

I117 (12) CSV (その2)

知念

北陸先端科学技術大学院大学 情報科学研究科
School of Information Science,
Japan Advanced Institute of Science and Technology

漢字への拡張

- 文字コード
 - ◇ JIS (ISO-2022-JP)
 - ◇ EUC (EUC-JP)
 - ◇ SJIS
 - ◇ UTF-8
- 今回は EUC のみ対応
- その他漢字への変換はツールに任せる

漢字コード

- ASCII は 0x00–0x7E まで、数百文字
- 漢字は数万文字なので表現できない
 - ◇ バイトでは不可能
- マルチバイトで表現する
 - ◇ 可変長
 - ◇ 固定長

JIS — 基本

JIS X 0201

- ASCII と同じような位置付け
- 文字（カナと記号）を 1バイトで表現（7/8ビット）
- 通称「半角カナ」「1バイトカナ」

JIS X 0208

- 文字（漢字、カナ、記号）を 2バイトで表現
- 区点コード、区→1バイト目、点→2バイト目
- 1978, 83, 90, 97 年に改定されている

※ 関連規格が JIS X 0212, 0213 等多数

ISO-2022JP

- メールなど古くからのサービスに使われる
- ISO 2022 の文字コードの日本向け
- 可変長、漢字の前後に切替えシーケンスがつく
 - ◇ ESC \$ B JIS83 や ESC (B ASCII
 - ◇ ESC \$ @ JIS78 や ESC (J JISローマ字

```
% cat /tmp/v
```

```
演習
```

```
% od -c /tmp/v
```

```
0000000 033  $  B  1  i  =  , 033  (  B  \n
```

```
% od -t x1 /tmp/v
```

```
0000000 1b 24 42 31 69 3d 2c 1b 28 42 0a
```

ISO-2022JP (cont.)

演 習

ESC \$ B 1 i = , ESC (B \n

1b	24	42	31	69	3d	2c	1b	28	42	0a
----	----	----	----	----	----	----	----	----	----	----

- インターナショナル
 - ◇ シーケンスにより、多数の文字や国に対応
- 7ビットで処理するのが主流
- 先頭から読まないと正確に解釈できない

EUC

- 固定長、多くのUNIX で採用
- バイリンガル
 - ◇ 日本向けは EUC-JP
 - ◇ 中国語や韓国語向けでは別の文字
- 8ビット目をセットしたJIS文字

```
% nkf -e /tmp/v
```

演習

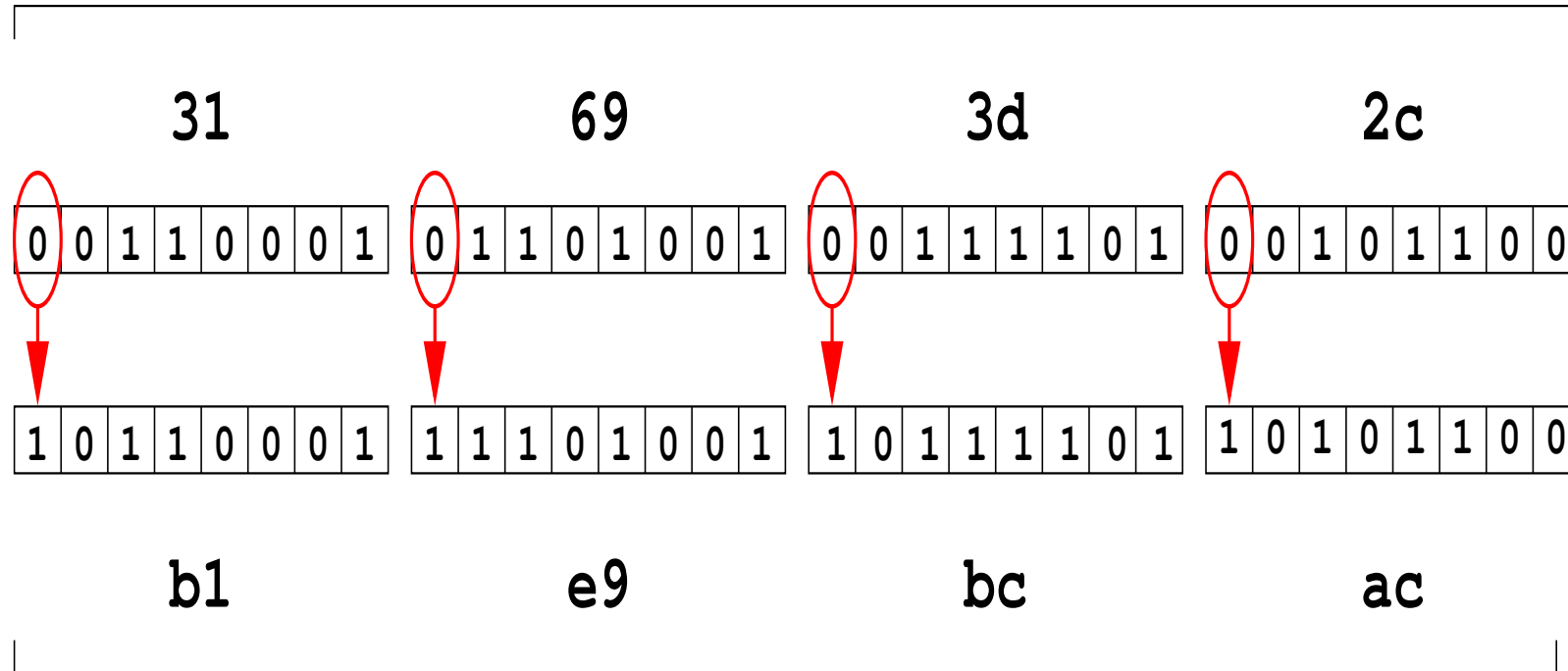
```
% nkf -e /tmp/v|od -t x1
```

```
00000000 b1 e9 bd ac 0a
```

```
00000005
```

JIS から EUC のビット演算 (8ビット目をセット)

JIS code



EUC code

SJIS

- 固定長、パソコンで普及している
- ASCII と 1バイトカナ文字に衝突しないようにした文字体系
- 特別な変換式で JIS から変換する

```
% nkf -s /tmp/v | od -t x1  
00000000 89 89 8f 4b 0a  
00000005
```

- この 4b は ASCII の K にあたる（2バイト目なので問題なし）複雑

UTF-8

- インターナショナルを目指す
- もともとは固定長を目指していた
- 可変長、英数字 1, 漢字 3バイトでほぼ固定
- JIS に準拠していない
- 単純な式ではなく、変換には表が必要

```
% nkf -w /tmp/v | od -t x1  
00000000 e6 bc 94 e7 bf 92 0a  
00000007
```

※ 16ビット単位の UTF-16 も存在する

漢字コード・ポイント

- 固定長か可変長
- JIS 準拠かそれ以外
 - ◇ 多くは JIS 準拠
- バイリンガルかインターナショナルか
 - ◇ 同時にいくつかの言語や文字を扱えるか
 - ◇ L10N: localization
 - ◇ I18N: internationalization

csvparse.c の EUC 対応

- 8ビット目に対応するだけ
 - ◇ 8ビット目を拡張しても衝突するルールはない
- TEXTDATA を拡張

```
TEXTDATA = %x20-21/ %x23-2B/ %x2D-7E/ %x80-FF
```

```
#define ISTEXT ( (ch>=0x20&&ch<=0x21) || \  
                (ch>=0x23&&ch<=0x2b) || \  
                (ch>=0x2d&&ch<=0x7e) || \  
                (ch>=0x80&&ch<=0xff) )
```

- int → char 変換の箇所で 8ビット目を受け付ける

```
if(c+1<vlen && (ch>=0 && ch<=0xff)) {  
    *p++ = ch%255;  
    *p = '\\0';  
    c++;  
}  
else {  
    fprintf(stderr, "ignore charator\\n");  
}
```

- これだけで EUC を扱えるようになる

実行結果

```
% ./pcr1f < vv.csv
"野口英世",1000 LF
"樋口一葉",5000 LF
"福沢諭吉",10000 LF
"夏目漱石",1000 LF
"新渡戸稲造",5000 LF
"聖徳太子",10000
```

⇒

```
% nkf -e vv.csv | ./a.out
0-0 field '野口英世'
0-1 field '1000'
1-0 field '樋口一葉'
1-1 field '5000'
2-0 field '福沢諭吉'
2-1 field '10000'
3-0 field '夏目漱石'
3-1 field '1000'
4-0 field '新渡戸稲造'
4-1 field '5000'
5-0 field '聖徳太子'
5-1 field '10000'
no LF (record separator)
```

演習

1) SJIS を受け付けるよう変更せよ★★★

- SJIS のコード体系が他のルールと衝突しないか確認せよ

※ 漢字フィルタのライブラリやプログラム類を使ってはならない