**INDEX**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr.No.** | **Date** | **Experiment Performed** | **Page**  **No** | **Sign** | **Remark** |
| **1** |  | Write an X86/64 ALP to accept five 64 bit Hexadecimal numbers from user and store  them in an array and display the accepted numbers. |  |  |  |
| **2** |  | Write an X86/64 ALP to accept a string and to display its length. |  |  |  |
| **3** |  | Write an X86/64 ALP to find the largest of given Byte/Word/Dword/64-bit numbers. |  |  |  |
| **4** |  | Write a switch case driven X86/64 ALP to perform 64-bit hexadecimal arithmetic operations (+,-,\*, /) using suitable macros. Define procedure for each operation. |  |  |  |
| **5** |  | Write an X86/64 ALP to count number of positive and negative numbers from the array. |  |  |  |
| **6** |  | Write X86/64 ALP to perform non-overlapped block transfer without string specific  instructions. Block containing data can be defined in the data segment. |  |  |  |
| **7** |  | Write X86/64 ALP to perform overlapped block transfer with string specific instructions  Block containing data can be defined in the data segment. |  |  |  |
| **8** |  | Write X86/64 ALP to perform multiplication of two 8-bit hexadecimal numbers. Use  successive addition and add and shift method. (use of 64-bit registers is expected). |  |  |  |
| **9** |  | Write x86 ALP to find the factorial of a given integer number on a command line by using recursion. Explicit stack manipulation is expected in thecode. |  |  |  |
| **10** |  | Study Assignment:  Motherboards are complex. Break them down, component by component, and Understand how they work. Choosing a motherboard is a hugely important part of building a PC. Study- Block diagram, Processor Socket, Expansion Slots, SATA, RAM, Form Factor, BIOS, Internal Connectors, External Ports, Peripherals and Data Transfer, Display, Audio, Networking, Overclocking, and Cooling. 4.  <https://www.intel.in/content/www/in/en/support/articles/000006014/boards-and-kits/desktop-boards.html> |  |  |  |

|  |
| --- |
| **Subject: MICROPROCESSOR LAB (MPL)**  **Name:** |
| **Class: Roll No.:** |
| **Semester: Sem-II Year: 2020-21** |
| **Date of Performance: Date of Submission:** |
| **Examined:** |

**Assignment No-03**

**Title:-** Find the largest of given numbers.

**Assignment Name: -** an X86/64 ALP to find the largest of given Byte/Word/Dword/64-bit numbers.

**Objective-**

* To understand the assembly language program
* To understand 64 bit interrupt.

**Outcome-**

* Students will be able to write code for how to find the largest of given
* Students will be able to understand different assembly language instruction.

### [Prerequisite](http://dictionary.reference.com/browse/prerequisite) -

System call of Unix for Assembly language Program.

**Hardware Requirement-**

Desktop PC

**Software Requirement-**

Ubuntu 14.04,

Assembler: NASM version 2.10.07

Linker: ld

**Introduction:-**

**Theory:**

**Algorithm:**

1. Start

2. Initialise section .data

3. Define variable for array,large

4. Using cmp instruction find larger number from array.

5. Display largest number

6. Terminate program using system call

6. Stop

**Conclusion:-** Hence we implemented an ALP find the largest of given array.

**Questions:-**

Q.1.Explain macro used with Example?

Q.2 Explain CMP instruction?

Q.3 Draw and explain TSS segment of 80386?

|  |
| --- |
| **Subject: MICROPROCESSOR LAB (MPL)**  **Name:** |
| **Class: Roll No.:** |
| **Semester: Sem-II Year: 2020-21** |
| **Date of Performance: Date of Submission:** |
| **Examined:** |

**Assignment No-08**

**Title:-**Multiplication

**Assignment Name: -** Write X86/64 ALP to perform multiplication of two 8-bit hexadecimal numbers. Use successive addition and add and shift method. Accept input from the user.

**Objective-**

* To understand the different algorithm for multiplication.
* To understand how to write procedure.

**Outcome-**

* Students will be able to write code for doing multiplication.

### [Prerequisite](http://dictionary.reference.com/browse/prerequisite) -

System call of Unix for Assembly language Program.

**Hardware Requirement-**

Desktop PC

**Software Requirement-**

Ubuntu 14.04,

Assembler: NASM version 2.10.07

Linker: ld

**Introduction:-**

**Guidelines for the algorithm:**

1. Display the menu.

Enter “1” – “ADD AND SHIFT METHOD.”

Enter “2” – “SUCCESSIVE ADDITION METHOD”.

Enter “3” – EXIT

2) Take choice from user then go to the respective subroutines.

ADD AND SHIFT METHOD

1. Initialize code and bss sections.
2. Accept multiplier and multiplicand variables in data segment.
3. Initialize product variable to zero.
4. Set count as number of bits in operand, which is 8.
5. Shift product to left by 1 bit and insert zero as LSB.
6. Transfer MSB of multiplier to carry flag by rotating it to left.
7. Check if carry flag is set or not. If yes add multiplicand to product.
8. Decrement count by 1.
9. Check count=0 else repeat step 5 through step 9 till count=0.
10. Display the final product.

#### SUCCESSIVE ADDITION METHOD

1. Define product=0.
2. Set count=multiplicand.
3. Add product=product + multiplier.
4. Decrement count.
5. Repeat step 3 and 4 till count=0
6. Display product variable value as final product.

**Conclusion: -** Hence we implemented an ALP to do multiplication.

**Questions:-**

1. Explain successive addition algorithm with example?
2. Explain what is Interrupt?