Lab 08: Multitasking

The exercises marked with * are obligatory to complete before coming weeks lecture.

The purpose of this lab is to get a better understanding of multitasking on a pc. You will get this information by compiling and running different C files containing code regarding multitasking.

Remember to write useful commands in your cheat sheets.

Download the resources for this lab through Blackboard.

Table of content

Exercise 01*

In this exercise the student is going to compile C source code in a docker container. To compile C code in a container, one must use the make command which is used for directing compilation. As well as a C compiler to actually compile the code.

The package “build-essential” includes both make-guile and gcc (GNU Compiler Collection). The first one enabling the make command, and the second one being the compiler.

Your tasks:

- Make a Dockerfile that:
  - Includes the provided C files
  - Installs the required packages
  - Compiles the provided C files
- Access the container and run the compiled scripts.

The exercise will return the following message:

```
** TIME USED **
```

Exercise 02*

In this exercise the student is going to run code1 and examine the output, as well as look into the code itself.

Your tasks:

- Run code1 and examine the output
- What is happening?
- What does the pthread_create do? What are the arguments?

The exercise will return the following message:

```
B563
B564
A548
A566
A567
```

Exercise 03*

In this exercise the student is going to run code2 and examine the output, as well as look into the code itself.

Your tasks:

- Run code2 and examine the output
- What is different from before?
- What is a mutex and how does it work?

The exercise will return the following message:

```
A499
A500
B501
B502
B503
B504
```

Exercise 04*

In this exercise the student is going to run code3 and examine the output, as well as look into the code itself.

Your tasks:

- Run code3 and examine the output
- What is happening?
- Why does the method count to 1000 twice?

The exercise will return the following message:

```
A998
A999
A1000
B1
B2
B3
```

Exercise 05*

In this exercise the student is going to run code4 and examine the output, as well as look into the code itself.

Your tasks:

- Run code4 and examine the output
- What is happening?
- Look at the code, why is it happening? Can you fix it?

The exercise will return the following message:

```
A1
```

Exercise 06*

In this exercise the student is going to run code5 and examine what is happening with htop.

Your tasks:

- Install htop in your container with the C code
- Run code5 in the container, and examine the cores with htop
- Edit the code, to make the threads run on the same core
- Examine what is happening with htop
- Try killing the process

The exercise will return the following message: