Functions in Assembly

Functions? What's that

Alexander Wang alwang@student.42.us.org
42 Staff pedago@42.fr

Summary: Learn about functions in Assembly as well as the C Calling Convention.

Contents

Chapter I

Foreword

There is unrest in the Galactic Senate. Several thousand solar systems have declared their intentions to leave the Republic. This separatist movement, under the leadership of the mysterious Count Dooku, has made it difficult for the limited number of Jedi Knights to maintain peace and order in the galaxy. Senator Amidala, the former Queen of Naboo, is returning to the Galactic Senate to vote on the critical issue of creating an ARMY OF THE REPUBLIC to assist the overwhelmed Jedi....

Chapter II

Introduction

Welcome to the 4th project of this long series. In the first two projects, you learned about binary numbers and hexadecimals. In the project prior to this you learned about very simple Assembly constructs. In this project you will be learning about functions. Now, why functions? Because they are very nice. You wouldn't want to have a bajillion jumps going in every possible direction now do you nor do you want everything to be stuck inside of _start. So in order to compensate for this we will be using functions.

Chapter III

Goals

In this project, the goal is to learn about functions and how they work. Subsequently, you will also be learning about the C calling convention, the most commonly used calling convention in binary files as well as implementing it yourself in one of the projects.

Chapter IV

Mandatory part

- The language you will be using is x86 Assembly
- You will compile everything using GAS or "as".
- Everything here will be done in a Linux 32-bit Virtual Machine
- The autograder has been tested only on a Ubuntu 32-bit system if one exists.
- As such, it is highly recommended you use Ubuntu 32-bit.

Chapter V

Exercise 00: Maximum Numbers

	Maximum Numbers
Topics to study :	
Files to turn in : `min.s`	
Notes : `n/a`	

- Modify `max.s` so it finds the minimum instead.
- Save the new modified version in `min.s`

Chapter VI

Exercise 01: Calling Conventions

	Calling Conventions
Topics to study :	
Files to turn in : `callConvention.s`	
Notes :	

- Using the C calling convention, implement the following as a **function** that will be **called** in _start:

$$f(x, y, z) = \begin{cases} \text{x * y if z} > 0 \\ \text{x + y if z} < 0 \\ \text{x - y if z} = 0 \end{cases}$$

- Must *restore the stack* and remove all local variables if any are used before exiting the program.

- Save your program in `callConvention.s`

Chapter VII

Turn-in and peer-evaluation

Turn your work in using your `GiT` repository, as usual. Only work present on your repository will be graded in defense.