

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КИЇВСЬКИЙ
ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО» ІНСТИТУТ
ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ КАФЕДРА МАТЕМАТИЧНИХ
МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

Система асемблювання і візуалізація роботи пам'яті

Виконавець роботи:
студент 4 курсу групи КА-54
Доманський Богдан Олегович

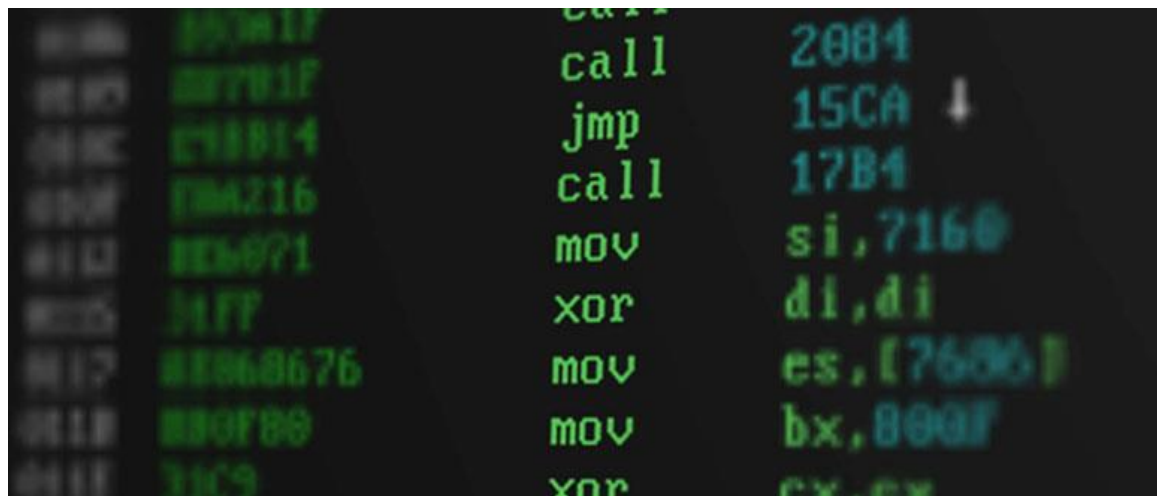
Науковий керівник:
к.т.н., в.о.зав.каф. ММСА
Тимощук Оксана Леонідівна

Мета дослідження: виконання процесу візуалізації роботи пам'яті, завдяки дослідженням в галузі асемблювання.

Об'єкт дослідження: синтаксис мови асемблера програми Corewar.

Предмет дослідження: асемблер, дизасемблер, віртуальна машина

Актуальність дослідження



```
0000 707A17 call 2084
0001 007B1F jmp 15CA
0002 C91B14 call 17B4
0003 00A216 mov si,7160
0004 00B071 xor di,di
0005 347F mov es,[7606]
0006 00B6B676 mov bx,800F
0007 000F80 xor cx,cx
```

Мова асемблера є потужним інструментом, що допомагає виявити всі особливості апаратного та програмного забезпечення.

Тож дане дослідження, тобто побудова системи асемблювання та віртуальної машини, є необхідним для розуміння архітектури комп'ютера.

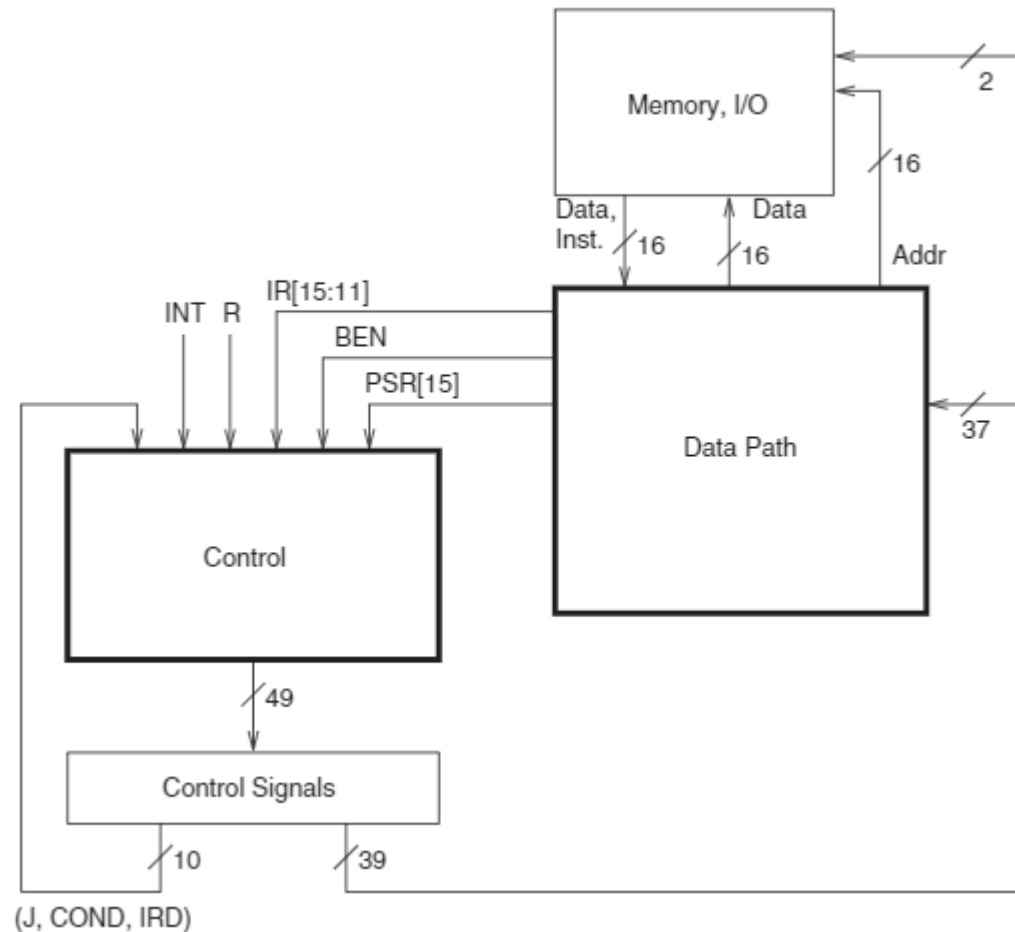
Постановка задачі

- Провести огляд предметної області, розглянути історію розвитку асемблера та віртуальної машини.
- Розглянути основні методи розробки на мові асемблера, та синтаксис мови асемблера
- Розробити програмний продукт згідно вивченим матеріалам

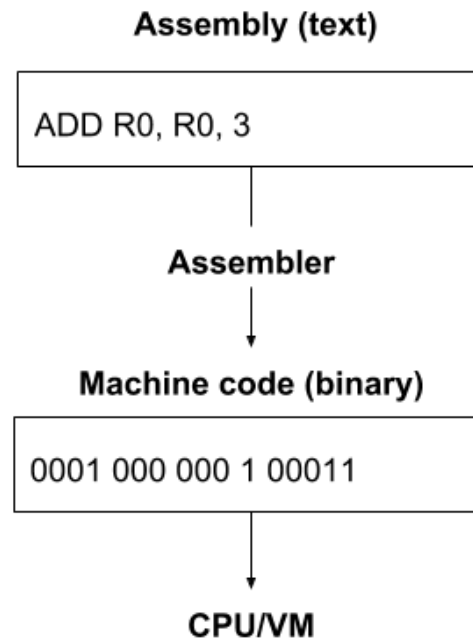
Теоретичні відомості

- Транслятор
- Процес компіляції
- Асемблер
- Віртуальна машина

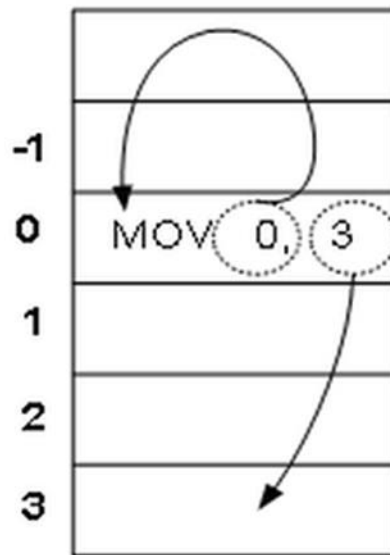
Приклад віртуальної машини LC-3



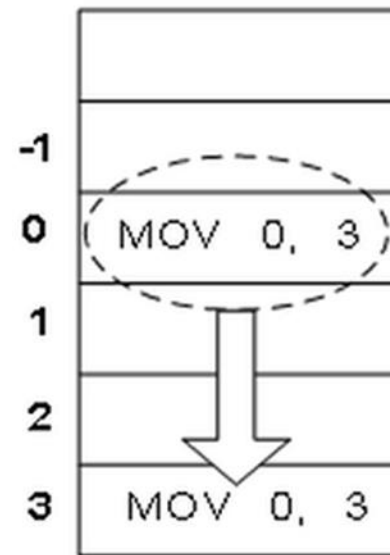
Процес переходу мови асемблера до виконання в VM



Інструкція MOV копіює сама себе в пам'яті



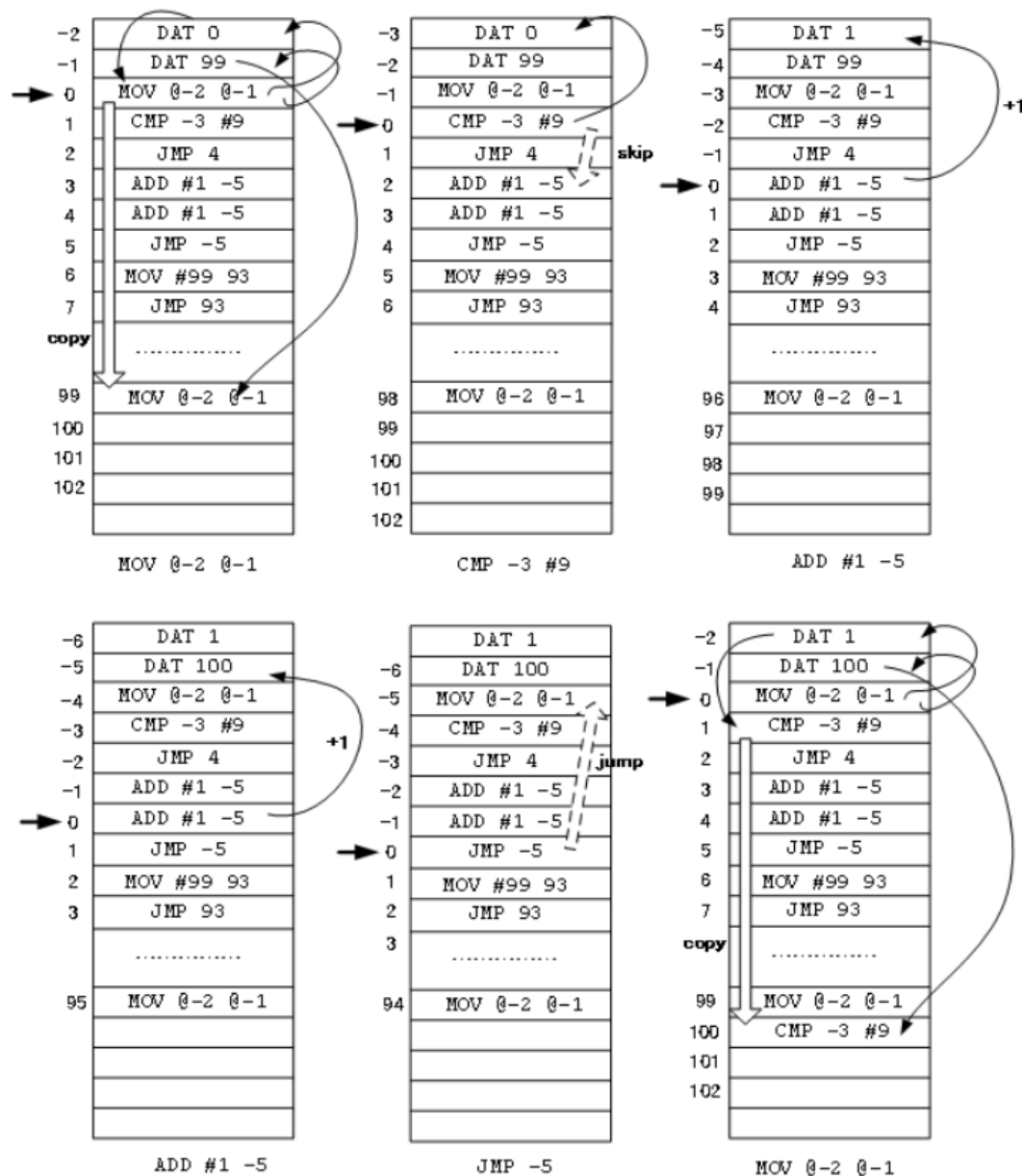
Before



After

Робота програми

Єдина завдання програми полягає в тому, щоб скопіювати саму себе в інше місце в пам'яті віртуальної машини MARS і продовжувати виконання з нового місця.



Синтаксис мови асемблера програми Corewar

Мова асемблера підпорядковується правилу — в одному рядку лише одна інструкція. Мова асемблера має певний набір з 16 операцій. Кожна з яких приймає від одного до трьох аргументів.

РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ

- Асемблер
- Дизасемблер
- Віртуальна машина
- Модуль візуалізації роботи пам'яті віртуальної машини

Асемблер

```
1  .name "Bot_1"
2  .comment "gd?"
3
4  start:
5  live %66
6  st r15, -30
7  kek:
8  live %66
9  add r13, r12, r15
10 st r1, -9
11 live %66
12 st r5, 50
13 fork %10
14 zjmp %:start
15
```



```
112  0000 0000 0000 0000 0000 0000 0000 0000
113  0000 0000 0000 0000 0000 0000 0000 0000
114  0000 0000 0000 0000 0000 0000 0000 0000
115  0000 0000 0000 0000 0000 0000 0000 0000
116  0000 0000 0000 0000 0000 0000 0000 0000
117  0000 0000 0000 0000 0000 0000 0000 0000
118  0000 0000 0000 0000 0000 0000 0000 0000
119  0000 0000 0000 0000 0000 0000 0000 0000
120  0000 0000 0000 0000 0000 0000 0000 0000
121  0000 0000 0000 0000 0000 0000 0000 0000
122  0000 0000 0000 0000 0000 0000 0000 0000
123  0000 0000 0000 0000 0000 0000 0000 0000
124  0000 0000 0000 0000 0000 0000 0000 0000
125  0000 0000 0000 0000 0000 0000 0000 0000
126  0000 0000 0000 0000 0000 0000 0000 0000
127  0000 0000 0000 0000 0000 0000 0000 0000
128  0000 0000 0000 0000 0000 0000 0000 0000
129  0000 0000 0000 0000 0000 0000 0000 0000
130  0000 0000 0000 0000 0000 0000 0000 0000
131  0000 0000 0000 0000 0000 0000 0000 0000
132  0000 0000 0000 0000 0000 0000 0000 0000
133  0000 0000 0000 0000 0000 0000 0000 0000
134  0000 0000 0000 0000 0000 0000 0000 0000
135  0000 0000 0000 0000 0000 0000 0000 0000
136  0000 0000 0000 0000 0000 0000 0000 0000
137  0000 0000 0000 0000 0000 0000 0000 0000
138  0100 0000 4203 700f ffe2 0100 0000 4204
139  540d 0c0f 0370 01ff f701 0000 0042 0370
140  0500 320c 000a 09ff da
```

Дизасемблер

```
1  00ea 83f3 426f 745f 3100 0000 0000 0000
2  0000 0000 0000 0000 0000 0000 0000 0000
3  0000 0000 0000 0000 0000 0000 0000 0000
4  0000 0000 0000 0000 0000 0000 0000 0000
5  0000 0000 0000 0000 0000 0000 0000 0000
6  0000 0000 0000 0000 0000 0000 0000 0000
7  0000 0000 0000 0000 0000 0000 0000 0000
8  0000 0000 0000 0000 0000 0000 0000 0000
9  0000 0000 0000 0000 0000 0029 676f 3f00
10 0000 0000 0000 0000 0000 0000 0000 0000
11 0000 0000 0000 0000 0000 0000 0000 0000
12 0000 0000 0000 0000 0000 0000 0000 0000
13 0000 0000 0000 0000 0000 0000 0000 0000
14 0000 0000 0000 0000 0000 0000 0000 0000
15 0000 0000 0000 0000 0000 0000 0000 0000
16 0000 0000 0000 0000 0000 0000 0000 0000
```



```
1  .name "Bot_1"
2  .comment "go?"
3
4      live    %66
5      st r15, -30
6      live    %66
7      add r13, r12, r15
8      st r1, -9
9      live    %66
10     st r5, -65486
11     fork    %-65526
12     zjmp    %-38
13
```

Віртуальна машина

[illegible]

STOPPED MUSIC: OFF

Cycles/second : 1000

Current cycle : 8717

Champ : kek	
Lives	0
Last live	6150

```

Champ : helltrain
Lives      2482
Last live  8715

```

Current period :

Last period :

CYCLE_TO_DIE	1386
--------------	------

CYCLE_DELTA	50
-------------	----

NBR_LIVE 02712 / 21

MAX_CHECKS	00000 / 10
------------	------------

Q : +100	W : +10	E : +1
R : -1	T : -10	Y : -100

... (.)< (MAY THE FORCE BE WITH YOU)

Висновки

Було ознайомлення з теоретичним матеріалом
Після цього було розглянуто віртуальну машину
MARS, інструкції мови Redcode, синтаксис мови
асемблера процесорної архітектури типу RISK на
основі програми Corewar.

В якості практичного прикладу було розроблено
програмне забезпечення з використанням мови
Сі. Розроблений програмний продукт
складається з асемблера, дизасемблера,
віртуальної машини та інструменту для
візуалізації роботи пам'яті віртуальної машини.

Дякую за увагу