МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ



**Дніпропетровський національний університет  
залізничного транспорту імені академіка В. Лазаряна**

Кафедра «Технічна Кібернетика»

**Лабораторна робота №3**

**з дисципліни «архітектура комп’ютера»**

**на тему: «Системи числення. Представлення даних в пам'яті комп'ютера»**

Виконав:

студент гр.ПЗ1911

Сафонов Д.Є.

Прийняла: ас. Каф. КІТ

Нежуміра О.І.

Дніпро, 2019

Тема. Системи числення. Представлення даних в пам'яті комп'ютера.

Мета. Вивчити правила переводу чисел з десяткової в двійкову та шістнадцяткову системи числення та правила представлення даних в пам'яті комп'ютера. Набути практичних навичок подання даних у відповідності до типу.

Задача: Виконати дослідження переводу чисел з десяткової в двійкову та шістнадцяткову системи числення. Дати їх внутрішнє (машинне) подання в знакових і беззнакових форматах типів байт, слово та подвійне слово (машинне подання даних повинне бути наведено в двійкової і шістнадцяткової системах числення). Виконати кодування тексту за допомогою таблиці ASCII в десяткової і шістнадцяткової системах числення. Розробити програму опису числових і символьних даних мовою асемблеру й одержати лістинг програми. Варіант 18

|  |  |  |
| --- | --- | --- |
| 18. | 222, -94, 45005, -20351, 683617, -193758 | Don’t bite off more than you can chew.  -99.378 |

# Протокол перетворення чисел з десяткової до двійкової і шістнадцяткової систем числення та представлення чисел у внутрішньому (машинному) форматі

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Decimal | Hexadecimal | Binary | Byte | Word | Doubleword |
| 222 | DE | D E  ↓ ↓  1101 1110 | DE | 00DE | 0000 00DE |
| -94 | -5E | 5 E  ↓ ↓  Binary -0101 1110  Inverse 1010 0001  Complement 1010 0010  ↓ ↓  A 2 | A2 | FFA2 | FFFF FFA2 |
| 45005 | AFCD | A F C D  ↓ ↓ ↓ ↓  1010 1111 1100 1101 | - | AFCD | 0000 AFCD |
| -20351 | -4F7F | 4 F 7 F  ↓ ↓ ↓ ↓  Binary 0100 1111 0111 1111  Inverse 1011 0000 1000 0000  Complement 1011 0000 1000 0001  ↓ ↓ ↓ ↓  B 0 8 1 | - | B081 | FFFF B081 |
| 683617 | D:\1uno1\diit\0000000004.png  A6E61 | A 6 E 6 1  ↓ ↓ ↓ ↓ ↓  1010 0110 1110 0110 0001 | - | - | 000A 6E61 |
| -193758 | D:\1uno1\diit\0000000005.png  -2F4DE | 2 F 4 D E  ↓ ↓ ↓ ↓ ↓  Binary 0010 1111 0100 1101 1110  Inverse 1101 0000 1011 0010 0001  Compl. 1101 0000 1011 0010 0010  ↓ ↓ ↓ ↓ ↓  D 0 B 2 2 | - | - | FFFD 0B22 |

# Представлення тексту у вигляді ASCII-кодів

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Text | D | o | n | ‘ | t |  | b | i | t | e |  | o | f | f |  |
| ASCII | 68 | 111 | 110 | 39 | 116 | 32 | 98 | 105 | 116 | 101 | 32 | 111 | 102 | 102 | 32 |
| Hex | 44 | 6f | 6e | 27 | 74 | 20 | 62 | 69 | 74 | 65 | 20 | 6f | 66 | 66 | 20 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Text |  | t | h | a | n |  | y | o | u |  | c | a | n |
| ASCII | 32 | 116 | 104 | 97 | 110 | 32 | 121 | 111 | 117 | 32 | 99 | 97 | 110 |
| Hex | 20 | 74 | 68 | 61 | 6e | 20 | 79 | 6f | 75 | 20 | 63 | 61 | 6e |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | c | h | e | w | . |
| 32 | 99 | 104 | 101 | 119 | 46 |
| 20 | 63 | 68 | 65 | 77 | 2e |

# Лістинг програми та його аналіз

Turbo Assembler Version 3.0 09/15/19 13:51:36 Page 1

01337.asm

1 ﻿;NAME CODE OPERAND

2

3 ;сегмент стека

4 0000 sseg segment stack 'stack'

5 0000 80\*(??) db 128 dup(?)

6 0080 sseg ends

7

8 ; DATA WRITING TO SEGMENT

9 0000 dseg segment 'data'

10

11 ; текст

12 0000 44 6F 6E E2 80 99 74+ db "Don’t bite off more than you can chew.", '-99.378'

13 20 62 69 74 65 20 6F+

14 66 66 20 6D 6F 72 65+

15 20 74 68 61 6E 20 79+

16 6F 75 20 63 61 6E 20+

17 63 68 65 77 2E 2D 39+

18 39 2E 33 37 38

19 ; BYTE

20 002F DE dbA db 222

21 0030 A2 dbB db -94

22 ; WORD

23 0031 00DE dwA dw 222

24 0033 FFA2 dwB dw -94

25 0035 AFCD dwC dw 45005

26 0037 B081 dwD dw -20351

27 ; DOUBLE WORD

28 0039 000000DE ddA dd 222

29 003D FFFFFFA2 ddB dd -94

30 0041 0000AFCD ddC dd 45005

31 0045 FFFFB081 ddD dd -20351

32 0049 000A6E61 ddE dd 683617

33 004D FFFD0B22 ddF dd -193758

34 0051 dseg ends ;DATA WRITING END

35

36 ;CODE SEGMENT

37 0000 cseg segment 'code'

38 assume cs:cseg, ds:dseg, ss:sseg

39

40 ;мітка початку програми з ім’ям start

41 0000 B8 0000s start: mov ax, dseg

42 0003 8E D8 mov ds, ax

43 ;повернення керування ОС

44 0005 B4 4C mov ah, 4Ch

45 0007 CD 21 int 21h

46 0009 cseg ends

47 end start ;END

Turbo Assembler Version 3.0 09/15/19 13:51:36 Page 2

Symbol Table

Symbol Name Type Value

??DATE Text "09/15/19"

??FILENAME Text "01337 "

??TIME Text "13:51:36"

??VERSION Number 0300

@CPU Text 0101H

@CURSEG Text CSEG

@FILENAME Text 01337

@WORDSIZE Text 2

DBA Byte DSEG:002F

DBB Byte DSEG:0030

DDA Dword DSEG:0039

DDB Dword DSEG:003D

DDC Dword DSEG:0041

DDD Dword DSEG:0045

DDE Dword DSEG:0049

DDF Dword DSEG:004D

DWA Word DSEG:0031

DWB Word DSEG:0033

DWC Word DSEG:0035

DWD Word DSEG:0037

START Near CSEG:0000

Groups & Segments Bit Size Align Combine Class

CSEG 16 0009 Para none CODE

DSEG 16 0051 Para none DATA

SSEG 16 0080 Para Stack STACK

Перший стовпчик - номер рядка, другий - зсув даних, записаних в рядку, щодо початку сегмента (скільки байтів займають дані попередніх рядків даного сегмента), третій - значення даних в шістнадцятковій системі, далі сам код програми, наприклад такі команди: як db, dw , dd, dq, dt - ці команди записують дані займаючи вказану кількість байтів (1, 2, 4, 8, 10); Один байт містить в собі 8 біт (1 біт - інформація приймаюча тільки два значення: 0 і 1), а значить один байт може приймати 2 ^ 8 = 256 значень, тобто числа 0-255 або ж якщо використовувати додатковий код числа від -128 до 127, проміжок чисел які можна записати в кожну із зазначених директив можна вирахує за формулою (2 ^ n) -1, де n - кількість байт в даній директиві, якщо записати число виходить за рамки даних проміжків або станеться помилка, або ви отримаєте число X-a\*2 ^ n (де a – найбільше ціле число при X>a\*2^n), через те, що запис цілих чисел в машинний код має циклічну систему тобто якщо ви захочете записати число 9 в один байт отримаєте 1111 + 2 = 0000 + 1 = 0001 = 1.

# Висновок

В ході виконання лабораторної роботи перевів числа з десяткової в двійкову і шістнадцяткову системи і дав їх у машинному форматі, виконав переклад тексту в машинний код за стандартом ASCII. Також виконав програму яка виконує перераховані вище дії автоматично і провів аналіз лістингу даної програми.