**Лабораторна робота №5**

**Ланцюжкові команди**

**Ціль роботи:** навчитися розробляти програми на мові Асемблер з використанням ланцюжкових команд

**Короткі теоретичні відомості**

Ланцюжкові команди використовуються для обробки великих обсягів даних, що знаходяться у сусідніх адресах пам'яті.

Ланцюжок може містити до 64 Кбайт даних. Кожна ланцюжкова операція реалізується трьома командами, що визначають типи операндів – байт, слово або подвійне слово. У ланцюжковій команді можливо не задавати тип операндів. Транслятор асемблера самостійно визначає тип елементів ланцюжків по їхньому опису й генерує одну із трьох машинних команд, що дозволяють обробляти байти, слова, подвійні слова. Над ланцюжками даних можуть виконуватися наступні операції:

• пересилання ланцюжка: **movs** адреса\_приймача, адреса\_джерела

• порівняння ланцюжків: **cmps** адреса\_приймача, адреса\_джерела

• сканування ланцюжка: **scas** адреса\_приймача

Організація роботи ланцюжкових команд

До особливостей ланцюжкових операцій можна віднести: розташування областей даних у пам'яті, вибір напрямку обробки даних, автоматичне просування до наступного елемента даного ланцюжка.

Для організації роботи ланцюжкових команд необхідно виконати наступні дії:

• сформувати фізичні адреси ланцюжків – адреса\_джерела, адре- са\_приймача;

• визначити початкові зсуви інформації в ланцюжках;

• визначити порядок зміни адрес у ланцюжках;

• визначити кількість повторів операції.

Застосування ланцюжкових команд накладає обмеження на області розташування даних. Оброблювані дані можуть перебувати в області пам'яті, описуваної сегментним регістром DS, тобто в поточному сегменті даних. Результати можуть міститися тільки в додатковому сегменті даних, який адресується сегментним регістром ES. Знаходження інформації всередині ланцюжків визначається по зсуву, занесеним в спеціальні індексні регістри. Індексний регістр SI визначає розташування інформації для ланцюжка-джерела, а індексний регістр DI – для ланцюжка-одержувача. Повні фізичні адреси для операндів ланцюжкових команд визначаються наступними регістровими парами: адреса\_ джерела – пари DS:SI; адреса\_приймача – пари ES:DI. Ланцюжкові команди самі виконують модифікацію регістрів SI й DI, що адресують операнди, забезпечуючи тим самим автоматичне просування по ланцюжку. Вміст регістрів SI й DI буде змінюватися на один, два або чотири відповідно до розмірності оброблюваних даних: байт, слово, подвійне слово

**Завдання:**

Сформувати масив даних. Перенести елементи масиву в пустий масив у зворотному напрямку.

**Рішення:**

Dat Segment

len equ 13

source db 'Hello, World!','$'

dest db len DUP (' ')

Dat EndS

Codl Segment

assume ds:data, es:data

start:

mov ax, dat

mov ds, ax

mov es, ax

cli

lea si, source

lea di, dest

mov cx, len + 1

rep movsb

mov si, offset dest

mov bx, 0

mov cx, len / 2

L1:

mov bp, len - 1

sub bp, bx

mov al, [si + bx]

mov dl, [si + bp]

mov [si + bx], dl

mov [si + bp], al

add bx, 1

loop L1

lea dx, dest

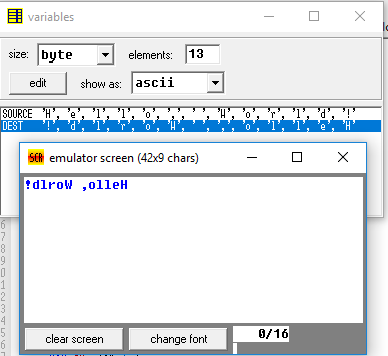
mov ah, 09h

int 21h

Codl EndS

End start

**Результат роботи програми:**



**Висновок**: на цій лабораторній роботі я навчився розробляти програми на мові Асемблер з використанням ланцюжкових команд.