Computer Organization & Assembly Language

Lab-7

Array, dup and source index register in Assembly Language-1

What is Array?

Array is a collection of character in sequence.

Why do we learn Array?

To Store many characters with single variable name in sequence in memory.

Where to initialize Array?

Array is defined in .data directives of program as variable.

How to initialize Array?

In same way as variable but with multiple values.

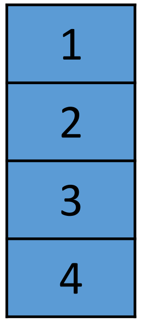
```
.model small
.stack 100h
.data

.code
Main proc

Main endp
End Main
```

Name of Array	Size	Value
Arr1	db	1,2,3,4
Arr1	db	‘a’,’b’,’c’
Arr1	db	‘abc’

Arr1 initialized in memory like



Array, dup and source index register in Assembly Language-2

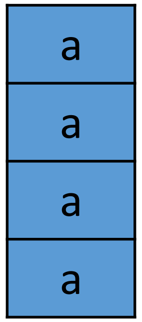
How to initialize Array?

In same way as variable but with multiple values.

Name of Array	Size	Value
Arr1	db	‘a’,’a’,’a’,’a’ (With same value in each cell)

You can use either way of DUP (Duplicate)

Arr1	db	4 Dup(‘a’)



Or you can initialize as unassigned

Arr1	db	?,?,?,?
Arr1	db	4 Dup(?)

Array, dup and source index register in Assembly Language-3

How to access Array?

```
.model small
.stack 100h
.data
Arr1        db          1,2,3,4
.code
Main proc
Mov dx, @data
Mov  ds,ax


Mov si, offset arr1
Main endp
End Main
```

Array Initialized

Heap memory initialized (mandatory to access variable)

Accessing address of first array into Source Index Register using offset

Address	Data
ah001	1
ah002	2
ah003	3
ah004	4

Array, dup and source index register in Assembly Language-4

How to access Array?

```
.model small
.stack 100h
.data
Arr1        db          1,2,3,4
.code
Main proc
Mov dx, @data
Mov  ds,ax
Mov si, offset arr1

Mov ds,[si]
mov ah,2
Int 21h

Main endp
End Main
```

Bracket form to access value from address. It will print the first value of the array.

Address	Data
ah001	1
ah002	2
ah003	3
ah004	4

Array, dup and source index register in Assembly Language-5

How to access Array?

```
.model small
.stack 100h
.data
Arr1 db 49,50,51,52
.code
Main proc
Mov ax, @data
Mov  ds,ax
Mov si, offset arr1
Mov dx,[si]
mov ah,2
Int 21h
Inc si
Mov dx,[si]
mov ah,2
Int 21h
mov ah,4ch
int 21h
Main endp
End Main
```

ASCII values

Add 1 in source index register to access the 2nd value from the address.

Address	Data
ah001	1
ah002	2
ah003	3
ah004	4

Program to print an two array values in Assembly Language-6

```
.model small
.stack 100h
.data
Arr1 db 49,50,51,52
.code
Main proc
Mov ax, @data
Mov  ds,ax
Mov si, offset arr1
Mov dx,[si]
mov ah,2
Int 21h
Inc si
Mov dx,[si]
mov ah,2
Int 21h
mov ah,4ch
int 21h
Main endp
End Main
```

DosBox Commands

- Edit Filename.asm (to create new file if not exists/open existing file)
- MASM Filename.asm; (to convert into object file using MASM assembler)
- LINK Filename.obj; (to convert object file into execution file using linker)
- To execute the exe file you just created,
  - Filename.exe (it will execute)

- NOTE: (Semicolon is mandatory while converting via assembler and linker only)

Get number in the form of array and display it in Assembly Language-7

```
dosseg
.model small
.stack 100h
.data
msg db 'Please 5 Digits in terms of array: $'
array db 6 dup('$')
.code
main proc
mov ax,@data
mov ds,ax
mov dx,offset msg
mov ah,9
int 21h
mov bl,','
lea si,array
```

Load Effective Address
It is an indirect instructions used as pointer in which first variables points the address of second variable.

```
l1:
mov ah,1
int 21h
cmp al,13
je Print
cmp al,bl
je l1
mov [si],al
inc si
jmp l1
Print:
mov dx,10
mov ah,2
int 21h
mov dx,13
mov ah,2
int 21h
```

placing in array

```
mov dx,offset array
mov ah,9
int 21h
mov ah,4ch
int 21h
main endp
end main
```

displaying array to get confirm numbers are placed

DosBox Commands

- Edit Filename.asm (to create new file if not exists/open existing file)
- MASM Filename.asm; (to convert into object file using MASM assembler)
- LINK Filename.obj; (to convert object file into execution file using linker)
- To execute the exe file you just created,
  - Filename.exe (it will execute)

- NOTE: (Semicolon is mandatory while converting via assembler and linker only)