ISLAMIC UNIVERSITY OF TECHNOLOGY (IUT)
ORGANISATION OF ISLAMIC COOPERATION (OIC)
Department of Computer Science and Engineering (CSE)

SEMESTER FINAL EXAMINATION **WINTER SEMESTER, 2016-2017**

DURATION: 3 Hours **FULL MARKS: 150**

CSE 4503: Microprocessors and Assembly Language

Programmable calculators are not allowed. Do not write anything on the question paper.

There are **8 (eight)** questions. Answer any **6 (six)** of them.

Figures in the right margin indicate marks.

---

1. a) i. What is PLT and GOT? 6
   ii. What's the motivation for using PLT stub as function trampoline?
   iii. Why NOT call the shared library functions directly?
   b) Explain how PLT and GOT is used to resolve the base address for any function from any shared libraries which is compiled with Position Independent Code parameter. (3 states) 6
   c) What is meant by the term Buffer Overflow? How has this problem been resolved in the existing Linux library? 5
   d) Code to call a function with 3 parameters from "_start" stub in intel x86 architecture. Demonstrate the same code in x86_64 architecture. 8

2. a) What is Address Space Layout Randomization (ASLR)? Why Global Offset Table has the term 'Global' and 'Offset' in it? 4
   b) Show the byte representation of the following commands- 6
      i. MOV AX, [6072H]
      ii. MOV CX, DX
   c) Write two assembly programs, 15
   Program-I will take input and display the inputs (see test case) and store it in the stack.
   Consider the content of the stack won't change.
   Now, Program-II will take the values from stack and display the outputs as shown below.

Program-I	**Program-II**
Enter Dividend: 10	Quotient: 3
Enter Divisor: 3	Remainder: 1

Note: Dividend will be maximum of 2 digits and divisor of at maximum 1 digit.

3. a) What are NULL Bytes? Write down its significance in the buffer exploits? 6
   b) Explain steps of how interrupt is processed in 8086 microprocessors? How does 8086 get the address of any particular Interrupt Service Routine? 6
   c) How is the 'Interrupt Vector Table' and 'Interrupt Service Routine' related? Specify, what they hold and why. 6
   d) Explain Call and return mechanism for a near procedure and show how the IP and Stack are affected by procedure calls. 7

4. a) Analyze the given code: 4

```
section .data
1.      name db "/bin/sh"
section .text
2.      global _start

_start:
        push 0
```

3. push name
4. mov rax,59
5. mov rdi,rsp
6. mov rsi,0
7. mov rdx,0
8. syscall

Write comments for instructions 1 to 8. Mention for what purpose the instructions have been used. 6

b) Write down the significance of Accumulator register before and after function calls by conventional programming. 15

c) Remove the NULL bytes from the given stub to generate a new code in 32bit assembly which can be used as a shellcode to exploit buffer. [You can work with decimal of the respective hex values, i.e. 10d= 0xA]

shell: file format elf32-i386

Disassembly of section .text:

```
08048060 <_start>:
 8048060:     b8 66 00 00 00          mov    eax,0x66
 8048065:     bb 01 00 00 00          mov    ebx,0x1
 804806a:     6a 00                   push   0x0
 804806c:     6a 01                   push   0x1
 804806e:     6a 02                   push   0x2
 8048070:     89 e1                   mov    ecx,esp
 8048072:     cd 80                   int    0x80
 8048074:     89 c2                   mov    edx,eax
 8048076:     b8 66 00 00 00          mov    eax,0x66
 804807b:     bb 0e 00 00 00          mov    ebx,0xe
 8048080:     6a 04                   push   0x4
 8048082:     54                      push   esp
 8048083:     6a 02                   push   0x2
 8048085:     6a 01                   push   0x1
 8048087:     52                      push   edx
 8048088:     89 e1                   mov    ecx,esp
 804808a:     cd 80                   int    0x80
 804808c:     b8 66 00 00 00          mov    eax,0x66
 8048091:     bb 02 00 00 00          mov    ebx,0x2
 8048096:     6a 00                   push   0x0
 8048098:     66 68 2b 67             pushw  0x672b
 804809c:     66 6a 02                pushw  0x2
 804809f:     89 e1                   mov    ecx,esp
 80480a1:     6a 10                   push   0x10
 80480a3:     51                      push   ecx
 80480a4:     52                      push   edx
 80480a5:     89 e1                   mov    ecx,esp
 80480a7:     cd 80                   int    0x80
```

5. a) Draw and add a brief explanation of the flag register in 80286. What are the new flags introduced in 80386 and 80486? 6+4

b) Draw the flow diagrams for the three I/O data transfer techniques and label the states. 15

6. a) State the differences of Memory mapped I/O and Isolated I/O with figures. 4

b) "Software interrupts are prioritized more than the hardware interrupts when they occur at the same time" - justify the statement. 4

c) Suppose, a multicore processor was carrying out multiple instructions which is obvious. 9

Also, each of the core was responsible for parallel execution of multiple instructions. How was this possible?
Contrast the ideas of Pipelining and Superscalar property in microprocessors.

d) Briefly mention and explain the pipelining stages of Floating Point Unit. 8

7. a) What is an addressing mode? What are different addressing modes? Explain Based Indexed addressing modes. 8

b) Write some instructions to replace each upper-case letter in the following string by lower case equivalent using Base addressing mode. 7

```
MSG  DB  'LIFE IS GOOD',$
```

c) The algorithm given below multiplies two unsigned numbers A and B. Using the algorithm, write a procedure MULTIPLY to multiply two numbers A and B. 10
[Assume, A and B are already stored in AX and BX.]

```
Product = 0
REPEAT
        IF LSB of B is 1
        THEN
        Product = Product + A
        END_IF
        Shift left A
        Shift right B
UNTIL B=0
```

8. a) Given, 7

```
MOV EAX, 0x2002b2(RIP)
```

Accessing the offset from RIP register is possible only in **x86_64** architecture, show how this is handled in **x86** architecture.

b) What is the necessity of using stack segment in assembly language programming? With suitable examples demonstrate how to perform push and pop operation in stack. 6

c) What's the purpose of stack pointer(SP) and Base Pointer (BP) registers? Show the changes in the stack as the function foo executes. Show each state of changes till the stack becomes empty again. Initial stack state is NULL. 2+10

```
int main() {
        foo(3,'A',10);
        return 0;
}
```

```
void foo(int a, char b, int c){
    int k;
    k= a;
    if(k>0){
            k--;
            foo(k,'A',10);
    }
    else return;
}
```