [0.0198..., 0.166...]
\end{aligned}$$

für Argumente im normalisierten Bereich.

**Aufgabe 1.6** *Wie sieht das im unnormierten Bereich aus?*



#### 1.2.2.12 IEEE-Gleitkommazahlen

IEEE-Gleitkommazahlen im 64 Bitformat sind von der Form

$$x_d = s \cdot 2^e \cdot \sum_{k=1}^{53} f_k \cdot 2^{-k} \tag{1.1}$$

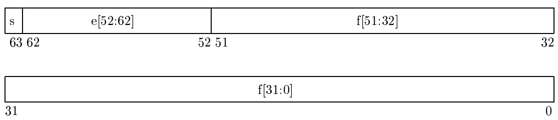
mit genau 53 signifikanten Dualstellen, wobei $s = \pm 1, -1021 \leq e \leq 1024$ und $f_i \in \{0, 1\}$ mit $f_1 \neq 0$ oder

$$x_d = s \cdot 2^{-1021} \cdot \sum_{k=2}^{53} f_k \cdot 2^{-k} \tag{1.2}$$

(inklusive der 0). Zusätzlich sind folgende drei Ausdrücke vordefiniert:

| | |
|---|---|
| $NaN$ | not a number |
| $+infty$ | $+\infty$ |
| $-infty$ | $-\infty$ |

Man beachte, daß sich die Zahl $\frac{1}{3}$ nicht exakt im IEEE-Format darstellen läßt. Die Darstellung einer Zahl im IEEE-Format erfolgt dabei durch das folgende Bitmuster:




Werte werden dabei entsprechend der folgenden Tabelle angenommen:

| **Double-Format Bitmuster** | **Wert** |
|---|---|
| $0 < e < 2047$ | $(-1)^s \cdot 2^{e-1023} \cdot 1.f$ (normal numbers) |
| $e = 0, f \neq 0$ | $(-1)^s \cdot 2^{-1022} \cdot 0.f$ (subnormal numbers) |
| (mindestens ein Bit in $f$ ist ungleich 0) | |
| $e = 0, f = 0$ | $(-1)^s \cdot 0.0$ (signed zero) |
| (alle Bits in $f$ sind 0) | |
| $s = 0, e = 2047, f = .000 \cdots 0$ | +INF (positive infinity) |
| (alle Bits in $f$ sind 0) | |
| $s = 1, e = 2047, f = .000 \cdots 0$ | -INF (negative infinity) |
| (alle Bits in $f$ sind 0) | |
| $s = u, e = 2047, f \neq 0$ | NaN (not a number) |
| (mindestens ein Bit in $f$ ist ungleich 0) | |

- **Maximum positive normal number** — die größte im IEEE Double-Format darstellbare endliche Zahl
- **Minimum positive normal number** — die kleinste im IEEE Double-Format darstellbare positive Zahl, bei der sicher ist, daß nur die letzte Binärstelle unsicher ist.
- **Maximum subnormal number** — die größte im IEEE Double-Format darstellbare Zahl, bei der es möglich ist, daß mehr als die letzte Binärstelle unsicher ist.
- **Minimum positive subnormal number** — die kleinste im IEEE Double-Format darstellbare positive Zahl; hier ist es möglich, daß mehr als die letzte Binärstelle unsicher ist.

Es ergeben sich dabei folgende Beispiele:

| Üblicher Name | Bitmuster (Hex) | Äquivalenter Wert |
|---|---|---|
| $+0$ | `00000000 00000000` | 0.0 |
| $-0$ | `80000000 00000000` | $-0.0$ |
| 1 | `3ff00000 00000000` | 1.0 |
| 2 | `40000000 00000000` | 2.0 |
| max normal number | `7fefffff ffffffff` | 1.7976931348623157e+308 |
| min positve normal number | `00100000 00000000` | 2.2250738585072014e−308 |
| max subnormal number | `000fffff ffffffff` | 2.2250738585072009e−308 |
| min positve subnormal number | `00000000 00000001` | 4.9406564584124654e−324 |
| $+\infty$ | `7ff00000 00000000` | Infinity |
| $-\infty$ | `fff00000 00000000` | −Infinity |
| Not-a-Number | `7ff80000 00000000` | NaN |

Im IEEE 32 Bitformat (single precision) gilt analog:



| **Gleitkomma Single-Precision Format** | |
|---|---|
| $s$ = sign (1 bit)<br>$e$ = biased exponent (8 bits)<br>$f$ = fraction (23 bits)<br>$u$ = undefined | |
| Normalized value ($0 < e < 255$) | $(-1)^s \cdot 2^{e-127} \cdot 1.f$ |
| Subnormal value ($e = 0$) | $(-1)^s \cdot 2^{-126} \cdot 0.f$ |
| Zero ($e = 0$) | $(-1)^s \cdot 0$ |
| Signalling NaN | $s = u, e = 255\,(\max)\; f = .0uu \cdots uu$ (mindenstens ein Bit in $f \neq 0$) |
| Quiet NaN | $s = u, e = 255\,(\max)\; f = .1uu \cdots uu$ |
| $-\infty$ | $s = 1, e = 255\,(\max)\; f = .000 \cdots 00$ |
| $+\infty$ | $s = 0, e = 255\,(\max)\; f = .000 \cdots 00$ |

Dazu noch einmal vergleichend das 64 Bitformat

| **Gleitkomma Double-Precision Format** | |
|---|---|
| $s$ = sign (1 bit)<br>$e$ = biased exponent (11 bits)<br>$f$ = fraction (52 bits)<br>$u$ = undefined | |
| Normalized value ($0 < e < 2047$) | $(-1)^s \cdot 2^{e-1023} \cdot 1.f$ |
| Subnormal value ($e = 0$) | $(-1)^s \cdot 2^{-1022} \cdot 0.f$ |
| Zero ($e = 0$) | $(-1)^s \cdot 0$ |
| Signalling NaN | $s = u, e = 2047\,(\max)\; f = .0uu \cdots uu$ (mindenstens ein Bit in $f \neq 0$) |
| Quiet NaN | $s = u, e = 2047\,(\max)\; f = .1uu \cdots uu$ |
| $-\infty$ | $s = 1, e = 2047\,(\max)\; f = .000 \cdots 00$ |
| $+\infty$ | $s = 0, e = 2047\,(\max)\; f = .000 \cdots 00$ |

und das 128 Bitformat:

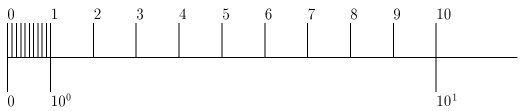


| **Gleitkomma Quad-Precision Format** | |
|---|---|
| $s$ = sign (1 bit)<br>$e$ = biased exponent (15 bits)<br>$f$ = fraction (112 bits)<br>$u$ = undefined | |
| Normalized value ($0 < e < 32767$) | $(-1)^s \cdot 2^{e-16383} \cdot 1.f$ |
| Subnormal value ($e = 0$) | $(-1)^s \cdot 2^{-16382} \cdot 0.f$ |
| Zero ($e = 0$) | $(-1)^s \cdot 0$ |
| Signalling NaN | $s = u, e = 32767\,(\max)\; f = .0uu \cdots uu$ (mindenstens ein Bit in $f \neq 0$) |
| Quiet NaN | $s = u, e = 32767\,(\max)\; f = .1uu \cdots uu$ |
| $-\infty$ | $s = 1, e = 32767\,(\max)\; f = .000 \cdots 00$ |
| $+\infty$ | $s = 0, e = 32767\,(\max)\; f = .000 \cdots 00$ |

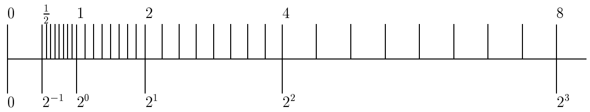


Zur Veranschaulichung der absoluten Abstände in Gleitkomma-Zahlensystemen:

**Decimal Representation**




**Binary Representation**




Die Lücken zwischen darstellbaren Single-Precision Format Gleitkommazahlen:

| **x** | **Nachbarzahl von $x$ in Richtung $+\infty$** | **Lücke** |
|---|---|---|
| 0.0 | 1.4012985e-45 | 1.4012985e-45 |
| 1.1754944e-38 | 1.1754945e-38 | 1.4012985e-45 |
| 1.0 | 1.0000001 | 1.1920929e-07 |
| 2.0 | 2.0000002 | 2.3841858e-07 |
| 16.000000 | 16.000002 | 1.9073486e-06 |
| 128.00000 | 128.00002 | 1.5258789e-05 |
| 1.0000000e+20 | 1.0000001e+20 | 8.7960930e+12 |
| 9.9999997e+37 | 1.0000001e+38 | 1.0141205e+31 |

Die üblichen Gleitkomma-Zahlensysteme im Vergleich:

| Wertebereich und Genauigkeit | | | | |
|---|---|---|---|---|
| **Format** | **Signifikante Binärstellen** | **min positiv normal number** | **max normal number** | **Signifikante Dezimalstellen** |
| single | 24 | $1.175\ldots 10^{-38}$ | $3.402\ldots 10^{+38}$ | 6-9 |
| double | 53 | $2.225\ldots 10^{-308}$ | $1.797\ldots 10^{+308}$ | 15-17 |
| double extended (SPARC) | 113 | $3.362\ldots 10^{-4932}$ | $1.189\ldots 10^{+4932}$ | 33-36 |
| double extended (x86) | 64 | $3.362\ldots 10^{-4932}$ | $1.189\ldots 10^{+4932}$ | 18-21 |

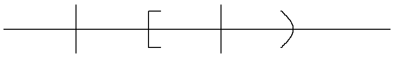
| **Format** | **Underflow-Schwelle** | |
|---|---|---|
| Single | Smallest Normal Number | 1.17549435e-38 |
| | Largest Subnormal Number | 1.17549421e-38 |
| Double | Smallest Normal Number | 2.2250738585072014e-308 |
| | Largest Subnormal Number | 2.2250738585072009e-308 |
| Double extended (SPARC) | Smallest Normal Number | 3.362103143112093506262677817321752 6e-4932 |
| | Largest Subnormal Number | 3.3621031431120935062626778173217520e-4932 |
| Double extended (x86) | Smallest Normal Number | 3.36210314311209350626e-4932 |
| | Largest Subnormal Number | 3.36210314311209350590e-4932 |

IEEE double precision entspricht dem Zahlenbereich $S(2, 53, -1021, +1024)$. Damit ist die größte darstellbare Zahl

$$maxReal = \sum_{k=1}^{53} 2^{-k} \cdot 2^{1024} = (1 - 2^{-53}) \cdot 2^{1024} = (1 - 2^{-p}) \cdot 2^{emax}$$

und für $[maxReal, +\infty)$ ergibt sich bei der Rundungsart $r_{near}$ und $b = 2$:

maxReal



maxReal $-2^{1024-53}$ maxReal + $\frac{1}{2}b^{emax-p} = b^{emax}(1 - \frac{1}{2}b^{-p})$



**Beispiele für gute IEEE-Algorithmen**

1. **Invarianz von Eigenschaften in $\mathbb{R}$ auch in `real`:**

```
function arithMittel(x,y : real) : real; {IEEE}
begin
      if sign(x)=sign(y) then
            arithMittel := x + (y-x)/2.0
      else
            arithMittel := (x+y)/2.0
end; {arithMittel}
```

Dieser Algorithmus weist folgende Eigenschaften auf, wobei ohne Einschränkung davon ausgegangen werden kann, daß $x \le y$ gilt:

(a) Fehler $\le 1\, ulp$, wobei $ulp$ für *unit in last place* steht

(b) Allgemein gilt $x \le arithMittel \le y$; falls zwischen $x$ und $y$ ein weiterer Gitterpunkt liegt, so gilt $x < arithMittel < y$.

(c) Es tritt in keinem Fall Over- oder Underflow auf.

Die angegebene zweite Eigenschaft des Algorithmus ist keineswegs selbstverständlich. Es gibt nämlich $x, y; x \ne y$ mit:

$$\frac{x \oplus y}{2} \notin [\min(x, y), \max(x, y)]$$

Man betrachte etwa im 13-stelligen Dezimalsystem die Zahlen

$$\begin{aligned} x &= 0.9\cdots 96 \cdot 10^0 \\ y &= 0.9\cdots 98 \cdot 10^0 \end{aligned}$$

Dann ist $x \oplus y = 0.19\cdots 9 \cdot 10^1$ und somit

$$\frac{x \oplus y}{2} = 0.9\cdots 95 \cdot 10^0.$$

Es wird nun der Zahlenbereich $S(10, 2, -5, +5)$ mit $S' := S \cup \{Underflowbereich\}$ und die Operation

$$\frac{x}{2} + \frac{y}{2}$$

behandelt.

(a) Für

$$x = 0.13 \cdot 10^{-5}, \quad y = 0.50 \cdot 10^{-4}$$



erhält man:

$$\begin{aligned} \frac{x}{2} &= 0.065 \cdot 10^{-5} \overset{in\,S'}{=} \begin{cases} 0.07 \cdot 10^{-5} & \text{bei grad. Underflow} \\ 0.0 & \text{trad.} \end{cases} \\ \frac{y}{2} &= 0.25 \cdot 10^{-4} \\ \frac{x}{2} + \frac{y}{2} &= \begin{cases} 0.257 \cdot 10^{-4} \overset{in\,S'}{=} 0.26 \cdot 10^{-4} & \\ 0.25 \cdot 10^{-4} & \text{trad.} \end{cases} \end{aligned}$$

(b) Für

$$x = 0.13 \cdot 10^{-5}, \quad y = 0.10 \cdot 10^{-5}$$

erhält man:

$$\begin{aligned} \frac{x}{2} &\overset{in\,S'}{=} 0.07 \cdot 10^{-5} \\ \frac{y}{2} &\overset{in\,S'}{=} 0.05 \cdot 10^{-5} \\ \frac{x}{2} + \frac{y}{2} &\overset{in\,S'}{=} 0.12 \cdot 10^{-5} \end{aligned}$$

(c) Sind

$$x = 0.13 \cdot 10^{-5}, \quad y = 0.11 \cdot 10^{-5},$$

so erhält man bei trad. Rechnung $\frac{x}{2} + \frac{y}{2} = 0$ und $x + \frac{y-x}{2} = x$. In $S'$ ist jedoch

$$\begin{aligned} \frac{x}{2} &\overset{in\,S'}{=} 0.07 \cdot 10^{-5} \\ \frac{y}{2} &\overset{in\,S'}{=} 0.06 \cdot 10^{-5} \\ \frac{x}{2} + \frac{y}{2} &\overset{in\,S'}{=} 0.13 \cdot 10^{-5} \end{aligned}$$

während

$$\begin{aligned} y - x &\overset{in\,S'}{=} -0.02 \cdot 10^{-5} \\ (y - x)/2 &\overset{in\,S'}{=} -0.01 \cdot 10^{-5} \\ x + (y - x)/2 &\overset{in\,S'}{=} 0.12 \cdot 10^{-5}. \end{aligned}$$

2. **Monotonie** (Voraussetzung für Nullstellensuchverfahren, z.B. Bijektion,...)
Der Graph von

$$p(x) = 170.4x^3 - 356.41x^2 + 168.97x + 18.601$$

liegt in $[1.09160791, 1.09160805] \times [-2 \cdot 10^{-13}, 5 \cdot 10^{-13}]$.



In beiden Fällen wurde mit 16-stelliger Arithmetik gerechnet bzw. auf ebenso viele Stellen (automatisch) verifiziert. Die beiden positiven Nullstellen des obigen Polynoms liegen sehr nahe zusammen, wie die Zerlegung in Linearfaktoren

$$p(x) = 170.4 \cdot \left(x - \frac{5 + \sqrt{35}}{10}\right) \left(x - \frac{5 - \sqrt{35}}{10}\right) \left(x - \frac{18601}{17040}\right)$$

und die unter Verwendung von ADE berechneten Einschließungen

$$\frac{5 + \sqrt{35}}{10} = 1.09160797830996^2_1 \quad \text{und} \quad \frac{18601}{17040} = 1.09160798122065^8_7$$

zeigen.

3. **Lösung einer quadratischen Gleichung (Auslöschung):**
Es wird der Rundungsfehlereinfluß bei 4-stelliger Arithmetik, also bei Gleitkommazahlen mit 4 Mantissenstellen betrachtet. Wir wollen das wohlbekannte Problem

$$ax^2 + bx + c = 0.0$$

nach der Lösungsformel $x_{1/2} = (-b \pm \sqrt{b^2 - 4.0 \cdot a \cdot c})/(2.0 \cdot a)$ mit 4-stelliger Arithmetik (d.h. alle Zwischenergebnisse haben nur 4 signifikante Stellen!) lösen für

$$a = +1.0004, \quad b = -200.01, \quad c = +1.0003 \tag{1.3}$$

<u>1. Fehler:</u> Representation von $a, b, c$ nur vierstellig, also

$$\bar{a} = +1.000, \quad \bar{b} = -200.0, \quad \bar{c} = 1.000$$

Die Rechnung ergibt im 1. Schritt den <u>2. Fehler</u>, nämlich einen Rundungsfehler bei der Subtraktion:

$$\bar{b}^2 - 4.0 \cdot \bar{a} \cdot \bar{c} = 40000 - 4.000 = 40000$$

Der 2. Schritt liefert

$$\bar{x}_1 = (200.0 + 200.0)/2.0 = 200.0$$

und im 3. Schritt ergibt sich durch einen <u>Folgefehler</u> des 2. Fehlers

$$\bar{x}_2 = (200.0 - 200.0)/2.0 = 0.0$$

Die exakten Lösungen lauten jedoch auf 4 signifikante Ziffern genau

$$\begin{aligned} \bar{x}_1 &= 200.0 \\ \bar{x}_2 &= 0.005000 \end{aligned}$$

Die exakten Lösungen des gestellten Problems (1.3) wären auf 4 Ziffern genau

$$\begin{aligned} x_1 &= 200.0 \\ x_2 &= 0.004999 \end{aligned}$$

gewesen.
Als stabilere Lösungsidee bietet sich für $a \neq 0$ nach der Umformung

$$ax^2 + bx + c = 0 \quad \Leftrightarrow \quad x_{1/2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

die vorzeichengünstige Wurzel

$$x_1 := \frac{-b + \mathrm{sign}(-b)\sqrt{b^2 - 4ac}}{2a}$$

an. Der entscheidende Vorteil ist, daß die Summanden im Zähler das gleiche Vorzeichen besitzen. Nach dem Satz von Vieta ist dann die zweite Wurzel gegeben durch:

$$x_2 := \frac{c}{a \cdot x_1},$$

wegen $x_1 x_2 = \frac{c}{a}$ und $x_1 + x_2 = -\frac{b}{a}$.

4. **Unvergleichbare IEEE-Zahlen:**
Man betrachte den folgenden Programmausschnitt (Fraktale/Apfelmännchen);

```
⋮
y1 = 0 ;
y2 = 0 ;
for (i=0;i<ITERATIONS;i++)
{
      x1 = y1 ;
      x2 = y2 ;
      y1 = x1*x1-x2*x2+c1 ;
      y2 = 2.0*x1*x2+c2 ;
      n2 = y1*y1+y2*y2 ; /* Norm ↑ 2 */
      if (n2>4.0) then
         /* skip */
      else plot(y1,y2) ;
}
```

Geplotten wird, wenn $\|y\|_2 \leq 2.0$, *aber:* Falls $y_i$ sehr groß (nahe $+\infty$), so wird wegen $y_1 = +\infty - \infty + c = NaN$ ebenfalls geplottet! Besser wäre also:



```
if (n2>4.0) then
      ...
else if unordered(n2,4.0) then
      ...
else ...
```

Beachte also:

```
if (a>b) then
      ...
else {a ≤ b ∨ unordered(a,b)}
      ...
```

**Extremsituationen**

Ausnahmebedingungen:

| IEEE-Ausnahmebedingungen | Grund des Auftretens | Beispiel | Ergebnisse ohne Exception-Header |
|---|---|---|---|
| Ungültige Operation | Die Operation ist für den auftretenden Operanden nicht definiert.<br>Diese Ausnahmebedingung kann beim x86 auch bei Stack-Fehlern auftreten | $\sqrt{(-x)}$ für $x > 0$<br>fp_op (signaling_NaN)<br>$0 \cdot \infty, 0/0, \infty/\infty, x$ rem 0<br>ungeordneter Vergleich<br>unerlaubte Umwandlung | Quiet NaN |
| Division durch Null | Der Divisor ist gleich Null und der Dividend ist eine endliche Zahl ungleich Null. Allgemeiner: ein exaktes unendliches Ergebnis wird durch eine Operation auf endlichen Operanden geliefert. | $x/0$ für $x \neq 0, \infty$ oder NaN<br>log(0) | $\infty$ mit richtigem Vorzeichen |



Tabellenfortsetzung

| | | | |
|---|---|---|---|
| Überlauf | Ein korrekt gerundetes Ergebnis ist größer als die größte darstellbare Zahl, d.h. der Bereich des Exponenten ist erschöpft. | *Double precision:*<br>MAXDOUBLE+1.0e294<br>exp(709.8)<br>*Single precision:*<br>(float)MAXDOUBLE<br>MAXFLOAT+1.0e32<br>expf(88.8) | Abhängig von der Rundungsart (RM) und dem Vorzeichen des Zwischenergebnisses<br>RM \| + \| −<br>RN \| $+\infty$ \| $-\infty$<br>RZ \| + max \| − max<br>R− \| + max \| $-\infty$<br>R+ \| $+\infty$ \| − max |
| Unterlauf | Tritt auf, wenn das exakte Ergebnis von Null verschieden ist, liegt stets im zwischen $\pm 2^{e_{min}}$ | *Double precision:*<br>nextafter(min_normal,$-\infty$)<br>nextafter(min_subnormal,$-\infty$)<br>MINDOUBLE/3.0<br>exp($-708.5$)<br>*Single precision:*<br>(float)MINDOUBLE<br>nextafterf(MINFLOAT,$-\infty$)<br>expf(-87.4) | Subnormal oder Null |
| Ungenauigkeit | Das gerundete Ergebnis stimmt nicht mit dem exakten Ergebnis überein. | 2.0/3.0<br>(float)1.12345678<br>log(1.1)<br>MAXDOUBLE+MAXDOUBLE, falls kein Überlauf auftritt | Ergebnis der Operation (gerundet), Unter- oder Überlauf |
| Unnormalisierter Operand (nur bei x86) | Versuch, eine arithmetische Operation mit unnormalen Operanden durchzuführen | 1.0+MINDOUBLE/2.0 | Ergebnis der Operation (gerundet), Unter- oder Überlauf |



**IEEE Funktionen:**

| Funktion | Beschreibung |
|---|---|
| `math.h` | header file |
| `copysign(x,y)` | $x$ mit $y$'s Vorzeichen |
| `fabs(x)` | Absolutbetrag von $x$ |
| `fmod(x,y)` | Rest bei Division von $x$ durch $y$ |
| `ilogb(x)` | Exponent zur Basis 2 von $x$ |
| `nextafter(x,y)` | Nächste darstellbare Zahl von $x$ in Richtung $y$ |
| `remainder(x,y)` | Rest bei Division von $x$ durch $y$ |
| `scalbn(x,n)` | $x \times 2^n$ |

| Funktion | Beschreibung |
|---|---|
| `sunmath.h` | header file |
| `fp_class(x)` | Klassifikationsfunktion |
| `isinf(x)` | Klassifikationsfunktion |
| `isnormal(x)` | Klassifikationsfunktion |
| `issubnormal(x)` | Klassifikationsfunktion |
| `iszero(x)` | Klassifikationsfunktion |
| `signbit(x)` | Klassifikationsfunktion |
| `nonstandard_arithmetic(void)` | Toggle hardware |
| `standard_arithmetic(void)` | Toggle hardware |

**IEEE Werte: Single Precision**

| IEEE Wert | Dezimalwert und hexadezimale Darstellung | C, C++ und FORTRAN |
|---|---|---|
| max normal | 3.40282347e+38 | `r = max_normalf();` |
| | `7f7fffff` | `r = r_max_normal();` |
| min normal | 1.17549435e−38 | `r = min_normalf();` |
| | `00800000` | `r = r_min_normal();` |
| max subnormal | 1.17549421e−38 | `r = max_subnormalf();` |
| | `007fffff` | `r = r_max_subnormal();` |
| min subnormal | 1.40129846e−45 | `r = min_subnormalf();` |
| | `00000001` | `r = r_min_subnormal();` |
| $\infty$ | Infinity | `r = infinityf();` |
| | `7f800000` | `r = r_infinity();` |
| quiet NaN | Nan | `r = quiet_nanf(0);` |
| | `7fffffff (SPARC, PowerPC,x86)` | `r = r_quiet_nan(0);` |
| | `7fbfffff (HP700)` | |
| signaling NaN | NaN | `r = signaling_nanf(0);` |
| | `7f800001 (SPARC, PowerPC, x86)` | `r = r_signaling_nan(0);` |
| | `7fc00001 (HP700)` | |

**IEEE Werte: Double Precision**

| IEEE Wert | Dezimalwert und hexadezimale Darstellung | C, C++ und FORTRAN |
|---|---|---|
| max normal | 1.7976931348623157e+308 | `x = max_normal();` |
| | `7fefffff ffffffff` | `x = d_max_normal();` |
| min normal | 2.2250738585072014e−308 | `x = min_normal();` |
| | `00100000 00000000` | `x = d_min_normal();` |
| max subnormal | 2.2250738585072009e−308 | `x = max_subnormal();` |
| | `000fffff ffffffff` | `x = d_max_subnormal();` |
| min subnormal | 4.9406564584214654e−324 | `x = min_subnormal();` |
| | `00000000 00000001` | `x = d_min_subnormal();` |
| $\infty$ | Infinity | `x = infinity();` |
| | `7ff00000 00000000` | `x = d_infinity();` |
| quiet NaN | Nan | `x = quiet_nan(0);` |
| | `7fffffff ffffffff (SPARC, PowerPC,x86)` | `x = d_quiet_nan(0);` |
| | `7ff7ffff ffffffff (HP700)` | |
| signaling NaN | NaN | `x = signaling_nan(0);` |
| | `7ff00000 00000001 (SPARC, PowerPC, x86)` | `x = d_signaling_nan(0);` |
| | `7ff80000 00000001 (HP700)` | |



| | | ulp error | |
|---|---|---|---|
| **Function** | **Range** | **f77 1.4** | **f77 2.0** |
| `exp` | $[-665.42, 665.42]$ | $-0.87$ ulp | $-0.85$ ulp |
| `log` | $[2^{-16.5}, 2^{16.5}]$ | $+0.81$ ulp | $+0.81$ ulp |
| `atan` | $(-\infty, +\infty)$ | $-0.59$ ulp | $-0.64$ ulp |
| `sin` | $[0.\pi/2)$ | $-0.64$ ulp | $-0.60$ ulp |
| `cos` | $[0.\pi/2)$ | $+0.63$ ulp | $+0.63$ ulp |

1 ulp $\hat{=}$ 1 unit in the last place

## 1.3 Peripherie

### 1.3.1 Eingabe

- Tastatur
- Maus (mechanisch, optisch), Trackball
- Strichcodeleser, Magnetkartenleser, *aktive* Telefonkarten
- Scanner mit OCR: maschninelle Schrift, Handschrift
- Spracheingabe
- CD's

### 1.3.2 Ausgabe

1. **Bildschirm**
   - alphanumerisch
   - graphisch (schwarz/weiß, bunt mit 8Bit/24Bit-Farbtiefe)
   - LCD-Displays für Overhead-Projektionen
   - Drucker (Tintenstrahl-, Laser-, Thermotransfer-)
   - Plotter
   - Diabelichter, Filmbelichter



### 1.3.3 Massenspeicher

1. Diskette, Festplatten, Disk-Array, MO-Platten, Wechselplatten
2. Magnetkassetten-Laufwerke für Backup: QIC, Exabyte, DAT, DLT
3. ZIP-, Syquest-Magnetwechselplatten

## 1.4 Datennetze

1. **LAN**
   - HW: TokenRing, Ethernet, FDDI, ATM
   - Protokolle: TCP/IP, OSI, IPX, DECnet, AppleTalk, ...
2. **WAN**
   - analog: Telex, Modems V.34
   - analog/digital: Modems V.90, K56flex, X2
   - digital: ISDN, X.25, ATM, ADSL
3. **Ethernet-Rahmen**: IEEE 802.3 CSMA/CD

| Bytes | 7 | 1 | 2 oder 6 | 2 oder 6 | 2 | 0-1500 | 0-46 | 4 |
|---|---|---|---|---|---|---|---|---|
| | Präambel | | Zieladresse | Quelladresse | | Daten | PAD | |

Beginn Rahmenbegrenzer, Ethernetadresse, Länge Datenfeld, Prüfsumme

4. **Token (Token-Ring)**

**Token**

| Byte | 1 | 1 | 1 |
|---|---|---|---|
| | SD | AC | ED |

**Datenrahmen**

| Byte | 1 | 1 | 1 | 2 oder 6 | 2 oder 6 | unbegrenzt | 4 | | |
|---|---|---|---|---|---|---|---|---|---|
| | SD | AC | FC | Zieladresse | Quelladresse | Daten | Prüfsumme | ED | FS |

starting delimiter, access control, ending delimiter, frame status



5. **IP-Kopf**

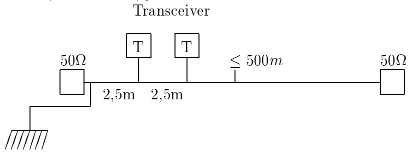
*je 32 Bit*

| 4 Bit | 4 | 8 | 1 | 1 | 1 | 13 |
|---|---|---|---|---|---|---|
| Version | HL | Dienstart | Gesamtlänge $\leq 65536$ | | | |
| Kennung | | | | NF | MF | Fragment-Offset |
| Lebensdauer $\leq$ 255sek | | Protokoll | Prüfsumme des Kopfes | | | |
| Quelladresse | | | | | | |
| Zieladresse | | | | | | |
| Option | | | | | | |

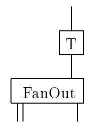
*HL=header length, NF=non fragmented, MF=more fragments*

6. **Ethernet**
   - Kabel: im Backbone Lichtwellenleiter nach FOIRL (alt) oder IEEE 802.3, oder

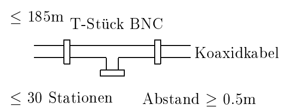
   (a) 10BASE5, Thick Wire, yellow cable



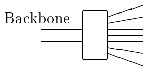

mit maximal ansteckbaren 100 Transceivern, Kaskadierung durch




   (b) 10BASE2, Thin Wire, Cheapernet

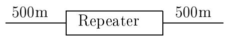


(c) 10BASE-T, Twisted Pair

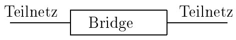



- Große Netze

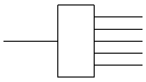
(a) Repeater: für längere Netze, um elektrischen Störungen zu begegnen, höhere Verfügbarkeit des Netzes: (maximal 2 Repeater dürfen zwischen je zwei Stationen hängen!)




(b) Bridge: filtert zusätzlich nach Kennung und Adresse, aber Zeitverzögerung

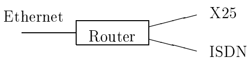



(c) Bus-Sternstruktur: Ethernet-Switch (∼ Multiport-Bridge mit schnellem internen Bus) oder schneller Backbone nach ATM/FDDI



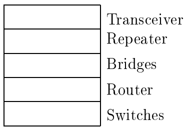
Bus-Sternstruktur

i.




ii. Hubs: z.B. auf G.15 (Etagenverteiler)

(d) SNMP-Management






- Protokolle
  (a) TCP/IP (UNIX und heterogene Netze)
  (b) OSI
  (c) IPX (Netware), NetBEUI (Windows NT Server, Windows for Workgrous), NetBIOS (OS/2)
  (d) DECnet (Digital-Equipment)
  (e) LAT (Digital-Equipment)
  (f) AppleTalk (Apple)
- Dienste: NFS, ftp, telnet, rsh/rlogin, smtp, http, ...



## 1.5 Internet

WWW-Suchmaschinen (Netsearch: Yahoo, Infosearch, Lycos, Magellan, ...), telnet/rlogin, ftp, archie/xarchie, e-mail (smtp, POP3, IMAP4), news (nntp, Schwarzes Brett), whois, X.500, www (gopher), wais = wide area information server, rsh, date, rcp, rdist, X-Window, NFS, RFS, NTP (time), talk, chat, Fax & Telefon übers Internet

- **telnet**

```
wminf4%:  telnet rs1.rrz.uni-koeln.de
Trying 134.95.100.208 ...
Connected to rs1.rrz.uni-koeln.de.
Escape character ist ']'.

AIX telnet (rs1)

IDM AIX Version 3.2.5 on rs1
(C) Copyrights by IBM and by others 1982, 1991
login:  kg099
kg099's Password:  NNNN



     Welcome to rs1.rrz.Uni-Koeln.DE (IBM RS/6000 58F, AIX 3.2.5)

     - Die Beratung (Tel:  0221-470-4563) ist per E-Mail
       unter beratung@rrz.Uni-Koeln.DE erreichbar
     - Betriebssystemprobleme, Anregungen etc.  bitte
       per E-Mail an mgr@rrz.Uni-Koeln.DE mitteilen
     - Antraege auf Aenderung der File Quota bitte per E-Mail
       an aixmgr@rrz.Uni-Koeln.DE.
     - Bitte beachten Sie die Batch-NQS-Betriebsregelung unter
       news nqs_betrieb

Last unsuccesful login:  Fri Jan 26 14:30:04 1996 on pts/28 from
wminf4.math.uni-wuppertal.de
Last login:  Tue Jul 2 14:30:20 1996 on pts/18 from wminf4@math.uni-wuppertal.
⋮
<↑]>
telnet> help
⋮
```



- **ftp**

**Name(xxx):** *account* oder *anonymous*
**Password:** *xxxx* oder *account@*

```
ftp> hash
ftp> binary
ftp> dir
ftp> get abc.txt oder put my.txt
ftp> help
ftp> quit
```

Auch Netscape benutzbar: ftp.uni-koeln.de, ftp.uni-stuttgart.de, ...

- **e-Mail**
  Z.B. mailtool; netscape; mailx

To: ______________
Subject: ______________
Cc: ______________

Adressen lauten dann beispielsweise account@wmcip4.math.uni-wuppertal.de, ...@compuserve.com, ...@aol.com, ...@uiucrmd.bitnet, ...@xxxxxx.uucp. Bei der Eingabe sollen möglichst keine Umlaute verwendet werden und reines 7Bit ASCII.
Multimedia-Inhalte können durch uuencode/uudecode oder (automatisch) MIME (=multi purpose Internet Mail Extensions) ermöglicht werden.

- **Newsgroups**

rtin/xrn/netscape

- **Archie**

Einstellungen:

**Archie-Host:** *archie.th-darmstadt.de*, archie.switch.ch, archie.doc.ic.ac.uk

**Search Type:** Exact, Subcase, Regexp, *Substring*

**Other:** FTP email address, *xxx@wmcip4.math.uni-wuppertal.de*

- **whois**

```
wminf4% whois buhl
```

```
Buhl, Fred (FB73)       fred@PURPLEFROG.COM    904-377-3376
Buhl, Ole (OB167)       hostmaster@MICRONET.DK
                                   +45 4289 3055 (FAX) +45 4566 1015
Buhl, Peter (PB539)     nomailbox@NOWHERE      45-7488-2222
Buhl, Timothy (TB2685) tcb1@ODYSSEY.NET
                                   607-753-9621 (FAX) 607-753-1547

The InterNIC Registration Services Host contains ONLY Internet
Information (Networks, ASN's, Domains, and POC's).
Please use the whois server at nic.ddn.mil for MILNET Information.
```

- **finger**

  `finger @rs1.rrz.uni-koeln.de, finger hg091@rs1.rrz.uni-koeln.de`

- **rwho**

  im LAN: rup/rusers im LAN

- **X.500**

  Vergleiche: `http://www.uni-koeln.de/dienste/x500.html`

- **netfind**

  `telnet monolith.cc.ic.ac.uk`

- **gopher**

  Vergleiche: `gopher://gopher.rrz.uni-koeln.de/1`,
  `gopher://gopher.rrz.uni-koeln.de/11/welt`

- **WWW**

  Vergleiche `http://www.uni-koeln.de/dienste/www/index.html`

- **Hyper-G**

  Vergleiche `http://www.tu-graz.ac.at`

- **Wais**

  Vergleiche: `http://gopher.rrz.uni-koeln.de/77/.wais/gopher-data/`

- **rlogin**

  `rlogin wmcip4 -l account`



- **rsh**

  rsh wmcip4

- **rcp**

  rcp wmcip4:Dateiname .

- **rdate**

  rdate wmcip4

- **rdist**

  zur Verteilung von Daten

- **talk**

  ```
  %mesg n
  %mesg y
  %talk buhl@wminf1
  ```

- **irc**

  `% irc Nickname` (= interrelay chat)



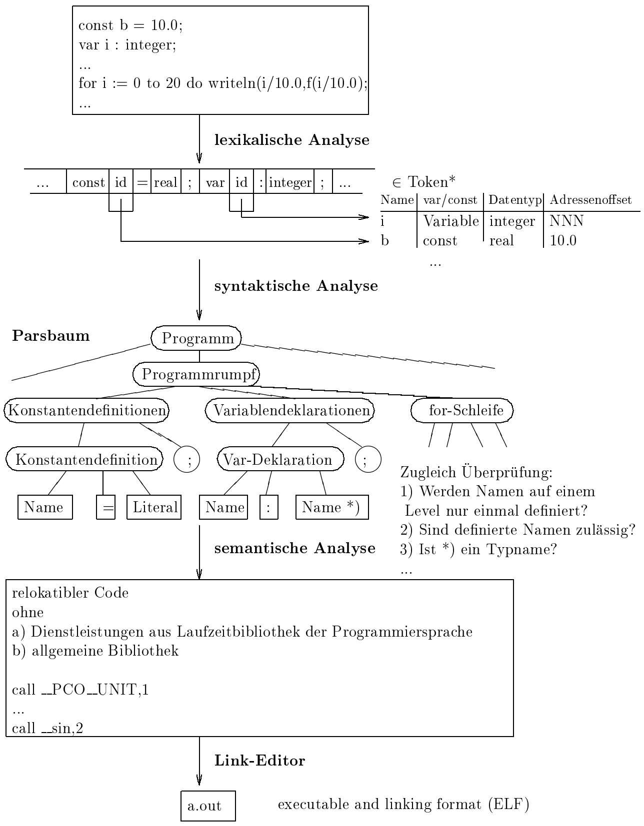
# Kapitel 2

# Systemsoftware / Basissoftware

## 2.1 Compiler und Programmentwicklung

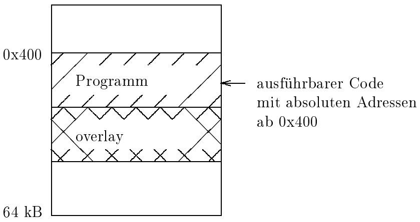
Ein Compiler führt einen in einer höheren Programmiersprache geschriebenen Algorithmus in ein Maschinenprogramm über:





**Relokatibler Code:**

- *CP/M-System:* Falls ein Overlay oder ROM-Code erzeugt werden soll, ist eine neue Compilation mit anderer Startadresse nötig.



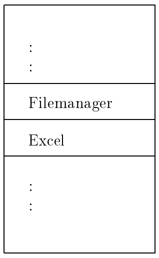

- *Idee:* Overlays, Bibliotheksmodule, ... werden *verschiebbar* mit *relativen* Adressen erzeugt.



```
**** RELOCATION INFORMATION ****

.rela.text:

Offset    Symndx  Type  Addend
0x10a9c   63      12    0
0x10aec   63      12    0
0x10ae8   63      9     0
0x10a98   63      9     0
.rela.ex_shared:

Offset    Symndx  Type  Addend
0x215cc   56      3     0
0x215c8   36      3     0
.rela.bss:

Offset    Symndx  Type  Addend
0x21648   46      19    0
0x21a70   53      19    0
.rela.plt:

Offset    Symndx  Type  Addend
0x21558   40      21    0
0x21564   68      21    0
0x21570   41      21    0
0x2157c   55      21    0
0x21588   47      21    0
0x21594   52      21    0
0x215a0   71      21    0
0x215ac   45      21    0
0x215b8   66      21    0
```

- *Möglichkeit zur Sicherheit von Prozess-Datenbereichen mit Hilfe von relokatiblem Code:*

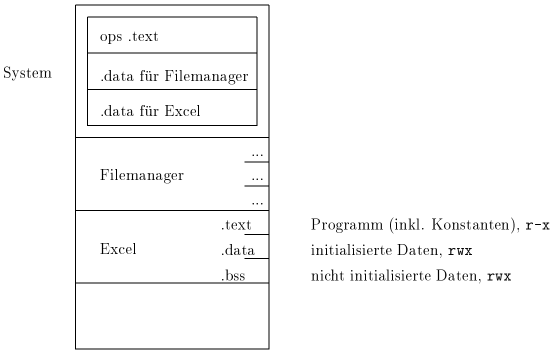





In Mehrprozesssystemen (preemptiv) sind zur selben Zeit viele Applikationen im Hauptspeicher:

1. relokatibler Code nötig!
2. Falls per Soft-/Hardware die Adressen von Speicherzugriffen validiert werden, so kann der Zugriff auf Datenbereiche anderer Applikationen unterdrückt werden.

Genauer gilt:

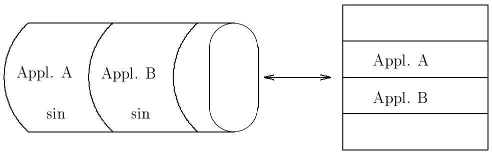



```
wminf4% size -f xspread
17(.interp) + 4796 (.hash) + 9536(.dynsym) + 5184(.dynstr)
+ 24(.rela.got) + 48(.rela.plt) + 181176(.text) + 12(.init)
+ 12(.fini) + 13872(.rodata) + 6995(.rodata1) + 72(.got) +
152(.dynamics) + 1360(.plt) + 31676(.data) + 28698(.data1)
+ 14708(.bss) = 299646
wminf%4 size -F xspread
223195(r-x) + 76672(rwx) = 300027
```

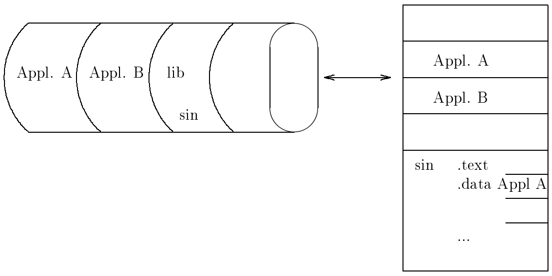


```
wminf%4 size xspread
223195 + 61964 + 14708 = 299867
```

**Statisches Linken**




**dynamisches Linken**




`dump -Lv Tabelle` liefert:

```
**** DYNAMIC SECTION INFORMATION ****

               .dynamic:
[INDEX] Tag    Value
[1]     NEEDED libpc.so.4
[2]     NEEDED libm.so.1
[3]     NEEDED libc.so.1
[4]     INIT   0x110ec
[5]     FINI   0x11124
[6]     RPATH  /opt/SUNWspro/lib
```

```
[7]     HASH   0x100e8
[8]     STRTAB 0x10778
[9]     STRSZ  0x225
[10]    SYMTAB 0x102e8
[11]    SYMENT 0x10
[12]    DEBUG  0x0
[13]    TEXTREL0x0
[14]    PLTGOT 0x21528
[15]    PLTSZ  0x6c
[16]    PLTREL 0x7
[17]    JMPREL 0x10a00
[18]    RELA   0x109a0
[19]    RELASZ 0xcc
[20]    RELAENT0xc
```

`nm Tabelle.o` liefert:

```
[Index] Value Size Type Bind Other Shndx Name
[1]     0     0    FILE LOCL 0     ABS   ./Tabelle.p
[2]     0     0    SECT LOCL 0     2
[3]     0     0    SECT LOCL 0     3
[4]     0     0    SECT LOCL 0     4
[5]     0     0    SECT LOCL 0     5
[6]     0     0    SECT LOCL 0     6
[7]     0     0    SECT LOCL 0     7
[8]     4     4    OBJT GLOB 0     COMMON1
[9]     4     4    OBJT GLOB 0     COMMONPROGFRAME
[10]    16    132  FUNC GLOB 0     2     f
[11]    0     0    NOTY GLOB 0     UNDEF __sin
[12]    0     0    NOTY GLOB 0     UNDEF output
[13]    176   396  FUNC GLOB 0     2     program
[14]    0     0    NOTY GLOB 0     UNDEF __PC0__tb_init
[15]    0     0    NOTY GLOB 0     UNDEF __PC0__UNIT
[16]    0     0    NOTY GLOB 0     UNDEF fputc
[17]    0     0    NOTY GLOB 0     UNDEF fprintf
```

`nm hello` zeigt auch die Symbole der eingebundenen Bibliotheksmodule:

```
Symbols from hello:

[Index] Value    Size Type Bind OtherShndx Name
```



```
[1]     0        0    FILE LOCL 0    ABS   hello
[2]     65748    0    SECT LOCL 0    1
[3]     65768    0    SECT LOCL 0    2
[4]     66280    0    SECT LOCL 0    3
[5]     67448    0    SECT LOCL 0    4
[6]     68000    0    SECT LOCL 0    5
[7]     68048    0    SECT LOCL 0    6
[8]     68072    0    SECT LOCL 0    7
[9]     68096    0    SECT LOCL 0    8
[10]    68208    0    SECT LOCL 0    9
[11]    69868    0    SECT LOCL 0    10
[12]    69924    0    SECT LOCL 0    11
[13]    69980    0    SECT LOCL 0    12
[14]    69984    0    SECT LOCL 0    13
[15]    69988    0    SECT LOCL 0    14
[16]    136316   0    SECT LOCL 0    15
[17]    136320   0    SECT LOCL 0    16
[18]    136488   0    SECT LOCL 0    17
[19]    136648   0    SECT LOCL 0    18
[20]    136680   0    SECT LOCL 0    19
[21]    136768   0    SECT LOCL 0    20
[22]    0        0    SECT LOCL 0    21
[23]    0        0    SECT LOCL 0    22
[24]    0        0    SECT LOCL 0    23
[25]    0        0    SECT LOCL 0    24
[26]    0        0    SECT LOCL 0    25
[27]    0        0    SECT LOCL 0    26
[28]    0        0    SECT LOCL 0    27
[29]    0        0    SECT LOCL 0    28
[30]    0        0    FILE LOCL 0    ABS   crti.s
[31]    68208    0    NOTY LOCL 0    9     _ex_text0
[32]    69980    0    NOTY LOCL 0    12    _ex_range0
[33]    136656   0    NOTY LOCL 0    18    _ex_shared0
[34]    0        0    FILE LOCL 0    ABS   crt1.s
[35]    136708   0    NOTY LOCL 0    19    __crt_scratch
[36]    0        0    FILE LOCL 0    ABS   values-Xa.c
[37]    0        0    NOTY LOCL 0    UNDEF
[38]    0        0    NOTY LOCL 0    UNDEF
[39]    0        0    FILE LOCL 0    ABS   ./hello.p
[40]    136768   0    NOTY LOCL 0    20    Bbss.bss
[41]    136720   0    NOTY LOCL 0    19    Ddata.data
[42]    0        0    FILE LOCL 0    ABS   ieee_retro.c
```



```
[43]    136728   35   OBJT LOCL 0    19    sccsid
[44]    0        0    NOTY LOCL 0    UNDEF
[45]    0        0    NOTY LOCL 0    UNDEF
[46]    0        0    FILE LOCL 0    ABS   __ieee_retro.S
[47]    0        0    FILE LOCL 0    ABS   crtn.s
[48]    69868    0    NOTY LOCL 0    9     _ex_text1
[49]    69980    0    NOTY LOCL 0    12    _ex_range1
[50]    136672   0    NOTY LOCL 0    18    _ex_shared1
[51]    68208    244  FUNC GLOB 0    9     _start
[52]    68688    1024 FUNC WEAK 0    9     ieee_retrospective
[53]    136768   4    OBJT GLOB 0    20    i
[54]    136680   4    OBJT GLOB 0    19    _environ
[55]    138160   0    OBJT GLOB 0    20    _end
[56]    137840   320  OBJT WEAK 0    20    _iob
[57]    69796    28   FUNC GLOB 0    9     __get_accrued
[58]    0        0    NOTY WEAK 0    UNDEF _ex_register
[59]    136772   4    OBJT GLOB 0    20    PROGFRAME
[60]    136316   0    OBJT GLOB 0    15    _GLOBAL_OFFSET_TABLE_
[61]    69768    28   FUNC GLOB 0    9     __get_traps
[62]    136536   0    FUNC GLOB 0    UNDEF atexit
[63]    136560   0    FUNC GLOB 0    UNDEF exit
[64]    69824    28   FUNC GLOB 0    9     __get_direction
[65]    68688    1024 FUNC GLOB 0    9     __ieee_retrospective
[66]    69868    56   FUNC GLOB 0    10    _init
[67]    136620   0    FUNC GLOB 0    UNDEF fputc
[68]    136776   1060 OBJT GLOB 0    20    output
[69]    136584   0    FUNC GLOB 0    UNDEF __PC0__tb_init
[70]    136716   4    OBJT GLOB 0    19    ___Argv
[71]    69740    28   FUNC GLOB 0    9     __get_nonstandard
[72]    69852    16   FUNC GLOB 0    9     __get_precision
[73]    136320   0    OBJT GLOB 0    16    _DYNAMIC
[74]    136596   0    FUNC GLOB 0    UNDEF __PC0__UNIT
[75]    137840   320  OBJT GLOB 0    20    __iob
[76]    68480    208  FUNC GLOB 0    9     program
[77]    136572   0    FUNC GLOB 0    UNDEF _exit
[78]    0        0    NOTY WEAK 0    UNDEF _ex_deregister
[79]    136680   4    OBJT WEAK 0    19    environ
[80]    136712   4    OBJT GLOB 0    19    __cg89_used
[81]    136712   4    OBJT GLOB 0    19    __cg92_used
[82]    136712   4    OBJT GLOB 0    19    __fnonstd_used
[83]    136763   0    OBJT GLOB 0    19    _edata
[84]    136488   0    OBJT GLOB 0    17    _PROCEDURE_LINKAGE_TABLE_
```



```
[85]    0        0    NOTY GLOB 0    ABS   __fsr_init_value
[86]    70779    0    OBJT GLOB 0    14    _etext
[87]    69984    4    OBJT GLOB 0    13    _lib_version
[88]    136632   0    FUNC GLOB 0    UNDEF fflush
[89]    69712    28   FUNC WEAK 0    9     ieee_retrospective_
[90]    136548   0    FUNC GLOB 0    UNDEF main
[91]    136684   24   OBJT GLOB 0    19    __environ_lock
[92]    69924    56   FUNC GLOB 0    11    _fini
[93]    136608   0    FUNC GLOB 0    UNDEF fprintf
[94]    69712    28   FUNC GLOB 0    9     __ieee_retrospective_
```

`truss a.out` enthüllt die komplexen Systemprozesse, die ein Programmstart initiiert:

```
execve("./a.out", 0xEFFFF3B0, 0xEFFFF3B8) argc = 1
open("/dev/zero", O_RDONLY) = 3
mmap(0x00000000, 4096, PROT_READ|PROT_WRITE|PROT_EXEC, MAP_PRIVATE,
3, 0) = 0xEF7C0000
open("/usr/lib/libpc.so.4", O_RDONLY) Err#2 ENOENT
open("/usr/openwin/lib/libpc.so.4", O_RDONLY) Err#2 ENOENT
open("/usr/ucblib/libpc.so.4", O_RDONLY) Err#2 ENOENT
open("/usr/dt/lib/libpc.so.4", O_RDONLY) Err#2 ENOENT
open("/opt/local/lib/libpc.so.4", O_RDONLY) Err#2 ENOENT
open("/opt/SUNWspro/lib/libpc.so.4", O_RDONLY) = 4
fstat(4, 0xEFFFF064) = 0
mmap(0x00000000, 4096, PROT_READ|PROT_EXEC, MAP_SHARED, 4, 0)
= 0xEF7B0000
mmap(0x00000000, 143360, PROT_READ|PROT_EXEC, MAP_PRIVATE, 4,
0) = 0xEF780000
munmap(0xEF790000, 61440) = 0
mmap(0xEF79F000, 12384, PROT_READ|PROT_WRITE|PROT_EXEC, MAP_PRIVATE|MAP_FIX
4, 61440) = 0xEF79F000
close(4) = 0
open("/usr/lib/libm.so.1", O_RDONLY) = 4
fstat(4, 0xEFFFF064) = 0
mmap(0xEF7B0000, 4096, PROT_READ|PROT_EXEC, MAP_SHARED|MAP_FIXED,
4, 0) = 0xEF7B0000
mmap(0x00000000, 155648, PROT_READ|PROT_EXEC, MAP_PRIVATE, 4,
0) = 0xEF750000
munmap(0xEF765000, 61440) = 0
mmap(0xEF774000, 6056, PROT_READ|PROT_WRITE|PROT_EXEC, MAP_PRIVATE|MAP_FIXE
4, 81920) = 0xEF774000
close(4) = 0
```

```
open("/usr/lib/libc.so.1", O_RDONLY) = 4
fstat(4, 0xEFFFF064) = 0
mmap(0xEF7B0000, 4096, PROT_READ|PROT_EXEC, MAP_SHARED|MAP_FIXED,
4, 0) = 0xEF7B0000
mmap(0x00000000, 618496, PROT_READ|PROT_EXEC, MAP_PRIVATE, 4,
0) = 0xEF680000
munmap(0xEF6FE000, 61440) = 0
mmap(0xEF70D000, 28768, PROT_READ|PROT_WRITE|PROT_EXEC, MAP_PRIVATE|MAP_FIX
4, 512000) = 0xEF70D000
mmap(0xEF715000, 4792, PROT_READ|PROT_WRITE|PROT_EXEC, MAP_PRIVATE|MAP_FIXE
3, 0) = 0xEF715000
close(4) = 0
open("/usr/lib/libdl.so.1", O_RDONLY) = 4
fstat(4, 0xEFFFF064) = 0
mmap(0xEF7B0000, 4096, PROT_READ|PROT_EXEC, MAP_SHARED|MAP_FIXED,
4, 0) = 0xEF7B0000
close(4) = 0
mprotect(0x00010000, 5243, PROT_READ|PROT_WRITE|PROT_EXEC) =
0
open("/usr/platform/SUNW,SPARCstation-10/lib/libc_psr.so.1",
O_RDONLY) Err#2 ENOENT
mprotect(0x00010000, 5243, PROT_READ|PROT_EXEC) = 0
close(3) = 0
getrlimit(RLIMIT_NOFILE, 0xEFFFEEE8) = 0
brk(0x00021BB0) = 0
brk(0x00023BB0) = 0
access("/usr/lib/locale/de/LC_CTYPE/ctype", 4) = 0
open("/usr/lib/locale/de/LC_CTYPE/ctype", O_RDONLY) = 3
read(3, " \ 0 ( (".., 521) = 521
close(3) = 0
access("/usr/lib/locale/de/LC_NUMERIC/numeric", 4) = 0
open("/usr/lib/locale/de/LC_NUMERIC/numeric", O_RDONLY) = 3
read(3, " , ", 2) = 2
close(3) = 0
open("/usr/lib/locale/de/LC_TIME/time", O_RDONLY) = 3
close(3) = 0
access("/usr/lib/locale/de/LC_COLLATE/coll.so", 4) Err#2 ENOENT
access("/usr/lib/locale/de/LC_COLLATE/CollTable", 4) = 0
open("/usr/lib/locale/de/LC_MONETARY/monetary", O_RDONLY) =
3
close(3) = 0
stat("/usr/lib/locale/de/", 0xEFFFEAE0) = 0
open("/usr/dt/lib/nls/msg/de/SUNW_SPRO_SC_libpc.cat.cat", O_RDONLY)
```



```
Err#2 ENOENT
open("/usr/dt/lib/nls/msg/C/SUNW_SPRO_SC_libpc.cat.cat", O_RDONLY)
Err#2 ENOENT
open("../../lib/locale/de/LC_MESSAGES/SUNW_SPRO_SC_libpc.cat",
O_RDONLY) Err#2 ENOENT
open("/usr/lib/locale/de/LC_MESSAGES/SUNW_SPRO_SC_libpc.cat",
O_RDONLY) Err#2 ENOENT
sigaction(SIGQUIT, 0x00000000, 0xEFFFF238) = 0
sigaction(SIGQUIT, 0xEFFFF1B8, 0x00000000) = 0
sigaction(SIGILL, 0x00000000, 0xEFFFF238) = 0
sigaction(SIGILL, 0xEFFFF1B8, 0x00000000) = 0
sigaction(SIGTRAP, 0x00000000, 0xEFFFF238) = 0
sigaction(SIGTRAP, 0xEFFFF1B8, 0x00000000) = 0
sigaction(SIGEMT, 0x00000000, 0xEFFFF238) = 0
sigaction(SIGEMT, 0xEFFFF1B8, 0x00000000) = 0
sigaction(SIGFPE, 0x00000000, 0xEFFFF238) = 0
sigaction(SIGFPE, 0xEFFFF1B8, 0x00000000) = 0
sigaction(SIGBUS, 0x00000000, 0xEFFFF238) = 0
sigaction(SIGBUS, 0xEFFFF1B8, 0x00000000) = 0
sigaction(SIGSEGV, 0x00000000, 0xEFFFF238) = 0
sigaction(SIGSEGV, 0xEFFFF1B8, 0x00000000) = 0
sigaction(SIGSYS, 0x00000000, 0xEFFFF238) = 0
sigaction(SIGSYS, 0xEFFFF1B8, 0x00000000) = 0
sigaction(SIGXCPU, 0x00000000, 0xEFFFF238) = 0
sigaction(SIGXCPU, 0xEFFFF1B8, 0x00000000) = 0
sigaction(SIGXFSZ, 0x00000000, 0xEFFFF238) = 0
sigaction(SIGXFSZ, 0xEFFFF1B8, 0x00000000) = 0
ioctl(1, TCGETA, 0xEFFFE8FC) = 0
ioctl(1, TCGETA, 0xEFFFE8FC) = 0
0,00000000000000e+00
write(1, " 0 , 0 0 0 0 0 0 0 0 0".., 22) = 22
1,00000000000000e-01
write(1, " 1 , 0 0 0 0 0 0 0 0 0".., 22) = 22
2,00000000000000e-01
write(1, " 2 , 0 0 0 0 0 0 0 0 0".., 22) = 22
3,00000000000000e-01
write(1, " 3 , 0 0 0 0 0 0 0 0 0".., 22) = 22
4,00000000000000e-01
write(1, " 4 , 0 0 0 0 0 0 0 0 0".., 22) = 22
5,00000000000000e-01
write(1, " 5 , 0 0 0 0 0 0 0 0 0".., 22) = 22
6,00000000000000e-01
write(1, " 6 , 0 0 0 0 0 0 0 0 0".., 22) = 22
```



```
7,00000000000000e-01
write(1, " 7 , 0 0 0 0 0 0 0 0 0".., 22) = 22
8,00000000000000e-01
write(1, " 8 , 0 0 0 0 0 0 0 0 0".., 22) = 22
9,00000000000000e-01
write(1, " 9 , 0 0 0 0 0 0 0 0 0".., 22) = 22
1,00000000000000e+00
write(1, " 1 , 0 0 0 0 0 0 0 0 0".., 22) = 22
1,10000000000000e+00
write(1, " 1 , 1 0 0 0 0 0 0 0 0".., 22) = 22
1,20000000000000e+00
write(1, " 1 , 2 0 0 0 0 0 0 0 0".., 22) = 22
1,30000000000000e+00
write(1, " 1 , 3 0 0 0 0 0 0 0 0".., 22) = 22
1,40000000000000e+00
write(1, " 1 , 4 0 0 0 0 0 0 0 0".., 22) = 22
1,50000000000000e+00
write(1, " 1 , 5 0 0 0 0 0 0 0 0".., 22) = 22
1,60000000000000e+00
write(1, " 1 , 6 0 0 0 0 0 0 0 0".., 22) = 22
1,70000000000000e+00
write(1, " 1 , 7 0 0 0 0 0 0 0 0".., 22) = 22
1,80000000000000e+00
write(1, " 1 , 8 0 0 0 0 0 0 0 0".., 22) = 22
1,90000000000000e+00
write(1, " 1 , 9 0 0 0 0 0 0 0 0".., 22) = 22
2,00000000000000e+00
write(1, " 2 , 0 0 0 0 0 0 0 0 0".., 22) = 22
lseek(0, 0, SEEK_CUR) = 18206
_exit(0)
```

`ldd -r hello` zeigt die benötigten (dynamisch gelinkten) Bibliotheken:

```
libpc.so.4 => /opt/SUNWspro/lib/libpc.so.4
libc.so.1 =>  /usr/lib/libc.so.1
libdl.so.1 => /usr/lib/libdl.so.1
```

`truss hello` im Falle einer statisch gelinkten Applikation (erzeugt mittels `pc -Bstatic ...`):



```
execve("./hello", 0xEFFFF3A8, 0xEFFFF3B4) argc = 2
open("/dev/zero", O_RDONLY) = 3
mmap(0x00000000, 4096, PROT_READ|PROT_WRITE|PROT_EXEC, MAP_PRIVATE,
3, 0) = 0xEF7C0000
close(3) = 0
getrlimit(RLIMIT_NOFILE, 0xEFFFEEE0) = 0
brk(0x00046D90) = 0
brk(0x00048D90) = 0
access("/usr/lib/locale/de/LC_CTYPE/ctype", 4) = 0
open("/usr/lib/locale/de/LC_CTYPE/ctype", O_RDONLY) = 3
read(3, "\ 0 ( (".., 521) = 521
close(3) = 0
access("/usr/lib/locale/de/LC_NUMERIC/numeric", 4) = 0
open("/usr/lib/locale/de/LC_NUMERIC/numeric", O_RDONLY) = 3
read(3, " , ", 2) = 2
close(3) = 0
open("/usr/lib/locale/de/LC_TIME/time", O_RDONLY) = 3
close(3) = 0
access("/usr/lib/locale/C/LC_CTYPE/ctype", 4) = 0
open("/usr/lib/locale/C/LC_CTYPE/ctype", O_RDONLY) = 3
read(3, "\ 0 ( (".., 521) = 521
close(3) = 0
access("/usr/lib/locale/C/LC_NUMERIC/numeric", 4) = 0
open("/usr/lib/locale/C/LC_NUMERIC/numeric", O_RDONLY) = 3
read(3, " .\ 0", 2) = 2
close(3) = 0
open("/usr/dt/lib/nls/msg/de/SUNW_SPRO_SC_libpc.cat.cat", O_RDONLY)
Err#2 ENOENT
open("/usr/dt/lib/nls/msg/C/SUNW_SPRO_SC_libpc.cat.cat", O_RDONLY)
Err#2 ENOENT
open("../../lib/locale/de/LC_MESSAGES/SUNW_SPRO_SC_libpc.cat",
O_RDONLY) Err#2 ENOENT
open("/usr/lib/locale/de/LC_MESSAGES/SUNW_SPRO_SC_libpc.cat",
O_RDONLY) Err#2 ENOENT
sigaction(SIGQUIT, 0x00000000, 0xEFFFF230) = 0
sigaction(SIGQUIT, 0xEFFFF1B0, 0x00000000) = 0
sigaction(SIGILL, 0x00000000, 0xEFFFF230) = 0
sigaction(SIGILL, 0xEFFFF1B0, 0x00000000) = 0
sigaction(SIGTRAP, 0x00000000, 0xEFFFF230) = 0
sigaction(SIGTRAP, 0xEFFFF1B0, 0x00000000) = 0
sigaction(SIGEMT, 0x00000000, 0xEFFFF230) = 0
sigaction(SIGEMT, 0xEFFFF1B0, 0x00000000) = 0
sigaction(SIGFPE, 0x00000000, 0xEFFFF230) = 0
```

```
sigaction(SIGFPE, 0xEFFFF1B0, 0x00000000) = 0
sigaction(SIGBUS, 0x00000000, 0xEFFFF230) = 0
sigaction(SIGBUS, 0xEFFFF1B0, 0x00000000) = 0
sigaction(SIGSEGV, 0x00000000, 0xEFFFF230) = 0
sigaction(SIGSEGV, 0xEFFFF1B0, 0x00000000) = 0
sigaction(SIGSYS, 0x00000000, 0xEFFFF230) = 0
sigaction(SIGSYS, 0xEFFFF1B0, 0x00000000) = 0
sigaction(SIGXCPU, 0x00000000, 0xEFFFF230) = 0
sigaction(SIGXCPU, 0xEFFFF1B0, 0x00000000) = 0
sigaction(SIGXFSZ, 0x00000000, 0xEFFFF230) = 0
sigaction(SIGXFSZ, 0xEFFFF1B0, 0x00000000) = 0
ioctl(1, TCGETA, 0xEFFFE8FC) = 0
ioctl(1, TCGETA, 0xEFFFE8FC) = 0
0.00000000000000e+00
write(1, " 0 .  0 0 0 0 0 0 0 0 0".., 22) = 22
1.00000000000000e-01
write(1, " 1 .  0 0 0 0 0 0 0 0 0".., 22) = 22
2.00000000000000e-01
write(1, " 2 .  0 0 0 0 0 0 0 0 0".., 22) = 22
3.00000000000000e-01
write(1, " 3 .  0 0 0 0 0 0 0 0 0".., 22) = 22
4.00000000000000e-01
write(1, " 4 .  0 0 0 0 0 0 0 0 0".., 22) = 22
5.00000000000000e-01
write(1, " 5 .  0 0 0 0 0 0 0 0 0".., 22) = 22
6.00000000000000e-01
write(1, " 6 .  0 0 0 0 0 0 0 0 0".., 22) = 22
7.00000000000000e-01
write(1, " 7 .  0 0 0 0 0 0 0 0 0".., 22) = 22
8.00000000000000e-01
write(1, " 8 .  0 0 0 0 0 0 0 0 0".., 22) = 22
9.00000000000000e-01
write(1, " 9 .  0 0 0 0 0 0 0 0 0".., 22) = 22
1.00000000000000e+00
write(1, " 1 .  0 0 0 0 0 0 0 0 0".., 22) = 22
1.10000000000000e+00
write(1, " 1 .  1 0 0 0 0 0 0 0 0".., 22) = 22
1.20000000000000e+00
write(1, " 1 .  2 0 0 0 0 0 0 0 0".., 22) = 22
1.30000000000000e+00
write(1, " 1 .  3 0 0 0 0 0 0 0 0".., 22) = 22
1.40000000000000e+00
write(1, " 1 .  4 0 0 0 0 0 0 0 0".., 22) = 22
```

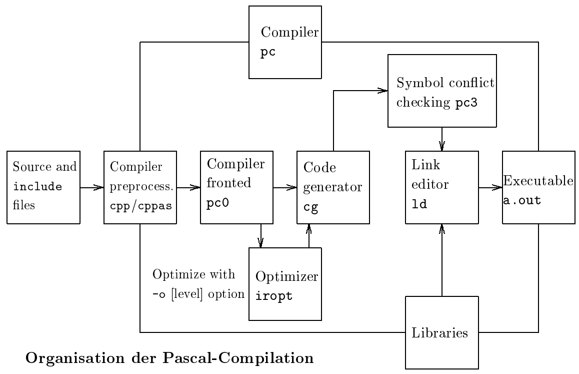


```
1.50000000000000e+00
write(1, " 1 .  5 0 0 0 0 0 0 0 0".., 22) = 22
1.60000000000000e+00
write(1, " 1 .  6 0 0 0 0 0 0 0 0".., 22) = 22
1.70000000000000e+00
write(1, " 1 .  7 0 0 0 0 0 0 0 0".., 22) = 22
1.80000000000000e+00
write(1, " 1 .  8 0 0 0 0 0 0 0 0".., 22) = 22
1.90000000000000e+00
write(1, " 1 .  9 0 0 0 0 0 0 0 0".., 22) = 22
2.00000000000000e+00
write(1, " 2 .  0 0 0 0 0 0 0 0 0".., 22) = 22
lseek(0, 0, SEEK_CUR) = 4840
_exit(0)
```

Für eine kurze Übersicht der durchgeführten System-Calls benutze man `truss -c hello`:

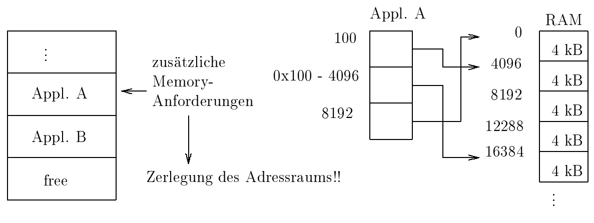
| syscall | seconds | calls | errors |
|---|---|---|---|
| _exit | .00 | 1 | |
| read | .00 | 2 | |
| write | .00 | 21 | |
| open | .01 | 19 | 10 |
| close | .01 | 9 | |
| brk | .00 | 2 | |
| stat | .00 | 1 | |
| lseek | .00 | 1 | |
| fstat | .00 | 4 | |
| access | .01 | 4 | 1 |
| ioctl | .00 | 2 | |
| execve | .00 | 1 | |
| sigaction | .01 | 20 | |
| mmap | .01 | 12 | |
| mprotect | .00 | 2 | |
| munmap | .00 | 3 | |
| getrlimit | .00 | 17 | |
| sys totals: | .05 | 1057 | 11 |
| usr time: | .04 | | |
| elapsed: | .29 | | |

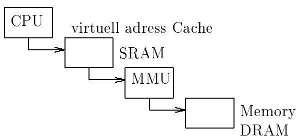





**Organisation der Pascal-Compilation**

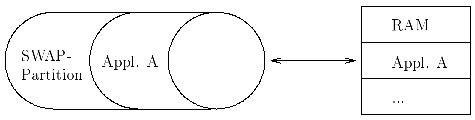
**Paging:** Virtuelles Memory. *Problem:* Es ergibt sich bei wachsender Laufzeit auch ein erhöter Speicherbedarf!






**Swapping:** *Problem:* Der aktuelle Applikations-Hauptspeicherbedarf ist größer als das vorhandender RAM.




1. `.text` braucht nicht geswapt zu werden. Warum?
2. Mischung von Swapping und Paging:

   memory fault ⟶ geswappte Page ⟶ RAM