NWI-IBC019: Operating systems / Exercises: week 1 / 2022 v0

NOTE *These weekly exercises are individual (unless marked otherwise).*

READING **10TH EDITION** *chapter 1 up to and incl. 1.6; chapter 2 up to and incl. 2.7.*

HAND IN *Please hand in the exercises in PDF format in Brightspace.*

NOTE *The first assignment (about forking) will be published in week 2.*

Exercise 1

What is the purpose of interrupts? How does an interrupt differ from a trap? Can traps be generated intentionally by a user program? If so, for what purpose?

Exercise 2

Some computer systems (e.g. microcontrollers) do not provide a privileged mode of operation in hardware. Is it possible to construct a secure operating system for these computer systems? Give arguments both that it is and that it is not possible.

Exercise 3

Describe a mechanism for enforcing memory protection in order to prevent a program from modifying the memory associated with other programs.

Exercise 4

Describe three general methods for passing parameters to the operating system.

Exercise 5

What are the advantages and disadvantages of using the same system-call interface for manipulating both files and devices?

Exercise 6

Read the instructions how to set up a C++ development environment (using the Visual Studio Code editor), and download the project files at `https://gitlab.science.ru.nl/operatingsystems/assignment1`. Try to get the project working. Run it. It should give you a prompt, and you can enter a single command which it will execute. Afterwards your shell will terminate (and probably return to the default shell).

What does this code print on the screen after running a command like `pwd` (followed by a enter)? If an unknown command like `foobar` is typed, what does the shell do?

Can you explain what happened, and the difference in behaviour between the two? I.e. why does the shell not continue with a new prompt after a command was found?

Can you pinpoint the source code line that makes the difference? What kind of function is on that line?