МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп‘ютерних систем

**Лабораторна робота №3**

**Ознайомлення з методами адресації даних**

**Варіант №21**

Виконав:

Студент групи КВ-51

Тимошенко І.О.

Київ

2016

**Постановка задачі**

1. Створити сегмент кодів (наприклад, з ім‘ям Code1), в якому реалізувати наступне

завдання: якщо сума індексів за адресами I1,I2,I3 парна, тоді в елемент масиву

А1[i1,i2,i3] записується сегментна частина логічної адреси цього елемента, а якщо

хоча б один з індексів непарний, тоді в елемент масиву А1[i1,i2,i3] записується

зміщення в сегменті цього елемента.

2. Створити ще один логічний сегмент даних (наприклад, з ім‘ям Data2), в якому

задати директиву **A2 dw 4 dup (5 dup (6 dup (0)))**

3. Створити ще один логічний сегмент кодів (наприклад, з ім‘ям Code2), в якому за

допомогою ланцюгових команд переписати масив А1 в А2 та підрахувати в масиві

А2 кількість елементів, які не містять сегментні частини логічних адрес.

Забезпечити безумовну передачу управління з сегменту Code1 в сегмент Code2 за

допомогою непрямої міжсегментної передачі управління.

**Код програми**

**.386**

**Data1 segment para use16**

**I1 dw ?**

**I2 dw ?**

**I3 dw ?**

**A1 dw 4 dup (5 dup (6 dup (0)))**

**Data1 ends**

**Code1 segment use16**

**ASSUME CS:Code1, DS:Data1**

**start:**

**mov ax, Data1**

**mov ds, ax**

**mov i1, 0**

**mov i2, 0**

**mov i3, 0**

**\_loop:**

**imul bx, i1, 5\*6\*2**

**imul bp, i2, 6\*2**

**xor si, si**

**mov si, i3**

**shl si, 1**

**lea di, [bp + si]**

**mov ax, i1**

**add ax, i2**

**add ax, i3**

**test ax, 1**

**jnz short odd**

**mov ax, i1**

**test ax, 1**

**jnz short odd**

**mov ax, i2**

**test ax, 1**

**jnz short odd**

**lea si, A1[bx + di]**

**mov A1[bx + di], si**

**jmp short end\_if**

**odd:**

**mov A1[bx + di], ds**

**end\_if:**

**inc i3**

**cmp i3, 6**

**jb \_loop**

**mov i3, 0**

**inc i2**

**cmp i2, 5**

**jb \_loop**

**mov i2, 0**

**inc i1**

**cmp i1, 4**

**jb \_loop**

**jmp far ptr \_label**

**Code1 ends**

**Data2 segment para use16**

**A2 dw 4 dup ( 5 dup (6 dup (0)))**

**Data2 ends**

**Code2 segment use16**

**\_label label far**

**Assume cs:Code2, ds:Data1, es:Data2**

**start2:**

**mov ax, Data2**

**mov es, ax**

**lea di, A2**

**lea si, A1**

**mov cx, 4\*5\*6**

**rep movsw**

**lea di, A2**

**mov cx, 4\*5\*6**

**mov ax, ds**

**mov bx, -1**

**cld**

**\_loop2:**

**repne scasw**

**inc bx**

**cmp cx, 0**

**jne \_loop2**

**mov ax, 4C00h**

**int 21h**

**Code2 ends**

**end start**