**КУРСОВА РОБОТА**

з дисципліни «Системне програмування»

на тему «Розробка компілятора програм мовою Асемблера»

Студента \_\_\_\_ курсу \_\_\_\_\_\_\_\_\_ групи

напряму підготовки

6.050102 «Комп’ютерна інженерія»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(прізвище та ініціали)

Керівник \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Національна оцінка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_\_\_ Оцінка: ECTS \_\_\_\_\_

Члени комісії \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (вчене звання, науковий ступінь, прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (вчене звання, науковий ступінь, прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (вчене звання, науковий ступінь, прізвище та ініціали)

ТЕХНІЧНЕ ЗАВДАННЯ

1. Вхідні дані транслятора - текстовий файл з довільною програмою на мові Асемблера, складеною в відповідності з обмеженнями, які задані в варіанті курсової роботи. Для підготовки програми на мові Асемблера використовується, наприклад, стандартний додаток OS Windows Блокнот.
2. На всі синтаксичні конструкції (ідентифікатори, константи, директиви, машинні команди, режими адресації і т.д.), які допускаються в TASM(MASM) і які виходять за рамки обмежень в варіанті курсової роботи повинно видаватись діагностичне повідомлення про синтаксичну помилку.
3. В результаті роботи транслятора повинен бути створений текстовий файл лістінга (розширення .lst). Формат файлу лістінга повинен співпадати з форматом файлу лістінга MASM або TASM. Діагностичні повідомлення формуються на українській мові. Таблиця символів в файлі лістінга може бути в довільному форматі.
4. Транслятор повинен аналізувати командний рядок, в якому задаються імена початкового файлу та файлу лістінга. Всі діагностичні повідомлення, які формуються в файлі лістінга додатково повинні виводитись на екран монітора. Крім того, на екран виводиться загальна кількість помилок, виявлених в початковій програмі.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

***Ідентифікатори***

Містять великі і малі букви латинського алфавіту та цифри. Починаються з букви. Великі та малі букви не відрізняються. Довжина ідентифікаторів не більше 8 символів

***Константи***

Шістнадцяткові, десяткові та двійкові константи

***Директиви***

END,

SEGMENT - без операндів, ENDS, програма може мати тільки один сегмент кодів і тільки один сегмент даних

DB,DW,DD з одним операндом - довільний арифметичний вираз над константами

***Розрядність даних та адрес***

32 - розрядні дані та зміщення в сегменті, 16 - розрядні дані та зміщення не використовуються

***Адресація операндів пам'яті***

Індексна адресація (Val1[eax],Val1[edi] і т.п.)

***Заміна сегментів***

Префікси заміни сегментів можуть задаватись явно, а при необхідності автоматично генеруються транслятором

***Машинні команди***

Movsb

Rep Movs mem

Add **reg,reg**

Or **reg,mem**

Test **mem,reg**

Mov **reg,imm**

Shl **mem,imm**

Jnz,

де **reg –** 8 або 32-розрядні РЗП;

**mem –** адреса операнда в пам’яті;

**imm –** 8 або 32-розрядні безпосередні дані (довільний арифметичний вираз над константами).

ТЕСТОВІ ФАЙЛИ

**Без помилок для створюваного компілятора**

DATA SEGMENT

ID1 DB 255

ID2 DW 1111B

iD3 DD 5BC35FF6H

AR1 DD 10

AR2 DD 10

DATA ENDS

CODE SEGMENT

BEGIN:

; AND test

MOV AL, 0FH

MOV BL, 11H + 00001111B \* 13

ADD AL, BL

ADD EBX, ECX

; OR test

MOV CL, 00H

OR CL, ID1

JNZ LABEL\_F

; TEST test

TEST ID1, BL

TEST ID3, EAX

; MOV test

MOV EAX, 0FFFFFFFH

MOV BL, 0AH

; INDEX test

TEST iD3[ESI], 147

TEST iD3[DI], 5H

LABEL\_F:

; SHL test

SHL ID3, 00000001B

SHL ID1, 11H

; JNZ TEST

JNZ BEGIN

; SEGMENT change

TMP1 DB 13

TEST TMP1, AL

TMP2 DW 0334H

SHL TMP2, 3

TMP3 DD 43444546H

TEST TMP3, EAX

MOV ECX, 10

MOV SI, OFFSET AR1

MOV DI, OFFSET AR2

REP MOVSB

CODE ENDS

END BEGIN

**Без помилок для MASM**

.386

DATA SEGMENT

ID1 DB 255

ID2 DW 1111B

iD3 DD 5BC35FF6H

AR1 DD 10

AR2 DD 10

DATA ENDS

CODE SEGMENT

ASSUME DS:DATA, CS:CODE

BEGIN:

; AND test

MOV AL, 0FH

MOV BL, 11H + 00001111B \* 13

ADD AL, BL

ADD EBX, ECX

; OR test

MOV CL, 00H

OR CL, ID1

JNZ LABEL\_F

; TEST test

TEST ID1, BL

TEST ID3, EAX

; MOV test

MOV EAX, 0FFFFFFFH

MOV BL, 0AH

; INDEX test

TEST iD3[ESI], 147

TEST iD3[DI], 5H

LABEL\_F:

; SHL test

SHL ID3, 00000001B

SHL ID1, 11H

; JNZ TEST

JNZ BEGIN

; SEGMENT change

TMP1 DB 13

TEST TMP1, AL

TMP2 DW 0334H

SHL TMP2, 3

TMP3 DD 43444546H

TEST TMP3, EAX

MOV ECX, 10

MOV SI, OFFSET AR1

MOV DI, OFFSET AR2

REP MOVSB

CODE ENDS

END BEGIN

**Його лістинг**

Microsoft (R) Macro Assembler Version 5.00 2/20/14 12:50:34

Page 1-1

.386

0000 DATA SEGMENT

0000 FF ID1 DB 255

0001 000F ID2 DW 1111B

0003 F65FC35B iD3 DD 5BC35FF

6H

0007 0A000000 AR1 DD 10

000B 0A000000 AR2 DD 10

000F DATA ENDS

0000 CODE SEGMENT

ASSUME DS:DATA, CS:CODE

0000 BEGIN:

; AND test

0000 B0 0F MOV AL, 0FH

0002 B3 D4 MOV BL, 11H + 00001

111B \* 13

0004 02 C3 ADD AL, BL

0006 03 D9 ADD EBX, ECX

; OR test

0008 B1 00 MOV CL, 00H

000A 0A 0D 00000000 R OR CL, ID1

0010 0F 85 0000003C R JNZ LABEL\_F

; TEST test

0016 84 1D 00000000 R TEST ID1, BL

001C 85 05 00000003 R TEST ID3, EAX

; MOV test

0022 B8 0FFFFFFF MOV EAX, 0FFFFFFFH

0027 B3 0A MOV BL, 0AH

; INDEX test

0029 F7 86 00000003 R 0000 TEST iD3[ESI], 147

0093

0033 67| F7 85 0003 R 0000 TEST iD3[DI], 5H

0005

003C LABEL\_F:

; SHL test

003C D1 25 00000003 R SHL ID3, 00000001B

0042 C0 25 00000000 R 11 SHL ID1, 11H

; JNZ TEST

0049 75 B5 JNZ BEGIN

; SEGMENT change

004B 0D TMP1 DB 13

004C 2E: 84 05 0000004B R TEST TMP1, AL

0053 0334 TMP2 DW 0334H

Microsoft (R) Macro Assembler Version 5.00 2/20/14 12:50:34

Page 1-2

0055 2E: 66| C1 25 00000053 SHL TMP2, 3

R 03

005E 46454443 TMP3 DD 43444546H

0062 2E: 85 05 0000005E R TEST TMP3, EAX

0069 B9 0000000A MOV ECX, 10

006E 66| BE 0007 R MOV SI, OFFSET AR1

0072 66| BF 000B R MOV DI, OFFSET AR2

0076 F3/ A4 REP MOVSB

0078 CODE ENDS

END BEGIN

Microsoft (R) Macro Assembler Version 5.00 2/20/14 12:50:34

Symbols-1

N a m e Size Length Align Combine Class

CODE . . . . . . . . . . . . . . 32 Bit 0078 PARA NONE

DATA . . . . . . . . . . . . . . 32 Bit 000F PARA NONE

Symbols:

N a m e Type Value Attr

AR1 . . . . . . . . . . . . . . L DWORD 0007 DATA

AR2 . . . . . . . . . . . . . . L DWORD 000B DATA

BEGIN . . . . . . . . . . . . . L NEAR 0000 CODE

ID1 . . . . . . . . . . . . . . L BYTE 0000 DATA

ID2 . . . . . . . . . . . . . . L WORD 0001 DATA

ID3 . . . . . . . . . . . . . . L DWORD 0003 DATA

LABEL\_F . . . . . . . . . . . . L NEAR 003C CODE

TMP1 . . . . . . . . . . . . . . L BYTE 004B CODE

TMP2 . . . . . . . . . . . . . . L WORD 0053 CODE

TMP3 . . . . . . . . . . . . . . L DWORD 005E CODE

@FILENAME . . . . . . . . . . . TEXT masmc

62 Source Lines

62 Total Lines

14 Symbols

51060 + 465484 Bytes symbol space free

0 Warning Errors

0 Severe Errors

**З типовими помилками для створюваного компілятора**

DATA SEGMENT

ID1 DB 255

ID2 DW 1111B

iD3 DD 5BC35FF6H

/IMWRONG DB 12

METOO DC F24H

AR1 DD 10

AR2 DD 10

DATA ENDS

CODE SEGMENT

BEGIN:

LABEL\_BACK:

; AND test

MOV AX, 0FH ; Error

MOV BL, 11H + 00001111B \* 13

ADD ID2, BL

ADD EBX, ECX

; OR test

MOV CL, 00H

OR CX, ID1

JNZ LABEL\_FRONT ; Error

; TEST test

TEST ID1, DS ; Error

TEST ID3, EAX

; MOV test

MOV EAX, 0FFFFFFFH

MOV BL, 0AH

; INDEX test

TEST iD3[EAX], 147 ; Error

TEST iD3[DI], 5H

LABEL\_FRONT:

; SHL test

SHL ID3, 00000001B

SHL ID1, 11111111H ; Error

; JNZ TEST

JNZ IDONTEXIST ; Error

; SEGMENT change

TMP1 DB AL ; Error

TEST TMP1, AL

TMP2 DW 0334H

SHL TMP2, 3

TMP3 DD 43444546H

TEST TMP3, EAX

MOV ECX, 10

LEA SI, AR1 ; Error

LEA DI, AR2 ; Error

REP MOVSB

CODE ENDS

END BEGIN

**З типовими помилками для MASM**

.386

DATA SEGMENT

ID1 DB 255

ID2 DW 1111B

iD3 DD 5BC35FF6H

/IMWRONG DB 12

METOO DC F24H

AR1 DD 10

AR2 DD 10

DATA ENDS

CODE SEGMENT

ASSUME DS:DATA, CS:CODE

BEGIN:

LABEL\_BACK:

; AND test

MOV AX, 0FH ; Error

MOV BL, 11H + 00001111B \* 13

ADD ID2, BL

ADD EBX, ECX

; OR test

MOV CL, 00H

OR CL, ID1

JNZ LABEL\_FRONT ; Error

; TEST test

TEST ID1, DS ; Error

TEST ID3, EAX

; MOV test

MOV EAX, 0FFFFFFFH

MOV BL, 0AH

; INDEX test

TEST iD3[EAX], 147 ; Error

TEST iD3[DI], 5H

LABEL\_FRONT:

; SHL test

SHL ID3, 00000001B

SHL ID1, 11111111H

; JNZ TEST

JNZ LABEL\_BACK ; Error

; SEGMENT change

TMP1 DB AL ; Error

TEST TMP1, AL

TMP2 DW 0334H

SHL TMP2, 3

TMP3 DD 43444546H

TEST TMP3, EAX

MOV ECX, 10

LEA SI, AR1

LEA DI, AR2

REP MOVSB

CODE ENDS

END BEGIN

**Його лістинг**

Microsoft (R) Macro Assembler Version 5.00 2/20/14 12:53:19

Page 1-1

.386

0000 DATA SEGMENT

0000 FF ID1 DB 255

0001 000F ID2 DW 1111B

0003 F65FC35B iD3 DD 5BC35FF

6H

/IMWRONG DB 12

masme.ASM(7): warning A4001: Extra characters on line

METOO DC F24H

masme.ASM(8): error A2105: Expected: instruction or directive

0007 0A000000 AR1 DD 10

000B 0A000000 AR2 DD 10

000F DATA ENDS

0000 CODE SEGMENT

ASSUME DS:DATA, CS:CODE

0000 BEGIN:

0000 LABEL\_BACK:

; AND test

0000 66| B8 000F MOV AX, 0FH

; Error

0004 B3 D4 MOV BL, 11H + 00001

111B \* 13

0006 00 1D 00000001 R ADD ID2, BL

masme.ASM(23): warning A4031: Operand types must match

000C 03 D9 ADD EBX, ECX

; OR test

000E B1 00 MOV CL, 00H

0010 0A 0D 00000000 R OR CL, ID1

0016 0F 85 00000042 R JNZ LABEL\_FRONT

; Error

; TEST test

001C 85 1D 00000000 R TEST ID1, DS

; Error

masme.ASM(32): error A2061: Improper use of segment register

0022 85 05 00000003 R TEST ID3, EAX

; MOV test

0028 B8 0FFFFFFF MOV EAX, 0FFFFFFFH

002D B3 0A MOV BL, 0AH

; INDEX test

002F F7 80 00000003 R 0000 TEST iD3[EAX], 147 ; Error

0093

0039 67| F7 85 0003 R 0000 TEST iD3[DI], 5H

0005

Microsoft (R) Macro Assembler Version 5.00 2/20/14 12:53:19

Page 1-2

0042 LABEL\_FRONT:

; SHL test

0042 D1 25 00000003 R SHL ID3, 00000001B

0048 C0 25 00000000 R 11111111 SHL ID1, 11111111H

masme.ASM(46): error A2050: Value out of range

; JNZ TEST

0052 75 AC JNZ LABEL\_BACK

; Error

; SEGMENT change

0054 00 TMP1 DB AL

; Error

masme.ASM(52): error A2052: Improper operand type

0055 2E: 84 05 00000054 R TEST TMP1, AL

005C 0334 TMP2 DW 0334H

005E 2E: 66| C1 25 0000005C SHL TMP2, 3

R 03

0067 46454443 TMP3 DD 43444546H

006B 2E: 85 05 00000067 R TEST TMP3, EAX

0072 B9 0000000A MOV ECX, 10

0077 8D 35 00000007 R LEA SI, AR1

007D 8D 3D 0000000B R LEA DI, AR2

0083 F3/ A4 REP MOVSB

0085 CODE ENDS

END BEGIN

Microsoft (R) Macro Assembler Version 5.00 2/20/14 12:53:19

Symbols-1

N a m e Size Length Align Combine Class

CODE . . . . . . . . . . . . . . 32 Bit 0085 PARA NONE

DATA . . . . . . . . . . . . . . 32 Bit 000F PARA NONE

Symbols:

N a m e Type Value Attr

AR1 . . . . . . . . . . . . . . L DWORD 0007 DATA

AR2 . . . . . . . . . . . . . . L DWORD 000B DATA

BEGIN . . . . . . . . . . . . . L NEAR 0000 CODE

ID1 . . . . . . . . . . . . . . L BYTE 0000 DATA

ID2 . . . . . . . . . . . . . . L WORD 0001 DATA

ID3 . . . . . . . . . . . . . . L DWORD 0003 DATA

LABEL\_BACK . . . . . . . . . . . L NEAR 0000 CODE

LABEL\_FRONT . . . . . . . . . . L NEAR 0042 CODE

TMP1 . . . . . . . . . . . . . . L BYTE 0054 CODE

TMP2 . . . . . . . . . . . . . . L WORD 005C CODE

TMP3 . . . . . . . . . . . . . . L DWORD 0067 CODE

@FILENAME . . . . . . . . . . . TEXT masme

66 Source Lines

66 Total Lines

15 Symbols

51060 + 465484 Bytes symbol space free

2 Warning Errors

4 Severe Errors