

# Стандарти представлення двобайтових символів.

Презентацію підготував  
Студент групи 200008 бск  
Масюк Д. В.

Unicode

# UCS-2, UCS-4(*Universal Character Set*)

ASCII 00010101

UCS-2 00000000 00010101

UCS-4 00000000 00000000 00000000 00010101

UTF-8,  
UTF-16,  
UTF-16LE FF FE,  
UTF-16BE FE FF,  
UTF-32,  
UTF-32LE FF FE 00 00,  
UTF-32BE 00 00 FE FF.

# Стандарт кодування UTF-8

Старший біт зліва. Початком коду є керуючий символ (виділено **жирним**)

- ◇ **0**xxxxxxx
- ◇ **110**xxxxx 10xxxxxx
- ◇ **1110**xxxx 10xxxxxx 10xxxxxx
- ◇ **11110**xx 10xxxxxx 10xxxxxx 10xxxxxx
  
- ◇ **0** - використовується 8-бітна кодування,
- ◇ **110** - використовується 16-бітна кодування,
- ◇ **1110** - використовується 24-бітна кодування,
- ◇ **11110** - використовується 32 бітна кодування.

# Приклад кодування фрази «Самостійна 1» в UTF-8.

Код в бінарному вигляді (старший біт ліворуч):

**11010000** 10100001 (С) **11010000** 10110000 (а) **11010000** 10111100 (м)

**11010000** 10111110 (о) **11010001** 10000001 (с) **11010001** 10000010 (т)

**11010001** 10010110 (і) **11010000** 10111001 (й) **11010000** 10111101(н)

**11010000** 10110000 (а) **00**100000 (пробіл) **00**110001 (1)

# Приклад кодування фрази «Самостійна 1» в UTF-16LE.

Код в бінарному вигляді (старший біт ліворуч):

11111111 11111110 (Показчик) 00100001 00000100 (С) 00110000 00000100 (а) 00111100  
00000100 (м) 00111110 00000100 (о) 01000001 00000100 (с)

01000010 00000100 (т) 01010110 00000100 (і) 00111001 00000100 (й)

00111101 00000100 (н) 00110000 00000100 (а) 00100000 00000000 (пробіл)

00110001 00000000 (1)

Розглянемо докладніше алгоритм кодування символів, номери яких перевищують значення 65535. Для прикладу в якості символу використовуємо літеру древнетюркського алфавіту





Номер запропонованого символу в таблиці Юнікод - (0x10C0C).

Алгоритм перетворення номера символу в код UTF-16 складається з 5 кроків:

- ◆ 1. З значення номера символу відняти число 0x10000. Дана операція дозволяє привести розмірність бінарного представлення номера символу до 20 бітам. Для запропонованого символу отримаємо:  $0x10C0C - 0x10000 = 0xC0C$ .
- ◆ 2. Для отриманого значення виділити старші 10 біт і молодші 10 біт. У прикладі число 0xC0C в бінарному вигляді представляється, як **00000000110000001100**, де жирним виділені 10 старших біт, а підкресленням - 10 молодших.
- ◆ 3. До шістнадцятиричним значенням 0xD800 (11011000 00000000) додати значення **0x03** (**00000000 00000011**), сформований 10 старшими бітами, отриманими на попередньому етапі.  $0xD800 + 0x03 = 0xD803$  (11011000 0000**0011**) - 16 старших біт кодового слова UTF-16.
- ◆ 4. До шістнадцятиричним значенням 0xDC00 (11011000 00000000) додати значення 0x0C (00000000 00001100), сформований 10 молодшими бітами, отриманими на кроці №2.  $0xDC00 + 0x0C = DC0C$  (11011100 00001100) - 16 молодших біт кодового слова UTF-16.
- ◆ 5. Кодова слово UTF-16, відповідне символу в прикладі, формується з біт, отриманих на кроках 3 і 4: 0xD803DC0C (11011000 00000011 11011100 00001100).