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

Department of Computer Science

Computer Architecture II
CSU34021

Tutorial 2
Intel's 64-bit Assembly with C/C++

Syed Asad Alam

Document History

Rev.	Date	Comment	Author
0.1	10-10-2020	Initial Draft	SAA
0.2	12-10-2020	Defined the procedures to be implemented	SAA
1.0	17-10-2020	Tutorial 2 released	SAA
1.1	26-09-2021	Modified for 2021-22 session	SAA
2.0	02-10-2021	v2.0 released	SAA

1 Learning Outcomes

This lab satisfies the following learning outcomes of the course:

LO1 Write simple x64 assembly language functions

LO2 Explain the x64 procedure calling conventions

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

2 Exercises

2.1 Program 1

The following procedure calculates the greatest common division (GCD) of two 64-bit integers:

```
long long gcd_recursion(long long a, long long b)
{
  if(b == 0)
    return a;
  else
    return gcd_recursion(b,a%b);
}
```

2.2 Program 2

The following procedure takes a user input through scanf, finds the maximum value of an array and adds the maximum value with the user input and prints the result while returning the sum. The user input should also be accessible from other C/C++ functions. A high level code is provided for reference. For finding the maximum value, a built in function has been used. Students are required to have their own implementation of finding the maximum value. They can either embed the logic within the use_scanf function or write another function and call it from use_scanf

```
long long use_scanf(long long array_size, long long* array)
{

  // Find the maximum value in an array
  // As a high level code, a built in function has been used
  // Students are required to have their own implementation of finding
  // the maximum value
  // They can either embed the logic within the use_scanf function or write
  // another function and call it from use_scanf
  long long *max_value = std::max_element(std::begin(array), std::end(array));

  long long inp_int;

  printf("Please enter an integer: ");
  scanf("%lld", &inp_int);

  long long sum = *max_value + inp_int;

  printf("The sum of the maximum value and user input (%lld, %lld): %lld\n",
         *max_value,inp_int,sum);

  return sum;
}
```

The scanf function requires two arguments. The first one is the format specifier (%lld) which can be defined as a string, similar to the string needed for printf and address of this string loaded as an argument and the second argument is the address of variable in memory where it will return the user input (as shown in the "C" code by &inp_int)

2.3 Program 3

The following are two procedures, with min5 calling min to calculate its return value.

```
_int64 min(_int64 a, _int64 b, _int64 c) {
    _int64 v = a;
    if (b < v)
        v = b;
    if (c < v)
        v = c;
    return v;
}

// inp_int: The user input in Program ''
_int64 min5(_int64 i, _int64 j, _int64 k, _int64 l)
{
  return min(min(inp_int, i, j), k, l);
}
```

Exercises

E1 Using Visual Studio (or similar), create a console application with files t2.h and t2.asm containing the x64 assembly language translations of the three programs mentioned above. Use t2Test.cpp to test these functions. Please note that the source code provided may need to be modified to work with the development environment you use.

E2 What is the maximum depth of the stack (in stack frames) during the calculation of gcd_recursion(14,21)? Draw a diagram showing the state of the stack at its maximum depth during the calculation of gcd_recursion(14,21).

Submission

Submit your answer, by combining your code files, diagram of the stack for the gcd_recursion(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/Syed.Asad.Alam/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 15, 2021 9 pm.