Computersystemen: Session 2

David Blinder, Jan Cornelis, Stijn Bettens, Tim Bruylants and Peter Schelkens

October 8, 2020

1 INTRODUCTION

All general-purpose registers on the 80386 represent 32-bit numbers. The assembler can process binary, decimal and hexadecimal notations of integers in the source code, which is written in assembly language. For example, the following immediate assignments are all translated to the exact same machine instruction:

```
mov eax, 10h      ; assigns decimal value 16
mov eax, 16       ; assigns decimal value 16
mov eax, 10000b   ; assigns decimal value 16
```

The assembler tool will then convert these numeric (human-readable) notations to bits as part of the binary instructions. However, displaying a value again as a decimal number on the screen requires writing code to explicitly convert the contents in a register to actual numeric characters that can be printed to the screen.

The goal of the following exercises is to learn to work with the stack. The top of the stack in the stack segment is pointed to by the register `ESP`. For more details, please take a look in the slides, or in the "Assembly Programming Compendium" on Canvas in the map "Manuals", in *section 4.3: The Stack*.

In the first part of this session, the goal is to print 32-bit integer values as human-readable decimal numbers to the screen. In practice this means that you will need to devise a way of 'extracting' the digits out of a decimal number. These digits can then be converted to the corresponding ASCII characters and printed to the screen.

2 HINTS

- Unsigned 32-bit numbers are in the range $[0; 2^{32} - 1] = [0; 4294967295]$. This implicates that the function should print a number that contains between 1 and 10 digits, depending on the value.
- The digits represent powers of 10 that can be calculated using division and modulus operations (DIV and IDIV instructions).
- The 80386 program counter (PC) uses ASCII codes to represent a set of printable built-in characters. The ASCII character codes for the digits '0' to '9' are sequential and start at ASCII code 30h (or 48). I.e. character '0' is ASCII code 48, '1' is ASCII code 49, and so on.
- Make use of function `02h` of interrupt `21h`.

3 References

1. Chapter 17 of the INTEL 80386 PROGRAMMER'S REFERENCE MANUAL 1986

4 Exercises

1. Write a program that takes a positive 32-bit number in EAX and prints it to the screen in decimal notation.

2. Write a program that takes a two's complement 32-bit number in EAX and prints it to the screen in decimal notation. (Use the code from exercise 1)

3. Write a program that prints out the nth Catalan number. The Catalan numbers C_n are a sequence of natural numbers indexed by n. They have various applications in combinatorial mathematics, and describe things such as: the number of possible binary search trees with n different keys, the number of ways a convex polygon can be triangulated, or the number of noncrossing partitions of an n-element set (cf. Fig. 4.1). The Catalan number C_n can be computed with the following recurrence relation:

$$C_0 = 1; \quad C_{n+1} = \sum_{i=0}^{n} C_i \cdot C_{n-i}, \forall n \in \mathbb{N} \tag{4.1}$$

   The first Catalan numbers for $n = 0, 1, 2, \ldots$ are:
   $1, 1, 2, 5, 14, 42, 132, 429, 1430, 4862, 16796, 58786, 208012, 742900, 2674440, 9694845, 35357670, \ldots$
   (Here too, use the code from exercise 1 to print the Catalan numbers)

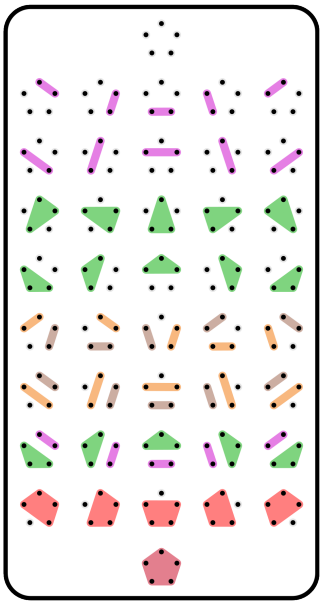

Figure 4.1: The $C_5 = 42$ noncrossing partitions of a 5-element set (source: `https://en.wikipedia.org/wiki/Catalan_number`)