

Lab 06: Operating Systems

The exercises marked with * are obligatory to complete the upcoming week's lecture.

The purpose of this lab is to introduce you to CPU and memory management when using Docker.

Remember to write useful commands in your cheat sheets.

Table of content

Exercise 01*	1
Exercise 02*	1
Exercise 03*	2

Exercise 01*

In this exercise the student is going to make a Python script that calculates the 35th Fibonacci number and prints the time it took the computer to calculate it. Below is the pseudo code for printing out the time it takes to calculate Fib35.

```
StartTime = CurrentTime;  
Fib(35);  
EndTime = CurrentTime;  
print(EndTime - StartTime);
```

Your tasks:

- Make a script that calculates Fib35

Hint:

<https://www.geeksforgeeks.org/python-program-for-program-for-fibonacci-numbers-2/>

- Make the script print the calculation time.

Hint: If the calculation is too fast, increase the Fib number, and if the calculation is taking too long, decrease the Fib number.

The exercise will return the following message:

```
[TIME]
```

Exercise 02*

In this exercise the student is going to run the Python script inside a container and change its CPU usage. The allowed CPU usage can be changed using one of the following commands:

- `--cpus=<value>`
i.e. `--cpus="1.5"` make sure the container uses 1.5 of the CPUs at max.

Your tasks:

1. Create a Dockerfile which runs the python script from the previous exercise.
2. Create a Docker image from the Dockerfile.
3. Spin up containers from the image and use different options for CPU.
Notice the difference in time it takes for the program to do its calculations.
4. Talk with the other members in your group and explain what is happening?

Hint: Information can be found [here](#)

The exercise will return the following message:

[TIME]

Exercise 03*

In this exercise the student is going to run the Python script inside a container and change its memory usage. The allowed memory usage can be changed using one of the following commands:

- `-m` or `--memory=<value>`
i.e. `-m="200m"` sets the container's maximum memory to 200 MB.
Hint: if the container doesn't print any longer it might have run out of memory.
- `--memory-swap`
i.e. `--memory-swap="1g" -m="200m"` sets the amount of memory the container can use before it needs to swap to 200 MB and enables it to swap with 800 MB.
Important: When using memory swap, the parameter memory (`-m`) has to be set.

Below is a code snippet off a python scripts which uses a lot of memory:

```
# Method which uses a lot of space
def use_space(loops):
    for i in range(0, loops):
        arr = bytearray(512000000)

# Program
import time
start_time = int(round(time.time() * 1000))
use_space(2)
end_time = int(round(time.time() * 1000))
print('time: ' + str(end_time - start_time))
```

Your tasks:

1. Create a python file similar to the one in the code snippet above.
2. Create a Dockerfile which runs the python script.
3. Create a Docker image from the Dockerfile.
4. Spin up containers from the image and use different options for memory.
Notice the difference in time it takes for the program to do its calculations.
5. Talk with the other members in your group and explain what is happening?

Hint: Information can be found [here](#)

The exercise will return the following message:

```
time: [TIME]
```