Міністерство освіти і науки України

Національний університет “Львівська політехніка”



**Звіт**

З лабораторної роботи №1

З дисципліни: «Системне програмування»

На тему: «ОСОБЛИВОСТІ ПРОГРАМУВАННЯ З ВИКОРИСТАННЯМ32-РОЗРЯДНОГО АСЕМБЛЕРА»

Виконав: ст.гр. КІ-38

Черкасов В. С.

Прийняв: Козак Н.Б.

**Львів 2020**

**Мета:** Ознайомитись з програмною моделлю 32-розрядних мікропроцесорів Intel та оволодіти навиками створення програм, використовуючи 32-розрядний Асемблер.

**Теоретичні відомості**

Програмна модель комп'ютера, частиною якої є **програмна модель мікропроцесора**, яка містить 32 регістри в тій чи іншій мірі доступних для використання програмістом.   
Дані регістри можна розділити на дві великі групи:

* 16 регістрів користувача;
* 16 системних регістрів.

У програмах на мові асемблера регістри використовуються дуже інтенсивно. Більшість регістрів мають певне функціональне призначення.

**Регістри користувача**

Як випливає з назви, призначеними для користувача регістри називаються тому, що програміст може використовувати їх при написанні своїх програм. До цих регістрів відносяться (рис.1):

* вісім 32-бітових регістрів, які можуть використовуватися програмістами для зберігання даних і адрес (їх ще називають регістрами загального призначення (РЗП|)):
  + **eax/ax/ah/al**;
  + **ebx/bx/bh/bl**;
  + **edx/dx/dh/dl**;
  + **ecx/cx/ch/cl**;
  + **ebp/bp**;
  + **esi/si**;
  + **edi/di**;
  + **esp/sp**.
* шість сегментних регістрів: **cs, ds, ss, es, fs, gs**;
* регістри управління та стану:
  + регістр прапорів **eflags/flags**;
  + регістр покажчика команди **eip/ip**.

Чому багато з цих регістрів приведені з розділовою межею? (рис. 1)  
Ні, це не різні регістри — це частини одного великого 32-розрядного регістра. Їх можна використовувати в програмі як окремі об'єкти. Так зроблено для забезпечення працездатності програм, написаних для молодших 16-розрядних моделей мікропроцесорів фірми Intel, починаючи з i8086. Мікропроцесори i486 і Pentium мають в основному 32-розрядні регістри. Їх кількість, за винятком сегментних регістрів, таке ж, як і у i8086, але розмірність більше, що і відбито в їх позначеннях — вони мають приставку **e** (**Extended**).

#### Регістри загального призначення

Всі регістри цієї групи дозволяють звертатися до своїх “молодших” частин (див. рис. 1).

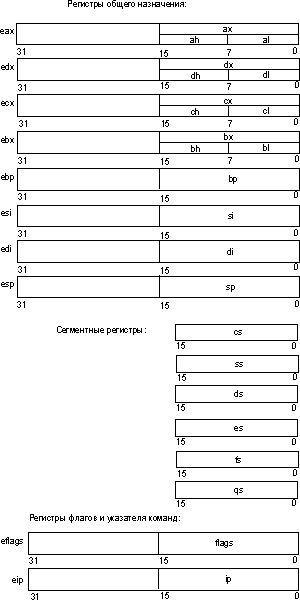
Відмітимо, що використовувати для самостійної адресації можна тільки молодші 16 і 8-бітові частини цих регістрів. Старші 16 біт цих регістрів як самостійні об'єкти недоступні.

До РЗП відносяться|:

* **eax/ax/ah/al** (Accumulator register) — акумулятор.

Застосовується для зберігання проміжних даних. У деяких командах використання цього регістра обов'язкове;

* **ebx/bx/bh/bl** (Base register) — базовий регістр.



**Рис. 1.** Регістри користувача мікропроцесорів i486 і Pentium

Застосовується для зберігання базової адреси деякого об'єкту в пам'яті;

* **ecx/cx/ch/cl** (Count register) — регістр-лічильник.

Застосовується в командах, що проводять деякі дії, що повторюються. Його використання часто неявно і приховано в алгоритмі роботи відповідної команди.   
Наприклад, команда організації циклу loop окрім передачі управління команді, що знаходиться за деякою адресою, аналізує і зменшує на одиницю значення регістра ecx/cx;

* **edx/dx/dh/dl** (Data register) — регістр даних.

Так само, як і регістр eax/ax/ah/al, він зберігає проміжні дані. У деяких командах його використання обов'язково; для деяких команд це відбувається неявно.

Наступні два регістри використовуються для підтримки операцій, що проводять послідовну обробку ланцюжків елементів, кожний з яких може мати довжину 32, 16 або 8 біт:

* **esi/si** (Source Index register) — індекс джерела.
* **edi/di** (Destination Index register) — індекс приймача (одержувача).

У архітектурі мікропроцесора на програмно-апаратному рівні підтримується така структура даних, як **стек**. Для роботи із стеком в системі команд мікропроцесора є спеціальні команди, а в програмній моделі мікропроцесора для цього існують спеціальні регістри:

* **esp/sp** (Stack Pointer register) — регістр покажчика стека.   
  Містить покажчик вершини стека в поточному сегменті стека.

**ebp/bp** (Base Pointer register) — регістр покажчика бази кадру стека.   
Призначений для організації довільного доступу до даних усередині стека.

**Виконання роботи**

1. Створити \*.exe програму, яка реалізовує обчислення, заданого варіантом виразу. **A={a[i]}** – попередньозаданий масив з **N** чисел. ***c, d*** - константи.
2. Переконатися у правильності роботи програми використовуючи VKDebug.
3. Скласти звіт про виконану роботу з приведенням тексту програми.
4. Дати відповідь на контрольні запитання.

**Варіант завдання:**

****

**Код програми:**

.586

.model flat, stdcall

option casemap: none

include masm32\include\windows.inc

include masm32\include\kernel32.inc

include masm32\include\masm32.inc

include masm32\include\debug.inc

include masm32\include\user32.inc

includelib masm32\lib\kernel32.lib

includelib masm32\lib\masm32.lib

includelib masm32\lib\debug.lib

includelib masm32\lib\user32.lib

.data

Arr dd 1,2,3,4,5

c\_parametr dd 1

d\_parametr dd 6

L dd 3

result dd 0

.code

start:

mov ecx,6

mov eax,1

mov ebx, 6

sub ebx, L

mov result, ebx

chck:

dec ecx

mov ebx,[Arr+ecx\*4-4]

cmp ebx, c\_parametr

jle decrement

cmp ebx, d\_parametr

jge decrement

mul ebx

cmp ecx,result

jne chck

decrement:

cmp ecx,result

je xt

dec result

jmp chck

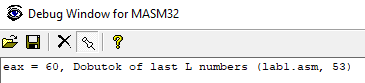
xt:

PrintDec eax,"Dobutok of last L numbers"

invoke ExitProcess,NULL

end start

**Результати виконання програми:**



**Висновок:** на даній лабораторній роботі я опанував роботу з програмною моделлю 32 розрядних процесорів Intel та оволодів навиками створення програм, використовуючи 32 розрядний асемблер.