

Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Лабораторна робота №1

з дисципліни
«Системне програмування»

на тему
«Внутрішнє представлення цілих і дійсних даних у процесорі ix86»

Виконав:

студент групи ІП-93

Домінський Валентин Олексійович

номер залікової книжки: 9311

номер у списку: 9

Перевірив:

Павлов Валерій Георгійович

Мета:

Вивчення форматів внутрішнього представлення цілих і дійсних чисел шляхом порівняння результатів, отриманих теоретично, з результатами, що містяться в отриманому розширеному лістингу програми

Порядок виконання роботи:

1. Сформувані наступні початкові числа на основі дати народження студента, записаної у форматі dd.mm.yyyу:
 - A – ціле двозначне число, що дорівнює dd;
 - B – ціле чотиризначне число, яке дорівнює ddm, записаним без роздільника;
 - C – ціле восьмизначне число, яке дорівнює ddmtyyyу, записаним без роздільника;
 - D – число, отримане шляхом ділення числа A на число N, де N - чотиризначний номер залікової книжки;
 - E – число, отримане шляхом ділення числа B на число N;
 - F - число, отримане шляхом ділення числа C на число N;(при діленні округлювати результат до 3 знаків після коми)

Дата народження студента: 22.02.2002

Отже:

- $A = 22$
- $B = 2202$
- $C = 22022002$
- $N = 9311$
- $D = A / N = 22 / 9311 = 0.002$
- $E = B / N = 2202 / 9311 = 0.236$
- $F = C / N = 22022002 / 9311 = 2365.16$

2. Представити отримані числа в двійковій системі числення (дробову частину округлювати до 10 знаків після коми).

Отже:

- $A = 10110$
- $B = 100010011010$
- $C = 1010100000000011101110010$
- $N = 10010001011111$

- $D = 0.0000000010$

0.002	*2
0.004	0
0.008	0
0.016	0
0.032	0
0.064	0
0.128	0
0.256	0
0.512	0
1.024	1
0.048	0
0.096	0

- $E = 0.0011110001$

0.236	*2
0.472	0
0.944	0
1.888	1
1.776	1
1.552	1
1.104	1
0.208	0
0.416	0
0.832	0
1.664	1
1.328	1

- $F = 100100111101.0010100011$

0.16	*2
0.32	0
0.64	0
1.28	1
0.56	0
1.12	1
0.24	0
0.48	0
0.96	0
1.92	1
1.84	1
1.68	1

3. За допомогою розрядної сітки показати в звіті представлення цілих чисел в наступних форматах:

- “ddmmууу” у вигляді символьного рядка;
- числа A и $-A$ у однобайтовому форматі Byte;
- числа A , B , $-A$ и $-B$ у двобайтовому форматі Word;
- числа A , B , C , $-A$, $-B$ и $-C$ у чотирьохбайтовому форматі ShortInt;
- числа A , B , C , $-A$, $-B$ и $-C$ у восьмибайтовому форматі LongInt;

1) “2202200” у вигляді символьного рядка

Число	Символ
2	32
2	32
0	30
2	32
2	32
0	30
0	30

2) числа $A = 22$ й $-A = -22$ у однобайтовому форматі Byte:

Число	Двійковий вигляд
A	0001 0110
$-A$	1110 1010

3) числа $A = 22$, $B = 2202$, $-A = -22$ й $-B = -2202$ у двобайтовому форматі Word

Число	Двійковий вигляд
A	0000 0000 0001 0110
$-A$	1111 1111 1110 1010
B	0000 1000 1001 1010
$-B$	1111 0111 0110 0110

- 4) числа $A = 22$, $B = 2202$, $C = 22022002$, $-A = -22$, $-B = -2202$ й $-C = -22022002$ у чотирьохбайтовому форматі ShortInt

Число	Двійковий вигляд
A	0000 0000 0000 0000 0000 0000 0001 0110
-A	1111 1111 1111 1111 1111 1111 1110 1010
B	0000 0000 0000 0000 0000 1000 1001 1010
-B	1111 1111 1111 1111 1111 0111 0110 0110
C	0000 0001 0101 0000 0000 0111 0111 0010
-C	1111 1110 1010 1111 1111 1000 1000 1101

- 5) числа $A = 22$, $B = 2202$, $C = 22022002$, $-A = -22$, $-B = -2202$ й $-C = -22022002$ у восьмибайтовому форматі LongInt

Число	Двійковий вигляд
A	0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0001 0110
-A	1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1110 1010
B	0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 1000 1001 1010
-B	1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 0111 0110 0110
C	0000 0000 0000 0000 0000 0000 0000 0000 0000 0001 0101 0000 0000 0111 0111 0010
-C	1111 1111 1111 1111 1111 1111 1111 1111 1111 1110 1010 1111 1111 1000 1000 1101

4. Представити числа $D = 0.002$, $E = 0.236$ и $F = 2365.16$ у нормалізованому вигляді:

Щоб використовувати дійсні числа їх треба нормалізувати. Робиться це за допомогою такої формули:

$$A = ZN * M * N^q$$

Де: ZN – знак числа;

M – мантіса числа;

N – основа системи числення;

q – показник;

Візьмемо для початку число $D = 0.002$, переведемо його в двійкову систему числення:

0 – це 0, а 0.002 – це 0.0000000010.

Отже $0.002 = 0.0000000010b$;

Тепер Нам треба здвинути всі числа зліва окрім одної направо або ж навпаки:

$$0.0000000010b = 1.0 * 2^{-9}$$

Таким чином отримуємо:

$$ZN = 0, M = 0.0, N = 2, q = -9$$

Робимо те саме з числами $E = 0.236$ та $F = 2365.16$:

$E = 0.236$:

0 – це 0, а 0.236 – це:

0.0011110001.

Отже $0.236 = 0.0011110001b$

$$0.0011110001b = 1.1110001 * 2^{-3}$$

$$ZN = 0, M = 1.1110001, N = 2, q = -3$$

$F = 2365.16$:

2365 – це 100100111101, а 0.16 – це:

0.0010100011.

Отже $2365.16 = 100100111101.0010100011b$

$$100100111101.0010100011b = 1.001001111010010100011 * 2^{11}$$

$$ZN = +1, M = 1.001001111010010100011, N = 2, q = 11$$

5. За допомогою розрядної сітки показати в звіті представлення дійсних чисел в наступних форматах:

- числа $D = 0.002$ и $-D = -0.002$ в у чотирьохбайтовому форматі Single (float);
- числа $E = 0.236$ и $-E = -0.236$ у восьмибайтовому форматі Double (double);
- числа $F = 2365.16$ и $-F = -2365.16$ у десятибайтовому форматі Extended (long double);

1) Числа $D = 0.002$ и $-D = -0.002$ в у чотирьохбайтовому форматі Single (float):

Додатне число $D = 0.002$

Нормалізоване $= 1.0 * 2^{-9}$

Знак: 0 – додатне

Порядок: До показника q додаємо $127 - 9 + 127 = 118_{10} = 1110110_2$

Мантіса: $1.0 = 0$

Тепер розміщуємо це у 32 бітах за стандартом IEEE 754:

31	30	23	22	0
0	0111 0110	000 0000 0000 0000 0000 0000		
Знак	Порядок	Мантіса		

Відповідь $= 0011\ 1011\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000 =$
3B000000

Від'ємне число $-D = -0.002$

Знак: 1 – від'ємне

Усе інше залишається тим самим:

31	30	24	23	0
1	1111 111	0000 0000 1000 0000 0000 0000		
Знак	Порядок	Мантіса		

Відповідь = 1011 1011 0000 0000 0000 0000 0000 0000 =
BB000000

2) Числа $E = 0.236$ и $-E = -0.236$ у восьмибайтовому форматі
Double (double):

Додатне число $E = 0.236$

Нормалізоване = $1.1110001 \cdot 2^{-3}$

Знак: 0 – додатне

Порядок: До показника q додаємо $1023 - 3 + 1023 = 1020_{10} =$
 1111111100_2

Мантіса: $1.1110001 = 1110001$

Тепер розміщуємо це у 64 бітах за стандартом IEEE 754:

63	62	52	51	0
0	0111 1111 100	1110 0010 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000		
Знак	Порядок	Мантіса		

Відповідь = 0011 1111 1100 1110 0010 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 = 3FCE200000000000

Від'ємне число $-F = -2365.16$

Знак: 1 – від'ємне

Усе інше залишається тим самим:

79	78	64	63	0
1	1000 0000 0001 010	1001 0011 1101 0010 1000 1100 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000		
Знак	Порядок	Мантіса		

Відповідь = 1100 0000 0000 1010 1001 0011 1101 0010 1000 1100
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 =
C00A93D28C0000000000

6. $2202200_{10} = 219A58_{16}$

$22_{10} = 16_{16}$

$-22_{10} = EA_{16}$

$2202_{10} = 089A_{16}$

$-2202_{10} = F766_{16}$

$22022002_{10} = 01500772_{16}$

$-22022002_{10} = FEAFF88E_{16}$

$0.002_{10} = 3B000000_{16}$

$-0.002_{10} = BB000000_{16}$

$0.236_{10} = 3FCE200000000000_{16}$

$-0.236_{10} = BFCE200000000000_{16}$

$2365.16_{10} = 400A93D28C0000000000_{16}$

$-2365.16_{10} = C00A93D28C0000000000_{16}$

7.

; Processors

.386

.model flat, stdcall

option CaseMap:None

; Libraries And Macroses

include /masm32/include/masm32rt.inc

.data?

BufferForText DB 256 DUP(?)

BufferDPlus DB 32 DUP(?)

BufferDMinus DB 32 DUP(?)

```
BufferEPlus DB 32 DUP(?)
BufferEMinus DB 32 DUP(?)
BufferFPlus DB 32 DUP(?)
BufferFMinus DB 32 DUP(?)
```

```
; Data Segment
```

```
.data
```

```
; Name Of Message Box
```

```
MsgBoxName DB "1-9-IP93-Dominskyi", 0
```

```
; Symbols
```

```
Symbols DB "Number is 22022002", 0
```

```
; Text Of Message Box
```

```
Form DB "Symbols - %s", 10,
"A plus = %d", 10, "A minus = %d", 10,
"B plus = %d", 10, "B minus = %d", 10,
"C plus = %d", 10, "C minus = %d", 10,
"D plus = %s", 10, "D minus = %s", 10,
"E plus = %s", 10, "E minus = %s", 10,
"F plus = %s", 10, "F minus = %s", 0
```

```
; A Byte Numbers
```

```
APlusByte DB +22
```

```
AMinusByte DB -22
```

```
; A Word Numbers
```

```
APlusWord DW +22
```

```
AMinusWord DW -22
```

```
; A ShortInt Numbers
```

```
APlusShortInt DD +22
```

```
AMinusShortInt DD -22
```

```
; A LongInt Numbers
```

```
APlusLongInt DQ +22
```

```
AMinusLongInt DQ -22
```

```
; B Word Numbers
```

```
BPlusWord DW +2202
```

```
BMinusWord DW -2202
```

```
; B ShortInt Numbers
```

```
BPlusShortInt DD +2202
```

```
BMinusShortInt DD -2202
```

```
; B LongInt Numbers
```

```
BPlusLongInt DQ +2202
```

```
BMinusLongInt DQ -2202
```

```
; C ShortInt Numbers
```

```
CPlusShortInt DD +22022002
```

```
CMinusShortInt DD -22022002
```

```
; C LongInt Numbers
```

```
CPlusLongInt DQ +22022002
```

```
CMinusLongInt DQ -22022002
```

```
; D Single (Float) Numbers
```

```
DPlusSingle DD +0.002
```

DMinusSingle DD -0.002

; D Double Numbers

DPlusDouble DQ +0.002

DMinusDouble DQ -0.002

; E Double Numbers

EPlusDouble DQ +0.236

EMinusDouble DQ -0.236

; F Double Numbers

FPlusDouble DQ +2365.16

FMinusDouble DQ -2365.16

; F Extended (Long Double) Numbers

FPlusExtended DT +2365.16

FMinusExtended DT -2365.16

; Code Segment

.code

; Enter point

Main:

invoke FloatToStr2, DPlusDouble, **addr** BufferDPlus

invoke FloatToStr2, DMinusDouble, **addr** BufferDMinus

invoke FloatToStr2, EPlusDouble, **addr** BufferEPlus

invoke FloatToStr2, EMinusDouble, **addr** BufferEMinus

invoke FloatToStr2, FPlusDouble, **addr** BufferFPlus

invoke FloatToStr2, FMinusDouble, **addr** BufferFMinus

invoke wsprintf, **addr** BufferForText, **addr** Form,

addr Symbols,

APlusShortInt, AMinusShortInt,

BPlusShortInt, BMinusShortInt,

CPlusShortInt, CMinusShortInt,

addr BufferDPlus, **addr** BufferDMinus,

addr BufferEPlus, **addr** BufferEMinus,

addr BufferFPlus, **addr** BufferFMinus

invoke MessageBox, 0, **offset** BufferForText, **offset** MsgBoxName, MB_OK

invoke ExitProcess, 0

; End of a program

end Main

8. Скріншот програми:

1-9-IP93-Dominskyi ✕

```
Symbols - Number is 22022002
A plus = 22
A minus = -22
B plus = 2202
B minus = -2202
C plus = 22022002
C minus = -22022002
D plus = 0.002
D minus = -0.002
E plus = 0.236
E minus = -0.236
F plus = 2365.16
F minus = -2365.16
```

OK

9.

ml /FI 1-9-IP93-Dominskyi.asm

10.

```
00000000      .data?
00000000 00000100 [   BufferForText DB 256 DUP(?)
00
]
00000100 00000020 [   BufferDPlus DB 32 DUP(?)
00
]
00000120 00000020 [   BufferDMinus DB 32 DUP(?)
00
]
00000140 00000020 [   BufferEPlus DB 32 DUP(?)
00
]
00000160 00000020 [   BufferEMinus DB 32 DUP(?)
00
]
00000180 00000020 [   BufferFPlus DB 32 DUP(?)
00
]
000001A0 00000020 [   BufferFMinus DB 32 DUP(?)
00
]

; Data Segment
00000000      .data
; Name Of Message Box
00000000 31 2D 39 2D 49   MsgBoxName DB "1-9-IP93-Dominskyi", 0
50 39 33 2D
44 6F 6D 69
6E 73 6B 79
69 00

; Symbols
```

00000013 4E 75 6D 62 65 Symbols DB "Number is 22022002", 0
72 20 69 73
20 32 32 30
32 32 30 30
32 00

; Text Of Message Box

00000026 53 79 6D 62 6F Form DB "Symbols - %s", 10,
6C 73 20 2D
20 25 73 0A
41 20 70 6C
75 73 20 3D
20 25 64 0A
41 20 6D 69
6E 75 73 20
3D 20 25 64
0A 42 20 70
6C 75 73 20
3D 20 25 64
0A 42 20 6D
69 6E 75 73
20 3D 20 25
64 0A 43 20
70 6C 75 73
20 3D 20 25
64 0A 43 20
6D 69 6E 75
73 20 3D 20
25 64 0A 44
20 70 6C 75
73 20 3D 20
25 73 0A 44
20 6D 69 6E
75 73 20 3D
20 25 73 0A
45 20 70 6C
75 73 20 3D
20 25 73 0A
45 20 6D 69
6E 75 73 20
3D 20 25 73
0A 46 20 70
6C 75 73 20
3D 20 25 73
0A 46 20 6D
69 6E 75 73
20 3D 20 25
73 00

"A plus = %d", 10, "A minus = %d", 10,
"B plus = %d", 10, "B minus = %d", 10,
"C plus = %d", 10, "C minus = %d", 10,
"D plus = %s", 10, "D minus = %s", 10,
"E plus = %s", 10, "E minus = %s", 10,
"F plus = %s", 10, "F minus = %s", 0

; A Byte Numbers

000000C9 16 APlusByte DB +22
000000CA EA AMinusByte DB -22

; A Word Numbers

000000CB 0016 APlusWord DW +22
000000CD FFEA AMinusWord DW -22

; A ShortInt Numbers

000000CF 00000016 APlusShortInt DD +22
000000D3 FFFFFFFEA AMinusShortInt DD -22

; A LongInt Numbers

000000D7 APlusLongInt DQ +22
000000000000000016
000000DF AMinusLongInt DQ -22
FFFFFFFFFFFFFFFFFEA

; B Word Numbers

000000E7 089A BPlusWord DW +2202
000000E9 F766 BMinusWord DW -2202

; B ShortInt Numbers

000000EB 0000089A BPlusShortInt DD +2202
000000EF FFFFF766 BMinusShortInt DD -2202

; B LongInt Numbers

000000F3 BPlusLongInt DQ +2202
00000000000000089A
000000FB BMinusLongInt DQ -2202
FFFFFFFFFFFFFFFF766

; C ShortInt Numbers

00000103 01500772 CPlusShortInt DD +22022002
00000107 FEAFF88E CMinusShortInt DD -22022002

; C LongInt Numbers

0000010B CPlusLongInt DQ +22022002
000000000001500772
00000113 CMinusLongInt DQ -22022002
FFFFFFFFFEAFF88E

; D Single (Float) Numbers

0000011B 3B03126F DPlusSingle DD +0.002
0000011F BB03126F DMinusSingle DD -0.002

; D Double Numbers

00000123 DPlusDouble DQ +0.002
3F60624DD2F1A9FC
0000012B DMinusDouble DQ -0.002
BF60624DD2F1A9FC

; E Double Numbers

00000133 EPlusDouble DQ +0.236
3FCE353F7CED9168
0000013B EMinusDouble DQ -0.236
BFCE353F7CED9168

; F Double Numbers

00000143 FPlusDouble DQ +2365.16
40A27A51EB851EB8
0000014B FMinusDouble DQ -2365.16
C0A27A51EB851EB8

; F Extended (Long Double) Numbers


```

00000153      FPlusExtended DT +2365.16
400A93D28F5C28F5C28F
0000015D      FMinusExtended DT -2365.16
C00A93D28F5C28F5C28F

; Code Segment
00000000      .code
; Enter point
00000000      Main:
    invoke FloatToStr2, DPlusDouble, addr BufferDPlus
    invoke FloatToStr2, DMinusDouble, addr BufferDMinus
    invoke FloatToStr2, EPlusDouble, addr BufferEPlus
    invoke FloatToStr2, EMinusDouble, addr BufferEMinus
    invoke FloatToStr2, FPlusDouble, addr BufferFPlus
    invoke FloatToStr2, FMinusDouble, addr BufferFMinus

    invoke wsprintf, addr BufferForText, addr Form,
    addr Symbols,
    APlusShortInt, AMinusShortInt,
    BPlusShortInt, BMinusShortInt,
    CPlusShortInt, CMinusShortInt,
    addr BufferDPlus, addr BufferDMinus,
    addr BufferEPlus, addr BufferEMinus,
    addr BufferFPlus, addr BufferFMinus

    invoke MessageBox, 0, offset BufferForText, offset MsgBoxName, MB_OK
    invoke ExitProcess, 0
; End of a program
end Main

```

12.

Число	Варіант у лістингу	Варіант у роботі
+A	16	16
-A	EA	EA
+B	089A	089A
-B	F766	F766
+C	01500772	01500772
-C	FEAFF88E	FEAFF88E
+D	3B03126F	3B031200
-D	7B03126F	7B031200
+E	3FCE353F7CED9168	3FCE200000000000
-E	BFCE353F7CED9168	BFCE200000000000
+F	400A93D28F5C28F5C28F	400A93D28C0000000000
-F	C00A93D28F5C28F5C28F	C00A93D28C0000000000

Висновок:

Завдяки цій лабораторній роботі Я ознайомився з Assembler, зі змінними, сегментами коду, функціями, форматуваннями, лістингом. Також перевірів цифри, які сам перевів у HEX формат з тими, які зробив Assembler. Зверху

видно, що деякі дані з лістингу не співпадають (зелений фон). Це можна пояснити тим, що Ми округлювали числа до 10 знаків після коми.