НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра системного програмування і спеціалізованих комп'ютерних

систем

**КУРСОВА РОБОТА**

з дисципліни «Системне програмування»

на тему: Розробка компілятора програм мовою Асемблера

Студента 2 курсу групи КВ-51

напряму підготовки

6.050101 «Ком`ютерна інженерія»

Марченко О.Б.

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)

Київ – 2017 рік

**Технічне завдання**

**Тема курсової роботи** – Розробка компілятора програм мовою Асемблера. Основним завданням і результатом роботи програми компілятора повинно бути створення для програм мовою Асемблера текстового файла лістинга (розширення .lst), подібного до файла лістинга компілятора MASM або TASM. Враховуючи, що створення компілятора є трудомістким процесом, який потребує значних витрат часу, у варіантах завдань на курсову роботу використовуються суттєві обмеження на перелік допустимих машинних інструкцій, режимів адресації даних і команд та допустимих директив. Будь який із запропонованих студентам варіантів індивідуальних завдань фактично вказує на підмножину стандартної мови Асемблера процесорів фірми Intel. Тим не менш, цієї підмножини достатньо для забезпечення мети розробки.

**Варіант 11 (КВ-51):**

*Ідентифікатори*

Містять малі букви латинского алфавіту та цифри. Починаються з букви. Великі та малі букви не відрізняються. Довжина ідентифікаторів не більше 4 символів

*Константи*

Шістнадцяткові, десяткові, двійкові та текстові константи

*Директиви*

END,

SEGMENT - без операндів, ENDS, ASSUME

MACRO (без параметрів або з одним параметром) ENDM

DB,DW,DD з одним операндом - константою (рядкові константи тільки для DB)

*Розрядність даних та адрес*

32 - розрядні дані та зміщення в сегменті, y випадку 16-розрядних даних та 16 - розрядних зміщень генеруються відповідні префікси зміни розрядності

*Адресація операндів пам'яті*

Базова індексна адресація з константним зміщенням ([edx+esi+6],[ebx+ecx+12], ([bx+si+6],[ bp+di+1] і т.п.) з оператором визначення типу (ptr) при необхідності

*Заміна сегментів*

Префікси заміни сегментів можуть задаватись тільки явно

*Машинні команди*

Neg **reg**

Dec **mem**

Adc **reg, mem**

Cmp **mem, reg**

Or **reg,reg**

Mov **reg,imm**

And **mem,imm**

Jc

Jmp (міжсегментна пряма адресація)

Де **reg –** 8 або 32-розрядні РЗП

**mem –** адреса операнда в пам’яті

**imm –** 8,16 або 32-розрядні безпосередні дані (константи)

**Опис загальної структури програми**

Програма складається з двох частин. Перша частина – це лексичний аналізатор, а друга часина – одночасна робота синтаксичного та семантичного аналізаторів. Синтаксичний аналізатор -- перевіряє кожну конструкцію на відповідність її до стандартів мови. Кожна конструкція мови розділяється на блоки , які і являють собою ці рекурсивні функції.

**ОПИС ДІЙ ТРАНСЛЯТОРА**

Зчитується файл у массив rows ,де перший iндекс -це номер рядка ,а другий позицiя слова у рядку.

За допомогою створенного словника лексем у виглядi бази данних mySQL та функцiй виявлення типiв лексем за кодом для кожного елементу массиву додаються властивостi ,а саме iм’я лексеми ,тип ,довжина ,та необхiднi лексеми пiд час синтасксичного та сематичного аналiзу .

На етапi синтасксичного аналiзу для всiх лексем послiдовно перевiряються на наявнiсть на необхiдному мiсцi записанi на попередньому етапi лексеми.

Далi перевiрка на правильнiсть розташування вiдповiдних рядкiв вiдносно сегментiв.

На останьому етапi виводиться пiдсумкова таблиця ,а також генеруються рядки помилок .

Після завершення аналізу ,у лістинг записується:

Для кожного рядка:

Вихідний рядок

“ERROR” при наявності помилки

**ОПИС РОЗРОБЛЕНИХ ФУНКЦІЙ ТРАНСЛЯТОРА**

**Processing.php**

У тілі програми запускається обробка отриманного з POST-форми файлу. Після чого запускається функція

start(‘\*.asm’)

Запускає обробку файлу та створює інтерфейс програми та записує кожне окреме слово файлу у багатовимірний масив.

Lexical\_analisation(array\_rows)

Запускає лексичний аналіз массиву записаного на попередньому етапі за допомогою функції:

Detect\_lexem(lexem)

Запускає перевірку на тип лексеми кожного слова ,проходячи по функціям перевірки усіма можливими типам і знаходячи спільні властивості.

syntax\_analisation(count\_of\_rows,array\_rows)

Запускає синтаксичний та граматичний аналіз масиву лексем . Спершу перевіряє наявність потрібних лексем на своїх позиція у рядку та записує масив лексем, які необхідно знайти в іншій частині коду. На наступному етапі перевіряє границі сегментів та відсутність в них сторонніх операндів.

table(count\_of\_rows,array\_rows)

Виводить результат трансляції у программу для зручного користування ним.

Check\_errors(count\_of\_rows,array\_rows)

Підраховує помилки та записує у \*.lst результат роботи транслятора.

**Create\_Table.php**

Створює базу данних лексем.

**Upload.php**

Необхідний файл для завантаження файлу користувачем через інтерфейс програми.

**db.php**

Створює зв’язок бази даних з бібліотекою redBean, за допомогою якої відбувається користування словником лексем.

**Css/**

Містить стилі зовнішнього вигляду програми.

**js/**

Містить функції обробки дій користувача та виводу їх результатів.

**libs/**

Містить підключену бібліотеку.

**Тестове завдання без помилок:**

.386

mcrs MACRO

And eax,ebx

ENDM

mcr MACRO parm

And eax,ebx

ENDM

data segment

stri db "big ukrainian boss's string"

hexa dw 3Eh

bina dd 01101101b

deca dd 21

data ends

cod1 segment

assume cs:cod1,ds:data

met2:

Adc ax,[ebx+eax+6]

Neg bh

Neg ebx

Dec ax

Adc ecx,[ebx+si+6]

Adc bx,bina

Adc ecx,stri

Cmp bx,ah

Cmp [ebx],eax

Cmp deca,ah

Cmp bina,eax

met1:

Or [eax+0110b],ebx

Or bh,bl

Or al,dh

Or cx,dx

Mov [bp+si+5],bl

Mov [eax+3Ah] ,bx

And [bx+di+5],8AF2h

And [ebx+6],8AF22E4Bh

Jc met1

And ecx,ebx

Jc near ptr met1

Jmp far ptr met3

cod1 ends

cod2 segment

assume cs:cod2,ds:data

met3:

Mov [bp+di+5],bl

Jmp far ptr met2

cod2 ends

END

**Файл лістингу MASM:**

Microsoft (R) Macro Assembler Version 5.00 6/26/17 12:19:47

Page 1-1

.386

mcrs MACRO

And eax,ebx

ENDM

mcr MACRO parm

And eax,ebx

ENDM

0000 data segment

0000 62 69 67 20 75 6B 72 stri db "big ukrainian boss's string"

61 69 6E 69 61 6E 20

62 6F 73 73 27 73 20

73 74 72 69 6E 67

001B 003E hexa dw 3Eh

001D 6D000000 bina dd 01101101b

0021 15000000 deca dd 21

0025 data ends

0000 cod1 segment

assume cs:cod1,ds:data

0000 met2:

0000 66| 13 4403 06 Adc ax,[ebx+eax+6]

0005 F6 DF Neg bh

0007 F7 DB Neg ebx

0009 66| 48 Dec ax

000B 13 4B 06 Adc ecx,[ebx+si+6]

000E 13 1D 0000001D R Adc bx,bina

final.ASM(27): warning A4031: Operand types must match

0014 13 0D 00000000 R Adc ecx,stri

final.ASM(28): warning A4031: Operand types must match

001A 3B DC Cmp bx,ah

final.ASM(30): warning A4031: Operand types must match

001C 39 03 Cmp [ebx],eax

001E 38 25 00000021 R Cmp deca,ah

final.ASM(33): warning A4031: Operand types must match

0024 39 05 0000001D R Cmp bina,eax

002A met1:

002A 09 58 06 Or [eax+0110b],ebx

002D 0A FB Or bh,bl

002F 0A C6 Or al,dh

0031 66| 0B CA Or cx,dx

0034 67| 88 5A 05 Mov [bp+si+5],bl

0038 66| 89 58 3A Mov [eax+3Ah] ,bx

003C 67| 81 61 05 00008AF2 And [bx+di+5],8AF2h

0044 81 63 06 8AF22E4B And [ebx+6],8AF22E4Bh

Microsoft (R) Macro Assembler Version 5.00 6/26/17 12:19:47

Page 1-2

004B 72 DD Jc met1

004D 23 CB And ecx,ebx

004F 72 D9 Jc near ptr met1

0051 EA 00000000 ---- R Jmp far ptr met3

0058 cod1 ends

0000 cod2 segment

assume cs:cod2,ds:data

0000 met3:

0000 67| 88 5B 05 Mov [bp+di+5],bl

0004 EA 00000000 ---- R Jmp far ptr met2

000B cod2 ends

ENDMicrosoft (R) Macro Assembler Version 5.00 6/26/17 12:19:47

Symbols-1

Macros:

N a m e Lines

MCR . . . . . . . . . . . . . . 1

MCRS . . . . . . . . . . . . . . 1

N a m e Size Length Align Combine Class

COD1 . . . . . . . . . . . . . . 32 Bit 0058 PARA NONE

COD2 . . . . . . . . . . . . . . 32 Bit 000B PARA NONE

DATA . . . . . . . . . . . . . . 32 Bit 0025 PARA NONE

Symbols:

N a m e Type Value Attr

BINA . . . . . . . . . . . . . . L DWORD 001D DATA

DECA . . . . . . . . . . . . . . L DWORD 0021 DATA

HEXA . . . . . . . . . . . . . . L WORD 001B DATA

MET1 . . . . . . . . . . . . . . L NEAR 002A COD1

MET2 . . . . . . . . . . . . . . L NEAR 0000 COD1

MET3 . . . . . . . . . . . . . . L NEAR 0000 COD2

STRI . . . . . . . . . . . . . . L BYTE 0000 DATA

@FILENAME . . . . . . . . . . . TEXT final

62 Source Lines

62 Total Lines

14 Symbols

51100 + 465444 Bytes symbol space free

4 Warning Errors

0 Severe Errors

**Файл лістингу створеної програми:**

Marchecko KV-51 v11 listing

0 : .386

Error near line 0:unexpected lexem ".386

" on line 0 on position 0

1 : mcrs MACRO

2 : And eax , ebx

3 : ENDM

4 : mcr MACRO parm

5 : And eax , ebx

6 : ENDM

7 : data segment

8 : stri db "big ukrainian boss's string"

9 : hexa dw 3Eh

10 : bina dd 01101101b

11 : deca dd 21

12 : data ends

13 : cod1 segment

14 : assume cs:cod1 , ds:data

15 : met2:

16 : Adc ax , [ebx+eax+6]

17 : Neg bh

18 : Neg ebx

19 : Dec ax

20 : Adc ecx , [ebx+si+6]

21 : Adc bx , bina

22 : Adc ecx , stri

23 : Cmp bx , ah

Error near line 23:Operand types must match on line 23

24 : Cmp [ebx] , eax

25 : Cmp deca , ah

26 : Cmp bina , eax

27 : met1:

28 : Or [eax+0110b] , ebx

29 : Or bh , bl

30 : Or al , dh

31 : Or cx , dx

32 : Mov [bp+si+5] , bl

33 : Mov [eax+3Ah] , bx

34 : And [bx+di+5] , 8AF2h

35 : And [ebx+6] , 8AF22E4Bh

36 : Jc met1

37 : And ecx , ebx

38 : Jc near ptr met1

39 : Jmp far ptr met3

40 : cod1 ends

41 : cod2 segment

42 : assume cs:cod2 , ds:data

43 : met3:

44 : Mov [bp+di+5] , bl

45 : Jmp far ptr met2

46 : cod2 ends

47 : END

Find 2 errors

**Тестове завдання з помилками:**

.386

mcrs MACRO

And parm,dd

ENDM

data segment

str db "big ukrainian boss's string"

hex dw 3Eh

bina dd 01101101b

deca dd 21

data ends

cod1 segment

assume cs:cod1,ds:data

Neg bh

Neg ebx

Dec deca

Adc bh, ptr [ebx+6]

Adc ecx,[ebx+6]

Adc bh,bina

Adc ecx,hexa

Cmp es:[ebx+6],ah

Cmp [ebx+6],eax

Cmp deca,ah

Cmp bina,eax

Or eax,bh

Or bh,bl

Or al,edx

Or ecx,edx

Mov eax,bl

Mov eax ,bx

Mov eax,ebx

Mov al,bl

Mov al,bx

Mov al,ebx

met1:

;And [bx+si],8Ah

And [bx+bp+di+5],8AF2h

And [ebx+6],8AF22E4Bh

And hexa,bh

And bina,bx

And deca,ebx

Jc met1

Jmp met2

cod1 ends

cod2 segment

assume cs:cod2,ds:data

met2:

cod2 ends

END

**Файл лістингу MASM:**

Microsoft (R) Macro Assembler Version 5.00 6/26/17 12:22:10

Page 1-1

.386

mcrs MACRO

And parm,dd

ENDM

0000 data segment

0000 62 69 67 20 75 6B 72 str db "big ukrainian boss's string"

false.ASM(7): warning A4016: Reserved word used as symbol: STR

61 69 6E 69 61 6E 20

62 6F 73 73 27 73 20

73 74 72 69 6E 67

001B 003E hex dw 3Eh

001D 6D000000 bina dd 01101101b

0021 15000000 deca dd 21

0025 data ends

0000 cod1 segment

assume cs:cod1,ds:data

0000 F6 DF Neg bh

0002 F7 DB Neg ebx

0004 FF 0D 00000021 R Dec deca

000A 12 7B 06 Adc bh, ptr [ebx+6]

false.ASM(19): error A2027: Operand expected

000D 13 4B 06 Adc ecx,[ebx+6]

0010 12 3D 0000001D R Adc bh,bina

false.ASM(22): warning A4031: Operand types must match

0016 13 0D 00000000 U Adc ecx,hexa

false.ASM(23): error A2009: Symbol not defined: HEXA

001C 26: 38 63 06 Cmp es:[ebx+6],ah

0020 39 43 06 Cmp [ebx+6],eax

0023 38 25 00000021 R Cmp deca,ah

false.ASM(28): warning A4031: Operand types must match

0029 39 05 0000001D R Cmp bina,eax

002F 0B C7 Or eax,bh

false.ASM(31): warning A4031: Operand types must match

0031 0A FB Or bh,bl

0033 0A C2 Or al,edx

false.ASM(33): warning A4031: Operand types must match

0035 0B CA Or ecx,edx

0037 8B C3 Mov eax,bl

false.ASM(36): warning A4031: Operand types must match

0039 8B C3 Mov eax ,bx

false.ASM(37): warning A4031: Operand types must match

003B 8B C3 Mov eax,ebx

003D 8A C3 Mov al,bl

003F 8A C3 Mov al,bx

false.ASM(40): warning A4031: Operand types must match

0041 8A C3 Mov al,ebx

false.ASM(41): warning A4031: Operand types must match

Microsoft (R) Macro Assembler Version 5.00 6/26/17 12:22:10

Page 1-2

0043 met1:

;And [bx+si],8Ah

0043 67| 81 61 05 00008AF2 And [bx+bp+di+5],8AF2h

false.ASM(45): error A2046: Multiple base registers

004B 81 63 06 8AF22E4B And [ebx+6],8AF22E4Bh

0052 20 3D 00000000 U And hexa,bh

false.ASM(48): error A2009: Symbol not defined: HEXA

0058 21 1D 0000001D R And bina,bx

false.ASM(49): warning A4031: Operand types must match

005E 21 1D 00000021 R And deca,ebx

0064 72 DD Jc met1

0066 E9 00000000 R Jmp met2

false.ASM(53): error A2064: Near JMP/CALL to different CS

006B cod1 ends

0000 cod2 segment

assume cs:cod2,ds:data

0000 met2:

0000 cod2 ends

END

Microsoft (R) Macro Assembler Version 5.00 6/26/17 12:22:10

Symbols-1

Macros:

N a m e Lines

MCRS . . . . . . . . . . . . . . 1

N a m e Size Length Align Combine Class

COD1 . . . . . . . . . . . . . . 32 Bit 006B PARA NONE

COD2 . . . . . . . . . . . . . . 32 Bit 0000 PARA NONE

DATA . . . . . . . . . . . . . . 32 Bit 0025 PARA NONE

Symbols:

N a m e Type Value Attr

BINA . . . . . . . . . . . . . . L DWORD 001D DATA

DECA . . . . . . . . . . . . . . L DWORD 0021 DATA

HEX . . . . . . . . . . . . . . L WORD 001B DATA

MET1 . . . . . . . . . . . . . . L NEAR 0043 COD1

MET2 . . . . . . . . . . . . . . L NEAR 0000 COD2

STR . . . . . . . . . . . . . . L BYTE 0000 DATA

@FILENAME . . . . . . . . . . . TEXT false

61 Source Lines

61 Total Lines

12 Symbols

51062 + 465482 Bytes symbol space free

10 Warning Errors

5 Severe Errors

**Файл лістингу створеної програми:**

Marchecko KV-51 v11 listing

0 : .386

Error near line 0:unexpected lexem ".386

" on line 0 on position 0

1 : mcrs MACRO

2 : And parm , dd

Error near line 2:"register/memory" is required at the next position of lexem on line 2 on position 0

"register/memory/const/variable" is missed after lexem on line 2 on position 0

"identifier" is required at the previous position of lexem on line 2 on position 3

"ds" is required before this row by lexem on line 2 on position 3

3 : ENDM

4 : data segment

5 : str db "big ukrainian boss's string"

6 : hex dw 3Eh

7 : bina dd 01101101b

8 : deca dd 21

9 : data ends

10 : cod1 segment

11 : assume cs:cod1 , ds:data

12 : Neg bh

13 : Neg ebx

14 : Dec deca

Error near line 14:"register/memory" is required at the next position of lexem on line 14 on position 0

15 : Adc bh , ptr [ebx+6]

16 : Adc ecx , [ebx+6]

17 : Adc bh , bina

18 : Adc ecx , hexa

Error near line 18:"register/memory/const/variable" is missed after lexem on line 18 on position 0

19 : Cmp es:[ebx+6] , ah

20 : Cmp [ebx+6] , eax

21 : Cmp deca , ah

22 : Cmp bina , eax

23 : Or eax , bh

Error near line 23:Operand types must match on line 23

24 : Or bh , bl

25 : Or al , edx

Error near line 25:Operand types must match on line 25

26 : Or ecx , edx

27 : Mov eax , bl

Error near line 27:Operand types must match on line 27

28 : Mov eax , bx

Error near line 28:Operand types must match on line 28

29 : Mov eax , ebx

30 : Mov al , bl

31 : Mov al , bx

Error near line 31:Operand types must match on line 31

32 : Mov al , ebx

Error near line 32:Operand types must match on line 32

33 : met1:

34 : ;And [bx+si] , 8Ah

Error near line 34:unexpected lexem ";And" on line 34 on position 0

35 : And [bx+bp+di+5] , 8AF2h

Error near line 35:"register/memory" is required at the next position of lexem on line 35 on position 0

unexpected lexem "[bx+bp+di+5]" on line 35 on position 1

36 : And [ebx+6] , 8AF22E4Bh

37 : And hexa , bh

Error near line 37:"register/memory" is required at the next position of lexem on line 37 on position 0

38 : And bina , bx

Error near line 38:"register/memory" is required at the next position of lexem on line 38 on position 0

39 : And deca , ebx

Error near line 39:"register/memory" is required at the next position of lexem on line 39 on position 0

40 : Jc met1

41 : Jmp met2

Error near line 41:"met name" is required at the next position of lexem on line 41 on position 0

42 : cod1 ends

43 : cod2 segment

44 : assume cs:cod2 , ds:data

45 : met2:

Error near line 45:Can not find in code "met name" that is required by lexem on line 45 on position 0

46 : cod2 ends

47 : END

Find 17 errors