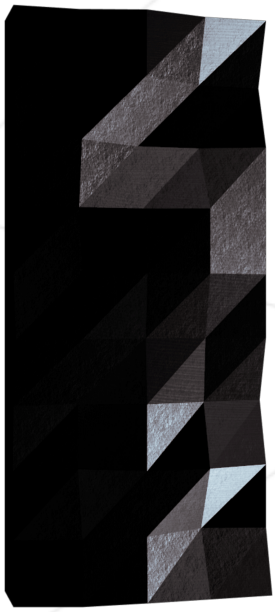

Introduction to Assembly

Finally we're learning Assembly

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

Summary: This project will have you learn about x86 Assembly and some basic constructs. In particular, you will be learning about registers and loops.

Contents

Chapter I

Foreword

Turmoil has engulfed the Galactic Republic. The taxation of trade routes to outlying star systems is in dispute.
Hoping to resolve the matter with a blockade of deadly battleships, the greedy Trade Federation has stopped all shipping to the small planet of Naboo.
While the Congress of the Republic endlessly debates this alarming chain of events, the Supreme Chancellor has secretly dispatched two Jedi Knights, the guardians of peace and justice in the galaxy, to settle the conflict....

Chapter II

Introduction

Welcome to the first project of this long series. In this first project, after going through the resources that you have hopefully read by this point prior to doing these exercises you would have had a pretty good understanding of the basics of binary numbers as well as hexadecimals. These numbers will appear pretty frequently throughout the rest of this series.

In this project, you will be learning about Assembly and have a relatively small taste in it. More precisely you will be learning about loops, and registers as well as a single system call that will be the groundwork for all the subsequent projects

Chapter III

Goals

The goal of this project is for you to learn simple Assembly constructs:

- Loops
- Infinite Loops
- System calls
- Registers

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: Exit Number

	Exit Number
Topics to study :	
Files to turn in : `exit.s`	
Notes : `n/a`	

- Change the number `exit.s` for the exit code from our very first project.

`Actually change the number. The autograder will check for this.`

Chapter VI

Exercise 01: Infinite Loops

	Infinite Loops
Topics to study :	
Files to turn in : `infLoop.s`	
Notes :	

- Create an Assembly program `infLoop.s` that is an infinite loop that only adds 2 to a register of your choosing.

Chapter VII

Exercise 02: Python to Assembly

	Python to Assembly
Topics to study :	
Files to turn in : `pyToAsm.s`	
Notes :	

- Translate the following Python Code into Assembly:

```
y = 0
for x in range(5, -5 -1):
    y += x
```

- Save your program in `pyToAsm.s`

`It doesn't need to compile but it must function exactly as the above snippet`

Chapter VIII

Exercise 03: Python to Assembly 2

	Python to Assembly 2
Topics to study :	
Files to turn in : `pyToAsm2.s`	
Notes :	

- Translate the following Python Code into Assembly that compiles and exits with the value of the resulting operations performed on it:

```
y = 5
for x in range(0, 10 2):
    y <<= 1
y -= 123
if y % 2 == 0:
    y <<= 1
elif y + 5 <= 42
    y >>= 1
elif y + 10 > 42:
    y += 1
else:
    y = 3
```

- Save your program in `pyToAsm2.s`

Chapter IX

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.