Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Department of Computer Science

Computer Architecture II
CSU34021

Tutorial 1
Intel's 32-bit Assembly with C/C++

Syed Asad Alam

Document History

Rev.	Date	Comment	Author
0.1	23-09-2020	Initial Draft	SAA
0.2	02-10-2020	Modifications to program 1 description	SAA
0.3	08-10-2020	Finalized the deadline	SAA
1.0	09-10-2020	Tutorial 1 released	SAA

1 Learning Outcomes

This lab satisfies the following learning outcomes of the course:

LO1 Write simple IA32 and x64 assembly language functions

LO2 Explain the IA32 and x64 procedure calling conventions

LO3 Write programs that mix C/C++ and IA32 or x64 assembly language functions

2 Exercises

2.1 Program 1

The following procedure evaluates the following polynomial:

$$x^2 + x + 1$$

```
int poly(int arg)
{
  int res;
  res = pow(arg,2);
  res = res + arg + 1;
  return(res);
}

int pow(int arg0, int arg1)
{
  int result,i;

  result = 1;
  for (i=1; i<=arg1;i++)
    result = result*arg0;
  return(result);
}
```

2.2 Program 2

The following procedure finds all multiples of K that are less than N by setting the corresponding of a N-sized array element to 1 whenever a multiple is found.

```
void multiple_k(uint16_t N, uint16_t K, uint16_t* array)
{
  for (uint16_t i = 0; i < N; ++i)
    {
      if((i+1)%K == 0)
        {
          array[i] = 1;
        }
      else
        {
          array[i] = 0;
        }
    }
}
```

2.3 Program 3

The following procedure calculates the factorial of a number N using recursion:

```
int factorial(int N)
{
  if (N==0)
    return 1;
  else
    return N*factorial(N-1);
}
```

Q1 Translate the code procedure above into IA32 assembly language using the basic code generation strategy outlined in lectures. You may then generate optimized versions of your code. The % operation can be implemented using the IA32 cwd and idiv instructions. Implement the above using Intel's x86 ISA. Follow all the calling conventions outined earlier. Also, take into account the word lengths specified in the functions above.

Q2 What is the maximum depth of the stack (in stack frames) during the calculation of factorial (5)? Draw a diagram showing the state of the stack at its maximum depth during the calculation of factorial (5).

Q3 Using Visual Studio (or similar), create a Win32 application with files t1.h and t1.asm containing the IA32 assembly language for poly (also pow which is called by poly), multiple_k and factorial . Use t1Test.cpp to test poly, multiple_k and factorial . Please note that the source code provided may need to be modified to work with the development environment you use.

Submission

Submit your answer, by combining your code files, diagram of the stack for the factorial (5) and a snapshot of the console window showing evidence that your program works in a single zip file clearly mentioning your name and ID, via Blackboard. The deadline is available at the course page (`https://www.scss.tcd.ie/~alams/csu34021.html`) and Blackboard.

Marks Distribution

This coursework will be marked out of 100. Translation of each program carries a maximum of 30 marks while the stack diagram carries a maximum of 10 marks.

Deadline

The deadline is: Friday, October 30, 2010 9 pm.