情報処理システム論 (15)

西暦2000年問題

- 西暦の下2桁だけを使ってきた
  - 1997 → 97
- 2000年になると下2桁は00に!
- 2000年まであと3年!

2000年がくると…

- ソートの順序が狂う
- 金利計算が狂う
- 年齢計算が狂う
- 有効期限判定が狂う
  - アクセス権、品質保証問題
- データの消失
  - 古いデータを自動的に消去する場合

一種の時限爆弾

どうやって4桁にしているのか

- “19”を前につけたす
  - 論外
- 1900を加算する
  - 2000年問題は2桁の記憶方法に依存する
  - たいがい、記憶領域は2桁
    - 00に戻る
    - オーバーフロー・エラーになる
  - やっぱり1999年までしか数えられない

その他のバリエーション

- 1980を加算する（たとえば）
  - 100年間は使える
- 2000年は閏年
  - 基本は4で割りきれる年
  - 100で割り切れる年は除外
  - でも400で割り切れる年は含む
  - たぶん大丈夫

2桁の数を記憶する方法

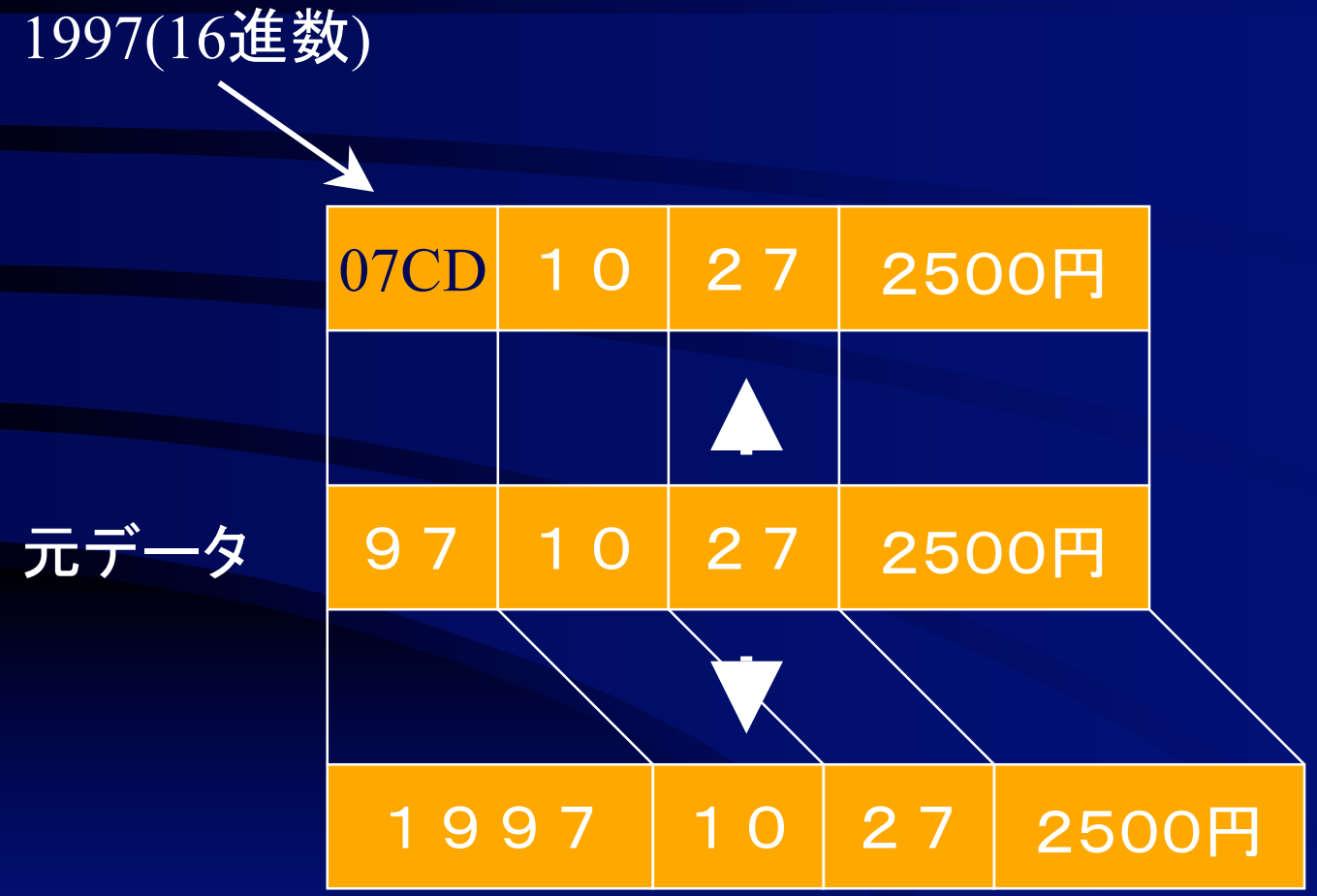
- 1バイトを2進表記で
  - 0 から 255 (FF) までが表現可能
- 1バイトを4ビット×2で (Packed BCD)
- 2バイトに数字文字を2桁分
  - BCD (Binary-Coded Decimal) 2進化10進数
  - 00 から 99 まで
- 2バイトに2進表記で
  - 0 から 65535 (FFFF) まで

4桁の数を記憶するには

- バイト数を増やす
  - 過去のプログラムと錯綜した場合
    - データの配置のずれが問題となる
- 数値の表現形式の変更
  - 過去のプログラムと錯綜した場合
    - データの値が混乱する

どちらにしてもデータの変換が必要

データの互換性



なにが対応していないのか

- ハードウェア
  - 計算機に組み込まれた時計
- OS (Operating System)
  - システム時計の扱い
  - アプリケーションとのやりとり
- アプリケーション

なぜ放置されたのか

- メモリー消費量の節約
  - 昔は非常に高価だった
- プログラムの寿命
  - こんなに長い間使われると予想できなかった
  - 計算機の寿命7年説
  - 5年で減価償却
  - システムのアップグレードによる継続利用
    - コストが安い

なぜ放置されたのか（続き）

- ISO/JISの規格が2桁だった
  - 1989 ANSI/ISO が4桁に
  - 1992 JIS が4桁に
- 入力文字数を減らす
  - 人間工学(?)

対策のための問題

- 企業の規模とプログラム規模
- 過去のデータとの互換性
- プログラマ不足
- 開発や変更のための時間や費用
- 動作確認のための時間
- システムの一斉更新の難しさ

プログラマ不足

- ねおだま
  - ネットワーク
  - オープンシステム
  - ダウンサイジング
  - マルチメディア
- 大型・汎用機技術者の減少
  - パソコンやワークステーションへ

プログラマ不足（続き）

- プログラミング言語の変遷
  - FORTRAN、COBOL から
  - C、Visual Basic などへ
- バブル崩壊後の人員削減
  - 計算機分野からの撤退
- 日本語の扱い
  - 英語圏のプログラマに頼りにくい

FORTRAN

```
      PROGRAM TEST
      INTEGER I, SUM
      SUM = 0
      DO 10 I=1,100
        SUM = SUM + 1
10    CONTINUE
      WRITE (6,100) SUM
100   FORMAT (I8)
      END
```

COBOL

Common Business Oriented Language

```
PROCEDURE DIVISION
    PERFORM VARYING I FROM 1 BY 1
        UNTIL I > 10
        SUM = SUM + I
    END-PERFORM
    STOP RUN.
```

なにが難しいのか

- 自由度の違い
  - C はたくさんの書き方がある
- 複雑なロジックの表現能力が低い
  - ポインタ、再帰呼び出し
- COBOLは画面・キーボード操作が強力
  - Cもライブラリが充実してきている
- 開発環境の差異

なにが難しいのか（続き）

- プログラマ教育
  - 熟練COBOLプログラマの減少
  - OJT (On the Job Training) が困難に
- ドキュメントの不足
  - 開発の経緯や思想、デザインが文書としてしっかり残されていない

別の問題たち

- 平成
  - 昭和から平成に変わった(1989)ときに、2000年問題も一緒に対策した(?)
- UNIXの時計
  - Tue Jan 19 03:14:07 2038
  - 32ビット
  - 00:00:00 January 1, 1970 を基点とする秒数

UNIXの問題の確認プログラム

```
main()
{
        long t = 0x7fffffff;
        printf("%s", ctime(&t));
}
```

卑近な例

アプリケーション		何年までいけるか
MS-DOS/Win95	File System	2099
	32bit FAT	2108
	NTFS	64bit
Excel 5.0/95		2019(2)/2078(4)
Excel 97		2029(2)/9999(4)
Access 2.0/95		1999(2)/9999(4)
1-2-3 97		2099
Notes		4713BC - 3200AD

参考文献

- 西暦2000年問題
  加藤忠宏, 技術評論社
  ISBN 4-7741-0473-6, 1280円