# Augean Stables

## Labour 04

Lou lou@42.us.org

*Summary:*

# Contents

# Chapter I

# Foreword

The fifth Labour of Hercules was to clean the Augean stables.
Eurystheus intended this assignment both as humiliating (rather than impressive, like the previous labours) and as impossible, since the livestock were divinely healthy (immortal) and therefore produced an enormous quantity of dung.
These stables had not been cleaned in over 30 years, and 3,000 cattle lived there.
However, Hercules succeeded by rerouting the rivers Alpheus and Peneus to wash out the filth.

Augeas reacted angrily because he had promised Hercules one tenth of his cattle if the job was finished in one day.
He refused to honour the agreement, and Hercules killed him after completing the tasks.
Hercules gave his kingdom to Phyleus, Augeas' son, who had been exiled for supporting Hercules against his father.

# Chapter II

# Introduction

Happily for you in this labour you won't have to clean stables of a very neglecting owner, you are coders for god sake!
So for this project you still have to undertake a similar labour as our hero so you can become a hero yourself as well.
As coders you won't have to clean stables but rather a very poorly written code with lots of leaks in it.
As our hero did with the two rivers you are going to be helped by two coding-rivers as well, they are called GDB and Valgrind.

From Indonesia with Love!

# Chapter III

# Goals

This project aims to help you get the basic notions of freeing memory.
You might not necessarily need to free much memory and you might not care now that you are writing rather small programs on powerful computers but imagine a program that's much more resource demanding or ran on a very limited server!
Then you're going to want to track every single and tiniest memory leak.

# Chapter IV

# General instructions

Using Valgrind and/or GDB or even without any of these tools you need to make sure that our program has no more memory leaks.
It needs to be perfectly waterproof (no leaks, waterproof, get it ;) ).

`As sure of yourself as you can be, it’s always a good thing to check and know how to use tools such as Valgrind and GDB.`

# Chapter V

# Mandatory part

For this you just need to turn in the same program provided but with no memory leaks. You can change whatever you want in it except the lines that are malloc-ing and the program needs to be performing the exact same task with the same result.

`Use Valgrind or GDB, you can never be too sure about leaks, they are sneaky!`

`Be sure that you didn’t alter the way the program works!`