

Локализация и интернационализация программного обеспечения

Инструментарий Современного Программиста

Иван Трепаков

NSU Sys.Pro

American Standard Code for Information Interchange

```
$ ascii -d
```

0 NUL	16 DLE	32	48 0	64 @	80 P	96 `	112 p
1 SOH	17 DC1	33 !	49 1	65 A	81 Q	97 a	113 q
2 STX	18 DC2	34 "	50 2	66 B	82 R	98 b	114 r
3 ETX	19 DC3	35 #	51 3	67 C	83 S	99 c	115 s
4 EOT	20 DC4	36 \$	52 4	68 D	84 T	100 d	116 t
5 ENQ	21 NAK	37 %	53 5	69 E	85 U	101 e	117 u
6 ACK	22 SYN	38 &	54 6	70 F	86 V	102 f	118 v
7 BEL	23 ETB	39 '	55 7	71 G	87 W	103 g	119 w
8 BS	24 CAN	40 (56 8	72 H	88 X	104 h	120 x
9 HT	25 EM	41)	57 9	73 I	89 Y	105 i	121 y
10 LF	26 SUB	42 *	58 :	74 J	90 Z	106 j	122 z
11 VT	27 ESC	43 +	59 ;	75 K	91 [107 k	123 {
12 FF	28 FS	44 ,	60 <	76 L	92 \	108 l	124
13 CR	29 GS	45 -	61 =	77 M	93]	109 m	125 }
14 SO	30 RS	46 .	62 >	78 N	94 ^	110 n	126 ~
15 SI	31 US	47 /	63 ?	79 O	95 _	111 o	127 DEL

ASCII

00:

10:

20: `!"#$%&'()*+,-./`

30: `0123456789:;<=>?`

40: `@ABCDEFGHIJKLMNO`

50: `PQRSTUVWXYZ[\]^_`

60: ``abcdefghijklmnopqrstuvwxyz`

70: `{|}~`

ASCII

00:	80:
10:	90:
20: !"#\$%&'()*+,-./	a0:
30: 0123456789:;<=>?	b0:
40: @ABCDEFGHIJKLMNO	c0:
50: PQRSTUVWXYZ[\]^_	d0:
60: `abcdefghijklmno	e0:
70: pqrstuvwxyz{ }~	f0:

Расширения ASCII

ASCII

00: 80: АБВГДЕЖЗИЙКЛМНОП
10: 90: РСТУФХЦЧШЩЪЫЬЭЮЯ
20: ̀!"#\$%&'()*+,-./ а0: абвгдежзийклмноп
30: 0123456789:;<=>? б0: 0123456789:;<=>?
40: @ABCDEFGHIJKLMNO с0: ABCDEFGHIJKLMNOP
50: PQRSTUVWXYZ[\]^_ д0: PQRSTUVWXYZ[\]^_
60: `abcdefghijklmno е0: рстуфхцчшщъыьэюя
70: pqrstuvwxyz{|}~ ф0: ЁёЄєӦӧӨӨ°••√№¤■

CP866




MS-DOS

Расширения ASCII

ASCII

CP866

KOI8-R

00:		80:	АБВГДЕЖЗИЙКЛМНОП
10:		90:	РСТУФХЦЧШЩЪЫЬЭЮЯ
20:	!"#\$%&'()*+,-./	a0:	абвгдежзийклмноп
30:	0123456789:;<=>?	b0:	 { }~`^_!@#%&'()*+,-./
40:	@ABCDEFGHIJKLMNO	c0:	 { }~`^_!@#%&'()*+,-./
50:	PQRSTUVWXYZ[\]^_	d0:	 { }~`^_!@#%&'()*+,-./
60:	`abcdefghijklmno	e0:	рстуфхцчшщъыьэюя
70:	pqrstuvwxyz{ }~	f0:	ЁёЄєİïŸŸ°•√№¤■

[illegible]

MS-DOS

Unix

Расширения ASCII

ASCII

00:
10:
20: _!"#\$%&'()*+,-./
30: 0123456789:;<=>?
40: @ABCDEFGHIJKLMNO
50: PQRSTUVWXYZ[\]^_
60: `abcdefghijklmnopqrstuvwxyz
70: pqrstuvwxyz{|}~

CP866

80: АБВГДЕЖЗИЙКЛМНОП
90: РСТУФХЦЧШЩЪЫЬЭЮЯ
a0: абвгдежзийклмноп
b0:
c0:
d0:
e0: рстуфхцчшщъыьэюя
f0: ЁёЄєӢӓӮӹ••√№¤■

MS-DOS

KOI8-R

-| □ □ □ □ □ □ □
■ ■ • √ ≈ ≤ ≥ ∫ ° ² ∙ ÷
= || ф ё г п т ч е љ џ ѣ
|| | Ё || | т т т т т т т
ю абцдефгхйклмно
пярстужввызшэщчъ
ЮАБЦДЕФГХИЙКЛМНО
ПЯРСТУЖВВЫЗШЭЩЧЪ

Unix

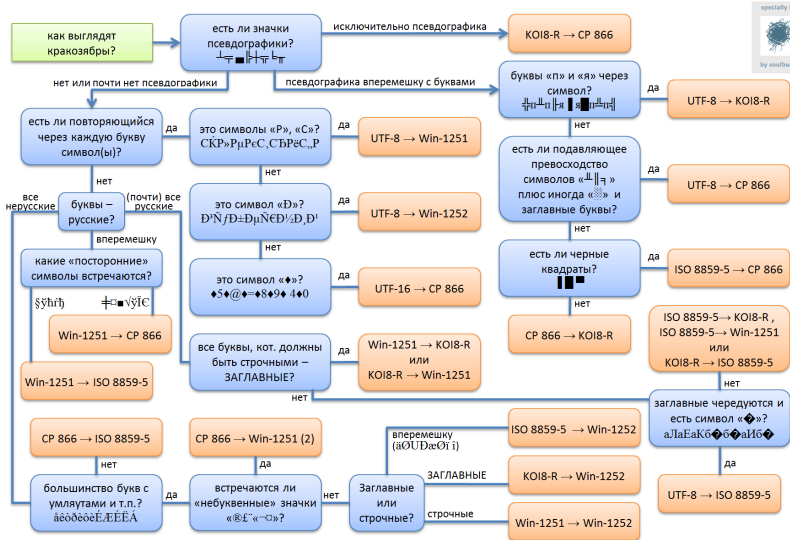
CP1251

ђѓ,ѓ„…†‡€%&‹›ќћџ
ђ' ' ' ' • — ™ љ › њ љ џ
ЎўЈѡГЃ\$ЁЄ«¬-®Ї
°±Іігµ¶•ё№є»јSsї
АБВГДЕЖЗИЙКЛМНОП
РСТУФХЦЧШЩЪЫЬЭЮЯ
абвгдежзийклмноп
рстуфхцчшщъыьэюя

Windows

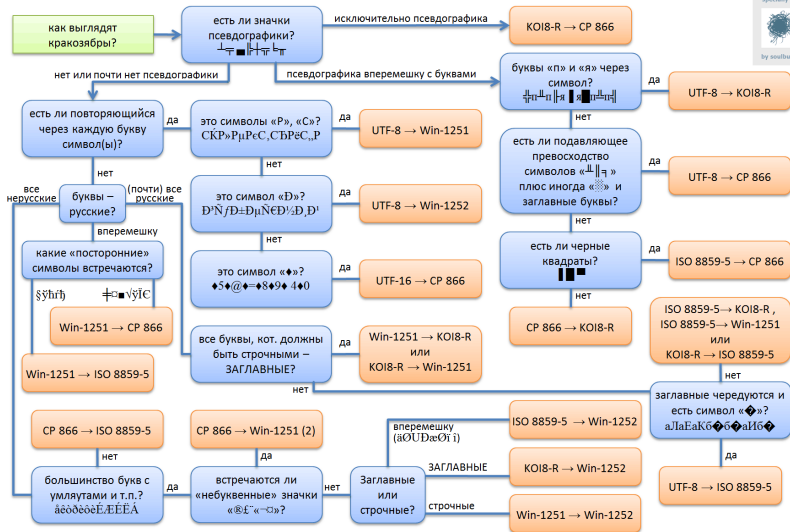
опХБЕР ЛХП!

ОПХБЕР ЛХП!



оПХБЕР ЛХП!

```
$ echo "оПХБЕР ЛХП!" \  
| iconv -t cp1251 \  
| iconv -f koi8-r  
Ноуап ку!
```

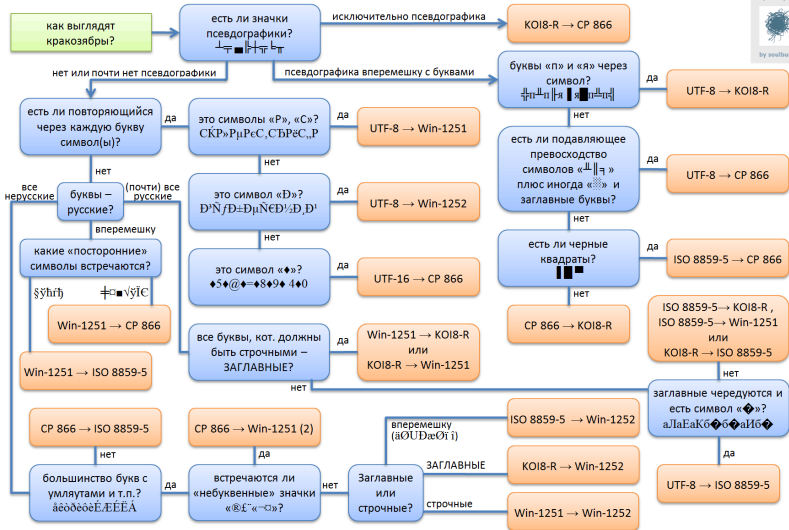


Крокозябры

ОПХБЕР ЛХП!

```
$ echo "оПХБЕР ЛХП!" \
  | iconv -t cp1251 \
  | iconv -f koi8-r
Норзеп кyo!
```

```
$ echo "оПХБЕР ЛХП!" \
  | iconv -t koi8-r \
  | iconv -f cp1251
Привет мир!
```



Unicode Standard

- Универсальное кодирование символов
- Количество различных кодов 1 114 112
- **Unicode 16.0** (2024 год) использует лишь 154 998

Цели

- **Универсальность** Содержит все возможные символы современных и древних языков, технических текстов, диакритику и emoji
- **Эффективность** *Plain text* кодирование в одном из трех стандартных форматов: **UTF-32**, **UTF-16**, **UTF-8**
- **Однозначность** Каждый код *однозначно* соответствует единственному символу



Code points

- Кодируют “абстрактные символы”
- Целые числа от 0 до $10FFFF_{16}$
- Стандартно записываются в hex с префиксом U+
- Имеют уникальные имена

Code points

- Кодируют “абстрактные символы”
- Целые числа от 0 до $10FFFF_{16}$
- Стандартно записываются в hex с префиксом U+
- Имеют уникальные имена

Инструменты

- `uname` (часть пакета `uniutils`)
- Модуль `unicodedata` в Python

```
import sys, unicodedata as U

for i, c in enumerate(sys.stdin.read()):
    s = '○' if U.category(c)[0] == 'M' else ''
    print('U+{0:04X}\t{1}{2}\t{3}'
          .format(ord(c), s, c, U.name(c)))
```


Code points

- Кодируют “абстрактные символы”
- Целые числа от 0 до $10FFFF_{16}$
- Стандартно записываются в hex с префиксом U+
- Имеют уникальные имена

Hello 🌍!

Инструменты

- `uname` (часть пакета `uniutils`)
- Модуль `unicodedata` в Python

```
import sys, unicodedata as U

for i, c in enumerate(sys.stdin.read()):
    s = '☹' if U.category(c)[0] == 'M' else ''
    print('U+{0:04X}\t{1}{2}\t{3}'
          .format(ord(c), s, c, U.name(c)))
```

Code points

- Кодируют “абстрактные символы”
- Целые числа от 0 до $10FFFF_{16}$
- Стандартно записываются в hex с префиксом U+
- Имеют уникальные имена

Инструменты

- `uname` (часть пакета `uniutils`)
- Модуль `unicodedata` в Python

```
import sys, unicodedata as U

for i, c in enumerate(sys.stdin.read()):
    s = '○' if U.category(c)[0] == 'M' else ''
    print('U+{0:04X}\t{1}{2}\t{3}'
          .format(ord(c), s, c, U.name(c)))
```

Hello 🌍!

U+0048	H	LATIN CAPITAL LETTER H
U+0065	e	LATIN SMALL LETTER E
U+006C	l	LATIN SMALL LETTER L
U+006C	l	LATIN SMALL LETTER L
U+006F	o	LATIN SMALL LETTER O
U+0020		SPACE
U+1F30E	🌍	EARTH GLOBE AMERICAS
U+0021	!	EXCLAMATION MARK

Code points

- Кодируют “абстрактные символы”
- Целые числа от 0 до $10FFFF_{16}$
- Стандартно записываются в hex с префиксом U+
- Имеют уникальные имена

Инструменты

- `uname` (часть пакета `unutils`)
- Модуль `unicodedata` в Python

```
import sys, unicodedata as U

for i, c in enumerate(sys.stdin.read()):
    s = '⬤' if U.category(c)[0] == 'M' else ''
    print('U+{0:04X}\t{1}{2}\t{3}'
          .format(ord(c), s, c, U.name(c)))
```

Hello 🌍!

```
$ printf 'Hello \U1F30E!' | uname
```

U+0048	H	LATIN CAPITAL LETTER H
U+0065	e	LATIN SMALL LETTER E
U+006C	l	LATIN SMALL LETTER L
U+006C	l	LATIN SMALL LETTER L
U+006F	o	LATIN SMALL LETTER O
U+0020		SPACE
U+1F30E	🌍	EARTH GLOBE AMERICAS
U+0021	!	EXCLAMATION MARK

Code points

- Кодируют “абстрактные символы”
- Целые числа от 0 до $10FFFF_{16}$
- Стандартно записываются в hex с префиксом U+
- Имеют уникальные имена

Инструменты

- `uname` (часть пакета `uniutils`)
- Модуль `unicodedata` в Python

```
import sys, unicodedata as U

for i, c in enumerate(sys.stdin.read()):
    s = '☺' if U.category(c)[0] == 'M' else ''
    print('U+{0:04X}\t{1}{2}\t{3}'
          .format(ord(c), s, c, U.name(c)))
```

Привет 🌍!

Code points

- Кодировать “абстрактные символы”
- Целые числа от 0 до $10FFFF_{16}$
- Стандартно записываются в hex с префиксом U+
- Имеют уникальные имена


Инструменты

- `uname` (часть пакета `unicutils`)
- Модуль `unicodedata` в Python

```
import sys, unicodedata as U

for i, c in enumerate(sys.stdin.read()):
    s = '○' if U.category(c)[0] == 'M' else ''
    print('U+{0:04X}\t{1}{2}\t{3}'
          .format(ord(c), s, c, U.name(c)))
```

Привет !

U+041F	П	CYRILLIC CAPITAL LETTER PE
U+0440	р	CYRILLIC SMALL LETTER ER
U+0438	и	CYRILLIC SMALL LETTER I
U+0432	в	CYRILLIC SMALL LETTER VE
U+0435	е	CYRILLIC SMALL LETTER IE
U+0442	т	CYRILLIC SMALL LETTER TE
U+0020		SPACE
U+1F30E		EARTH GLOBE AMERICAS
U+0021	!	EXCLAMATION MARK

Code points

- Кодировать “абстрактные символы”
- Целые числа от 0 до $10FFFF_{16}$
- Стандартно записываются в hex с префиксом U+
- Имеют уникальные имена

Инструменты

- `uname` (часть пакета `unicutils`)
- Модуль `unicodedata` в Python

```
import sys, unicodedata as U

for i, c in enumerate(sys.stdin.read()):
    s = '○' if U.category(c)[0] == 'M' else ''
    print('U+{0:04X}\t{1}{2}\t{3}'
          .format(ord(c), s, c, U.name(c)))
```

Привет 🌍!

```
$ printf 'Привет \U1F30E!' | uname
```

U+041F	П	CYRILLIC CAPITAL LETTER PE
U+0440	р	CYRILLIC SMALL LETTER ER
U+0438	и	CYRILLIC SMALL LETTER I
U+0432	в	CYRILLIC SMALL LETTER VE
U+0435	е	CYRILLIC SMALL LETTER IE
U+0442	т	CYRILLIC SMALL LETTER TE
U+0020		SPACE
U+1F30E	🌍	EARTH GLOBE AMERICAS
U+0021	!	EXCLAMATION MARK

Code points

- Кодируют “абстрактные символы”
- Целые числа от 0 до $10FFFF_{16}$
- Стандартно записываются в hex с префиксом U+
- Имеют уникальные имена

Йо-йо

Инструменты

- `uname` (часть пакета `uniutils`)
- Модуль `unicodedata` в Python

```
import sys, unicodedata as U

for i, c in enumerate(sys.stdin.read()):
    s = '○' if U.category(c)[0] == 'M' else ''
    print('U+{0:04X}\t{1}{2}\t{3}'
          .format(ord(c), s, c, U.name(c)))
```

Code points

- Кодируют “абстрактные символы”
- Целые числа от 0 до $10FFFF_{16}$
- Стандартно записываются в hex с префиксом U+
- Имеют уникальные имена

Инструменты

- `uname` (часть пакета `unutils`)
- Модуль `unicodedata` в Python

```
import sys, unicodedata as U

for i, c in enumerate(sys.stdin.read()):
    s = '○' if U.category(c)[0] == 'M' else ''
    print('U+{0:04X}\t{1}{2}\t{3}'
          .format(ord(c), s, c, U.name(c)))
```

Йо-йо

U+0439	Й	CYRILLIC CAPITAL LETTER SHORT I
U+043E	о	CYRILLIC SMALL LETTER O
U+002D	-	HYPHEN-MINUS
U+0438	и	CYRILLIC SMALL LETTER I
U+0306	̂	COMBINING BREVE
U+043E	о	CYRILLIC SMALL LETTER O

Code points

- Кодируют “абстрактные символы”
- Целые числа от 0 до $10FFFF_{16}$
- Стандартно записываются в hex с префиксом U+
- Имеют уникальные имена

Инструменты

- `uname` (часть пакета `unutils`)
- Модуль `unicodedata` в Python

```
import sys, unicodedata as U

for i, c in enumerate(sys.stdin.read()):
    s = '○' if U.category(c)[0] == 'M' else ''
    print('U+{0:04X}\t{1}{2}\t{3}'
          .format(ord(c), s, c, U.name(c)))
```

Йо-йо

```
$ printf 'Йо-и\u0306o' | uname
```

U+0439	Й	CYRILLIC CAPITAL LETTER SHORT I
U+043E	о	CYRILLIC SMALL LETTER O
U+002D	-	HYPHEN-MINUS
U+0438	и	CYRILLIC SMALL LETTER I
U+0306	̂	COMBINING BREVE
U+043E	о	CYRILLIC SMALL LETTER O

Code points



- Кодируют “абстрактные символы”
- Целые числа от 0 до $10FFFF_{16}$
- Стандартно записываются в hex с префиксом U+
- Имеют уникальные имена

Инструменты

- `uname` (часть пакета `uniutils`)
- Модуль `unicodedata` в Python

```
import sys, unicodedata as U

for i, c in enumerate(sys.stdin.read()):
    s = '○' if U.category(c)[0] == 'M' else ''
    print('U+{0:04X}\t{1}{2}\t{3}'
          .format(ord(c), s, c, U.name(c)))
```

Code points



- Кодируют “абстрактные символы”
- Целые числа от 0 до $10FFFF_{16}$
- Стандартно записываются в hex с префиксом U+
- Имеют уникальные имена

Инструменты

- `uname` (часть пакета `uniutils`)
- Модуль `unicodedata` в Python

```
import sys, unicodedata as U

for i, c in enumerate(sys.stdin.read()):
    s = '♂' if U.category(c)[0] == 'M' else ''
    print('U+{0:04X}\t{1}{2}\t{3}'
          .format(ord(c), s, c, U.name(c)))
```

U+1F468	👨	MAN
U+200D		ZERO WIDTH JOINER
U+1F469	👩	WOMAN
U+200D		ZERO WIDTH JOINER
U+1F467	👧	GIRL

Code points

- Кодируют “абстрактные символы”
- Целые числа от 0 до $10FFFF_{16}$
- Стандартно записываются в hex с префиксом U+
- Имеют уникальные имена

Инструменты

- `uname` (часть пакета `uniutils`)
- Модуль `unicodedata` в Python

```
import sys, unicodedata as U

for i, c in enumerate(sys.stdin.read()):
    s = '♂' if U.category(c)[0] == 'M' else ''
    print('U+{0:04X}\t{1}{2}\t{3}'
          .format(ord(c), s, c, U.name(c)))
```



```
$ printf '\U1F468\u200D\U1F469\u200D\U1F467' \
| uname
```

U+1F468	👨	MAN
U+200D		ZERO WIDTH JOINER
U+1F469	👩	WOMAN
U+200D		ZERO WIDTH JOINER
U+1F467	👧	GIRL

Q&A