

Das Jahr 2000 Problem - Sorgen mit der Zeit?

Vortrag beim Treffen Lions Club Kassel Brüder Grimm
am 20. August 1999

Lutz Wegner
FB Mathematik/Informatik
Universität Gh Kassel

Allgemein bekanntes Grundproblem (2 läppische Bytes)



Tatsächlich ist Silvester 1999 nur einer der problematischen Termine

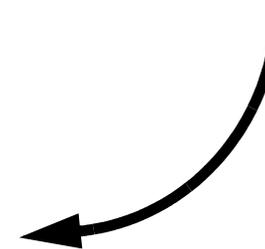
1998-01-01	Stopperfeld für Jahr 98
1999-01-01	Stopperfeld für Jahr 99
1999-06-30	Ende Finanzjahr 99
1999-09-09	magische Zahl 9/9/99
2000-01-01	Überlauf zweistelliges Jahr
2000-01-10	erstes neunstelliges Datum
2000-10-10	erstes zehnstelliges Datum
2000-02-29	Schaltjahr
2000-12-31	366. Tag im Jahr
2001-01-01	21. Jahrhundert
2038-01-18	2^{31} Sekunden seit 1. Januar 1970
2099-12-31	Ende der logischen DOS Uhr
2100-01-01	kein Schaltjahr
3000-01-01	kein Schaltjahr



Problem Nr. 1: Mutter Natur



Sie dreht sich nicht einmal
in $24 * 3600$ Sekunden



Abhilfe: Der Gregorianische Kalender

Geht auf Papst Gregor XIII zurück, im Oktober 1582 eingeführt;

läßt auf den 4. Oktober 1582 den 15. Oktober 1582 folgen

beinhaltet die aus der Schule bekannte Schaltjahr-Regel:

ein Jahr ist ein Schaltjahr (dem 28. Februar folgt ein 29. Februar, sonst folgt ein 1. März),

wenn es sich das Jahr ohne Rest durch 4 teilen läßt,

außer es läßt sich auch durch 100 ohne Rest teilen (kein Schaltjahr)

außer es läßt sich auch durch 400 ohne Rest teilen (Schaltjahr)

Der Gregorianische Kalender ist extrem unhandlich (aus DV Sicht eine Katastrophe):

Es gibt kein Jahr 0, die Abstände für Tage/Monat sind ungleich, die Tage pro Jahr sind ungleich (365 Tage oder 366), der 1. Wochentag jedes Jahrs wechselt (aber unregelmäßig), usw.

Einer der verantwortlichen Herrn (Christopher Clavius, 1537 - 1612)



Gibt es ein Y4K Problem?

Das gregorianische Jahr stimmt immer noch nicht mit dem tropischen Jahr überein. Trotz Schaltjahrregel ist es um ungefähr 0,0003 Tage \approx 26 Sekunden länger $\rightarrow 24 * 3600 : 26 = 3323,0769$: alle 4000 Jahre kein Schaltjahr? Die Encyclopedia Britannica zitiert (fälschlich) diese Regel, die auf Sir John Herschel 1830 zurückgehen soll. Tatsächlich gilt aber die vom Britischen Parlament 1752 angenommene unveränderte Regel, die auch für die amerikanischen Kolonien übernommen wurde, damit für die USA heute.

Eine bessere Näherung als 4000 wäre 3200. Da das gregorianische Jahr seit ca. 1600 ungefähr synchron mit der Natur läuft, wäre es vernünftig eine 3200-Regel einzuführen (Schalttag fällt aus wenn auch ohne Rest durch 3200 teilbar) und sie damit auch im Jahr 3200 zum ersten Mal anzuwenden (wenn der Fehler gerade einen halben Tag ausmacht).

Dafür hätten wir noch 1200 Jahre Zeit uns zu einigen, wenn wir nicht wieder auf letzten Drücker anfangen umzustellen.

Der Julianische Kalender

geht auf Anordnung Julius Cäsars zurück (ca. 50 v.Chr.), enthielt auf Ratschlag des Astronomen Sosigenes auch ein Schaltjahr (alle 4 Jahre).

Wird von den Astronomen benutzt, die vom Datum 0 (Julianischer Tag 0 = mittags am 1. Januar 4713 v. Chr.) an die Tage zählen. Der julianische Tag beginnt mittags *Coordinated Universal Time* (UTC, früher *Greenwich Mean Time*, GMT), die julianische Uhr läuft in Bruchteilen eines Tages. Mitternacht ist demnach 0,5 Tag, 13:00 Uhr dann 0.041666667 Tag (1/24). Der Uhrenbruchteil darf zum Tag addiert werden, so wäre 13:00 Uhr am 1. Januar 2000 dann JD 2451545.041666667. Die übliche Einheit bei astronomischen Beobachtungen ist 0.0001 Tag = 8,64 Sekunden.

Wäre ein gutes DV Datumssystem! Einige der Reparaturvorschläge basieren auf einer (teilweisen) Umstellung auf JD.

In UTC gibt es Schaltsekunden, das Sekundenfeld müßte von 0 bis 60 laufen!

Schaltjahrprüfungen

Häufige (falsche) Prüfung begrenzt auf ganzzahlige Teilbarkeit durch 4

```
function Ist_Schaltjahr(int Jahr): Boolean;  
begin  
    Ist_Schaltjahr := (Jahr mod 4 = 0)  
end;
```

Falsches Ergebnis 1900, 2100, 2200, ... richtiges Ergebnis 1600, 2000 (sic), 2400, ...

häufig in elektrischen Uhren zu finden, Zeitchips, usw.

Auch falsch, aber seltener anzufinden:

```
Ist_Schaltjahr := (Jahr mod 4 = 0) and not (Jahr mod 100 = 0)
```

Falsches Ergebnis 1600, 2000 (sic), 2400, ... richtiges 1900, 2100, 2200, ...

Problem

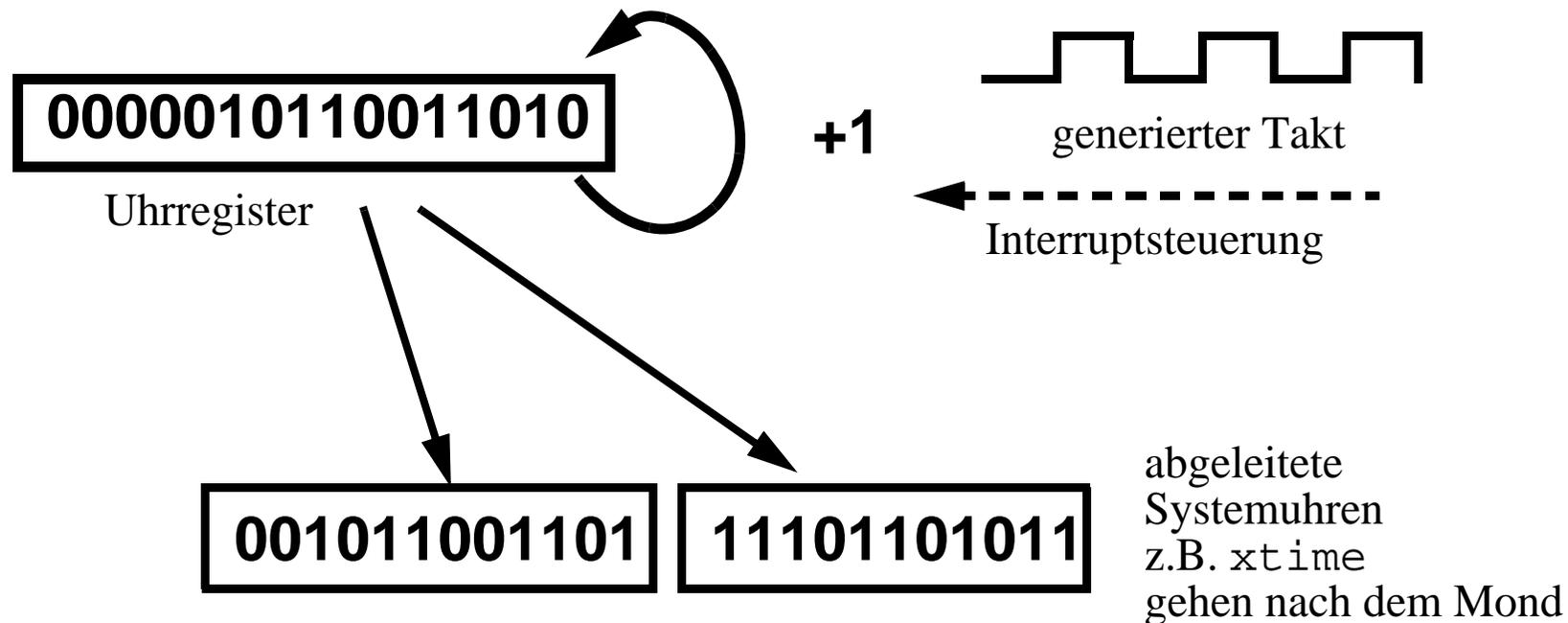
$a \bmod 4 = 0 \Leftrightarrow (a \operatorname{div} 4) * 4 = a \Leftrightarrow (a \operatorname{shr} 2) \operatorname{shl} 2 = a \Leftrightarrow a \operatorname{and} 0\dots03 = 0$ (bitweises und) $\Leftrightarrow \dots$

Falls Prüfung inline codiert im Assembler oder gar Binärcode, dann kaum zu identifizieren.

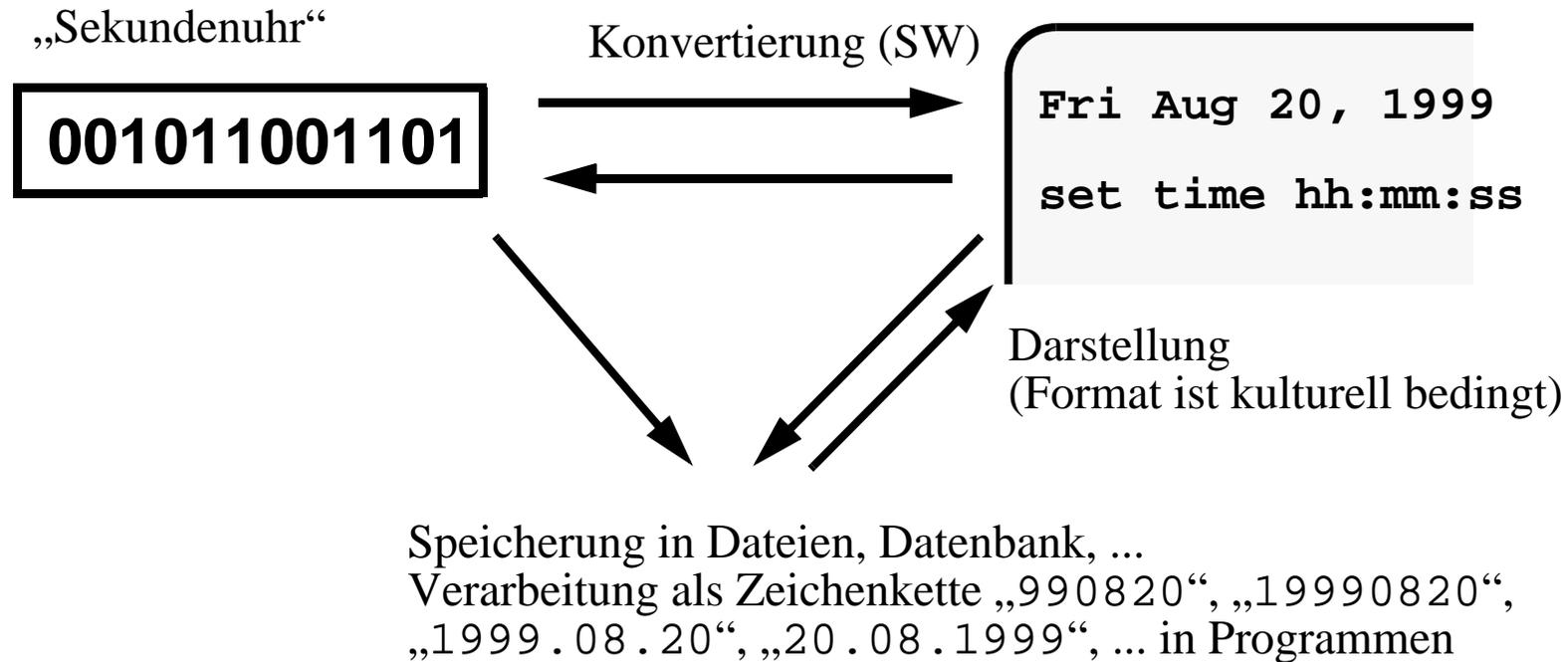
Problem Nr. 2: Sparsamkeit am falschen Platz

Wie werden Uhrzeit und Datum im Rechner gespeichert?

Alle Rechneruhren sind Zählregister, die in einem rechner- und/oder betriebssystemabhängigen Takt hochgezählt werden (z.B. Linux Kern in einer LongInt Variablen `jiffies` alle 10 Millisekunden).



Datums und Zeiterzeugung aus der Systemuhr



Kurzer Ausflug in Zahldarstellungen im Rechner

0	0000
1	0001
	⋮
9	1001
A	1010
	⋮
F	1111

Darstellungen für die Jahreszahl 1999 in 2 Bytes

3	9	3	9	ASCII (99)
1	9	9	9	BCD (1999)
0	7	C	F	Dualzahl

4 bits = 1 Nibble

Ein einfaches Konvertierungsbeispiel: UNIX

UNIX: sehr einfach; Systemuhr läuft in Sekunden ab 1.1.1970, 32 bit Wert mindestens, damit $2^{31} : (24 * 3600 * 365) \approx 68$ Jahre (genau bis 18. Januar 2038), bereits jetzt aber Trend zu 64 bit Maschinen, also kein Problem.

Für diejenigen, die gerne langfristig denken:

2^{64} Sekunden sind ca. $5.8 * 10^{11}$ Jahre = 580 Milliarden Jahre

die Erde selbst ist ungefähr 4 Milliarden Jahre alt, das sind $4 * 10^9$ Jahre

das Universum soll jünger als 16 Milliarden Jahre sein

die Sonne wird es keine 10 Milliarden Jahre mehr machen

mit 64 bit liegen auch diejenigen auf der sicheren Seite, die im Sinne der fail-safe Systeme lieber Hosenträger **und** Gürtel tragen.

Ein ziemlich krankes Beispiel: DOS (bitte BIOS Produktionsdatum bereit halten)

DOS kennt nur die Jahre 1980 bis 2099

Ein Aufruf der Systemfunktion Get Real-Time Clock Date (BIOS Interrupt 15h) liefert

CX	19 oder 20	BCD Jahr*	*Binary Coded Decimal, also zwei Dezimalziffern
DX	BCD Monat*	BCD Tag*	

daraus 16 bit DOS FILE_DATE Werte nach der Formel:

$(\text{Jahr}-1980) * 512 + \text{Monat} * 32 + \text{Tag}$, z.B. 31. Dez. 1999 = 10.143 = 279Fh

0 - 127	0 - 15	0 - 31
7	4	5

Die 7 gängigsten Reparaturmethoden für das Jahr 2000 Problem

- (1) Betrieb einschränken, Geschäftszweig verkaufen
- (2) Zuwarten (jetzt ist es eh zu spät großartig umzustellen)
- (3) Neues Release kaufen, auf SAP R/3 umstellen
- (4) YY Feld auf CCYY umstellen
- (5) Intelligentes Umcodieren
- (6) Datumsfenster, Pivotlösung
- (7) Uhr zurückstellen um 28 Jahre

Im Detail (von hinten nach vorn)

(7) Uhr zurückstellen - Historikerlösung - quick & dirty

1 Schalttag alle 4 Jahre * 7 Tage die Woche = Uhr 28 Jahre zurückstellen

damit sind alle Wochentage, Feiertage, usw. korrekt

Vom richtigen Datum werden 28 abgezogen, die letzten zwei Bytes werden gespeichert. Beim Laden umgekehrt 28 zugezählt, funktioniert mit Jahresangaben 1928 - 2027, alle Programme müssen entsprechend angepaßt werden (einfache Logik), Datensortierungen laufen ok

(6) Datumsfenster - Pivotlösung - Low Tech, high complexity

Es bleibt bei einer zweistelligen Jahresspeicherung YY.

Das Jahrhundert CC entscheidet man auf der Basis eines sog. **Pivotjahres**, z.B. fest 1970, d.h.

wenn $YY < 70$ dann $CC := 20$ sonst $CC := 19$

funktioniert für Angaben 1970 bis 2069, Sortierung und Datenbankschlüssel funktionieren NICHT, Programmänderung überall erforderlich, komplexe Logik (fehlerbehaftet)

Variation: verschiebliches Fenster:

wenn $YY < (\text{jetzt} - 50)$ dann $CC := 20$ sonst $CC := 19$

Beispiel: jetzt sei 1999, jetzt - 50 = 1949, davon die letzten zwei Ziffern = 49

Zinszahlung endet 19: $19 < 49 \rightarrow CC := 20$, Anleihe bis 2019

Zinszahlung endet 72: $72 \geq 49 \rightarrow CC := 19$, abgelaufen 1972

Beurteilung wie oben, 100 Jahre Fenster (keine Kriegsanleihe -1919)

Unangenehme Überraschung möglich (Zirkelbach, VW Gedas):

C-Systemfunktion liefert Jahr ab 1900, ab 2000 dreistellig \rightarrow Pivotrechnung falsch!!

(5) CCYY Jahreszahl in 2 Bytes unterbringen - Bitfummelei

verschiedene Varianten möglich

man bringt irgendwo ein 1-bit-Markierung (oder 1 Dezimalziffer, 1 Hexziffer) unter zur Unterscheidung des Jahrhunderts, z.B. auch im Zonenfeld des obersten Bytes

man speichert CCYY als vorzeichenlosen BCD-Wert ab

Lösungen mit Lilian-Datum (fester Tag als Anfangsjahr, Epoche)

keine Einigung, differierende „Standards“: ANSI X3.30-1985 setzt 1. Januar 1601

ANSI X3.51-1994, ISO 8601:1988, FIPS 4-1 NIST, DIN 5008 (Mai 1996), EN 28601,

ANSI Vorschlag für C-Schnittstelle setzt UTC = ATI (International Atomic Time) wie UNIX Startjahr auf 1. Januar 1970, 00:00:00 (obwohl es UTC und ATI erst seit 1972 gibt)

Julianisches Startdatum: 31. Dezember 4714 B.C.

schlecht: Microsoft Excel zählt ab 1.1.1900 die Tage in 16-bit Wort,

$2^{16} : 365 = 179,55068$, ab 31.12.2078 ist Schluß

(4) YY auf CCYY umstellen - Gewaltmethode

Überall Jahresfelder finden und auf vier Stellen erweitern

gilt als sicherste Methode - aufwendig

Datenbankschemata anpassen

Programme neu übersetzen (sofern Quellcode noch vorhanden)

(3) - (1) Auf R/3 umstellen, nichts tun, auswandern, ...

kommt eine Regress- und Klagewelle auf uns zu?

Bankbürgschaften sollen zehnmal so viel kosten wie eine Softwareinspektion ...

Waren zweistellige Jahreszahlen mangelnde Voraussicht oder gar unverantwortlicher Leichtsin?

Wer hatte (hat) ein Auto mit einem fünfstelligen Kilometerzähler? War die Norm vor 1920.

Das White House wies 1967 die amerikansiche Standardisierungsbehörde (National Bureau of Standards) an, den Streit in den Bundesbehörden über den Gebrauch von zweistelligen versus vierstelligen Jahreszahlen beizulegen. Auf Druck des Pentagons beließ man es bei zweistelligen Zahlen!

Tatsache ist, niemand hat in den Sechzigern geglaubt, daß die damals geschriebenen COBOL-Programme noch 20 und mehr Jahre laufen würden.

Wer in den Sechzigern für vierstellige Jahreszahlen plädierte, wurde schief angeschaut.

Plädieren Sie jetzt für fünfstellige Jahreszahlen?

Ab wann hätte jeder Softwarelieferant vierstellige Jahreszahlen einbauen müssen?

Die ISO Norm 8601 legt seit 1988 fest, daß „Jahresangaben vierstellig erfolgen müssen“, sagt ein Rechtsanwalt Graf von Westphalen lt. Information Week vom 22. Juli 99, aber: Norm enthält auch zweistellige Jahreszahlen!

Wo werden Datumsabfragen gemacht?

Botschaftenaustausch (Zeitstempel um vertauschte Pakete zu finden)

Aufräumroutinen (Aufträge löschen, die älter als ein Jahr sind)

Tourenplanung (wenn bis ... fertig, dann mitschicken mit ...)

Verwaltungen (Rentenversicherung, Einschulung, Verkehrsbetriebe, ...)

embedded systems (Automaten)

Geschätzte Größe des Problems

Capers Jones schätzt Kosten der Reparatur auf 70 Milliarden US \$

180 Milliarden Zeilen COBOL-Code auf IBM MVS Anlagen, ca. 900.000 COBOL Programmierer mit Pflege beauftragt, 93% aller Programme enthalten ein datumsrelevantes Feld

geschätzte \$1.25 Aufwand je Zeile COBOL ergäbe 225 Milliarden US \$ Änderungsaufwand

Wahrscheinliche Probleme

Geldautomaten erkennen Euro- oder Kreditkarten nicht, sind abgeschaltet oder leer

datumsgesicherte Safes funktionieren nicht

Telefongespräche in exotische Länder über Silvester dauern 99 Jahre

Türsicherungssysteme versagen und Türen stehen offen

Fahrstühle (Klimaanlage?) funktionieren nicht, weil 99 Jahre nicht gewartet

Erfahrungswert z.B. Frankfurter Flughafen: von 10.000 elektronischen Komponenten im gesamten Flughafen waren 46 für den Datumswechsel nicht tauglich (hauptsächlich Kassenautomaten für das Parkhaus).

Bei der Ver- und Entsorgung von Betriebsstoffen waren von 114 betriebsrelevanten Systemen 60 umzustellen und 10 auszutauschen

Ob irgendwo im kalten Ural ein Oberst mal seine SS20 vom Computer abgehängt hat und probeweise das Datum auf 1.1.2000 vorgestellt hat?