**Computer Organization and Assembly Language**

|  |  |
| --- | --- |
| **Lab 11** | |
| **Topic** | 1. String operations 2. Repeat instructions |

**PART 1**

**String Examples**

**Simple String(Example) movsb(Example)**

|  |  |
| --- | --- |
| jmp start  data1 db 'Ali, Irfan, Aslam, Imran';  data2 db 100 dup (?)  start:  lea si, data1  lea di, data2  mov cx, 21  l1:  mov al, [si]  mov [di], al  inc si  inc di  loop l1  .exit | jmp start  data1 db 'Ali, Irfan, Aslam, Imran';  data2 db 100 dup (?)  start:  lea si, data1  lea di, data2  mov cx, 21  l1:  movsb  loop l1  .exit |

**Using loop instruction(Example) Using REP instruction(Example)**

|  |  |
| --- | --- |
| jmp start  data1 db 'Ali, Irfan, Aslam, Imran';  data2 db 100 dup (?)  start:  lea si, data1  lea di, data2  mov cx, 21  l1:  movsb  loop l1  .exit | jmp start  data1 db 'Ali, Irfan, Aslam, Imran';  data2 db 100 dup (?)  start:  lea si, data1  lea di, data2  mov cx, 21  REP MOVSB  .exit |

**Using SCAS instruction(Example) Using CMPS instruction(Example)**

|  |  |
| --- | --- |
| STR1 db 'AAliah',0  Lea di, STR1;  MOV AL, 'A';  MOV CX, 7;  REPE SCASB | jmp start:  STR1 db 'AAliBA',0  STR2 db 'AAliah',0  start:  Lea di, STR1;  Lea si, STR2;  MOV CX, 7;  REPE CMPSB |

**Using LODSB instruction(Example) Using STOSB instruction(Example)**

|  |  |
| --- | --- |
| jmp start:  STR1 db 'AAliaA',0  STR2 db 'AAliBA',0  count db ?  start:  Lea di, STR1;  Lea si, STR2;  MOV CX, 7;  L1:  LODSB  SCASB  je L2  jne L3  L2:  inc count  L3:  loop L1 | jmp start:  STR1 db 'AAliaA',0  STR2 db 7 DUP(?),0  start:  Lea si, STR1;  Lea di, STR2;  MOV CX, 7;  L1:  LODSB  STOSB  loop L1 |

**Traversing array from left to right Traversing array from right to left**

|  |  |
| --- | --- |
| lea si,array1  mov cx,17  cld ;clear the direction flag  ;increments the si and di in string operations  rep lodsb  .exit  array1 db '10,January, 2018' | lea si,array1  mov cx,17  add si,16  std ;set direction flag  ;decrements the si and di in string operations  mov cx,17  rep lodsb  .exit  array1 db '10,January, 2018' |

**NOTES:**

|  |  |
| --- | --- |
| **movsb** | 1. Mov [DI],[SI] 2. Inc si 3. inc di |
| **movsw** | 1. [DI]🡨[SI] 2. add si,2 3. add di,2 |
| **scasb** | 1. cmp al,[DI];ZF=1 if same 2. inc di |
| **scasw** | 1. cmp ax,[DI];ZF=1 if same 2. add di,2 |
| **cmpsb** | 1. cmp [si],[di] 2. inc si 3. inc di |
| **cmpsw** | 1. cmp [si],[di] 2. add si,2 3. add di,2 |
| **lodsb** | 1. mov al,[si] 2. inc si |
| **lodsw** | 1. mov ax,[si] 2. add si,2 |
| **stosb** | 1. mov [di],al 2. inc di |
| **stows** | 1. mov [di],ax 2. add di,2 |

**PART 2**

***NOTE: USE STRING OPERATIONS MOVS, LODS, CMPS, STOS, SCAS.***

## Task 1:

Copy the **Array1** into **Array2** using String Instructions only.

Let Array1 db ‘Computer Organization and Assembly Language’

Array2 db 200 dup(?)

SOL:

Jmp start

array1 db 'Computer Organization and Assembly Language',0

array2 db 200 dup(0)

start:

lea si,array1

lea di,array2

mov cx,43

REP MOVSB

ret

**Task 2:**

Find the length of string using string instructions.

For example:

String1 db ‘I am a programmer’, 0

Size db ?

SOL:

Jmp start

string1 db 'I am a programmer',0

size db 0

start:

mov al,0

lea di,string1

L1:

SCASB

JZ exit

INC size

Jmp L1

exit:

ret

## Task 3:

Find the **Count** of a character ‘e’ in the given string.

‘UNIVERSITY OF CENTRAL PUNJAB, LAHORE’

SOL:

Jmp start

array1 db 'UNIVERSITY OF CENTRAL PUNJAB, LAHORE',0

count db 0

start:

lea di,array1

mov al,'E'

L1:

SCASB

JNZ next

INC count

next:

CMP [di],0

JZ exit

Jmp L1

exit:

ret

## Task 4:

Write a subroutine to concatenate two strings and return the updated string, use stack for taking parameters.

//String Concat(string &a,string &b,string &c) ;address of strings are passed as a reference

//{c=a+b

//return c;

//}

SOL:

Jmp start

stringConcat:

push bp

mov bp,sp

mov si,[bp+8]

mov di,[bp+4]

L1:

movsb

CMP [si],0

JZ L2

JMP L1

L2:

mov si,[bp+6]

L3:

movsb

CMP [si],0

JZ ex

JMP L3

ex:

pop bp

ret

start:

lea bx,arr1

PUSH bx

lea bx,arr2

PUSH bx

lea bx,arr3

PUSH bx

CALL stringConcat

POP bx

POP bx

POP bx

.exit

arr1 db 'I will write ',0

arr2 db 'my own code',0

arr3 db 0

## Task 5:

Write a subroutine that copy the string1 to string2 in the reverse order.

Example: array1 db ‘computer’

After copying.

array2 db ‘retupmoc’

SOL:

Jmp start

arr1 db 'computer',0

arr2 db 0

reverse:

push bp

mov bp,sp

mov di,[bp+6]

mov al,0

L1:

SCASB

JZ L2

INC cx

JMP L1

L2:

mov di,[bp+4]

mov si,[bp+6]

dec cx

add si,cx

INC cx

L3:

std

LODSB

cld

STOSB

loop L3

pop bp

ret

start:

lea bx,arr1

push bx

lea bx,arr2

push bx

CALL reverse

pop bx

pop bx

ret

## Task 6:

Write a subroutine that inverts the case of all the vowels in a string.

Example:

Array1 db ‘AbCdefGhIjklmnOpqrsTuvwXyz’

After execution.

Array1 db ‘abCdEfGhijklmnopqrsTUvwXyz’

SOL:

Jmp start

arr1 db 'AbcUizX',0

vowels db 'aeiouAEIOU',0

invertVowel:

push bp

mov bp,sp

mov si,[bp+4]

L1:

lea di,vowels

LODSB

L2:

SCASB

JNZ noInvert

dec si

xor [si],0x20

inc si

Jmp next

noInvert:

CMP [di],0

JNZ L2

next:

CMP [si],0

JNZ L1

pop bp

ret

start:

lea bx,arr1

push bx

CALL invertVowel

pop bx

ret

## Task 7:

Write a subroutine that takes the address of the array and reverses the words in the array.

Example:

Array1 db ‘I am a student of BSCS.’

After execution

Array1 db ‘BSCS of student a am I’

SOL:

Jmp start

arr1 db 'I am a student of BSCS',0

arr2 db 0

invertWords:

push bp

mov bp,sp

mov di,[bp+6]

mov al,0

len:

SCASB

JZ next

INC cx

Jmp len

next:

mov si,[bp+6]

mov di,[bp+4]

mov bx,[bp+4]

add di,cx

dec di

mov cx,0

L1:

LODSB

CMP al,' '

JZ L2

CMP al,0

JZ L2

INC cx

Jmp L1

L2:

dec si

dec si

std

L3:

MOVSB

loop L3

cld

INC si

L4:

LODSB

CMP al,' '

JZ L5

CMP al,0

JZ L5

Jmp L4

L5:

std

dec si

MOVSB

add si,2

mov cx,0

cld

CMP bx,si

JNZ L1

pop bp

ret

start:

lea bx,arr1

PUSH bx

lea bx,arr2

PUSH bx

CALL invertWords

POP bx

POP bx

ret