

Lecturers:	Prof. Dr. Florina M. Ciorba Dr. Osman Simsek	<code>florina.ciorba@unibas.ch</code> <code>osman.simsek@unibas.ch</code>
Assistants:	Thomas Jakobsche	<code>thomas.jakobsche@unibas.ch</code>
Tutors:	Reto Krummenacher Agni Ramadani Tom Rodewald	<code>reto.krummenacher@unibas.ch</code> <code>agni.ramadani@unibas.ch</code> <code>tom.rodewald@unibas.ch</code>

Exercise 4: Memory Management, File Systems (10 points)

Given: April 19, 2024

Deadline: May 14, 2024, 10:00

Objectives

- Understand memory allocation, initialization, access and deallocation
- Understand basic operations (open, read, write, and close) to deal with files
- Understand the basics of containerization

Tasks

- Task 1: Memory Management (4 points)
- Task 2: Virtual vs. Physical Memory (1 point)
- Task 3: File System (2 points)
- Task 4: Containerization (3 points)

Instructions

- You can solve this exercises in teams of two.
- Submit the solution of each task with detailed comments that clarify your solution.
- Show your solution and upload it to <https://adam.unibas.ch> with all deliverables in a ZIP folder with the naming scheme: *[GroupID]_Ex[SheetNo]_LastName1_LastName2*.
- In total, at least 65% of exercise points have to be obtained (with a min of 30% of each exercise).

Task 1: Memory Management**(4 points)**

You are given three source files, namely T1-1.c, T1-2.c, T1-3.c, your task is to understand and compile each of them. Each file has a certain issue that you must identify, report, and fix.

- T1-1.c is a simple program that asks users to enter a number representing the total number of random samples generated by the program. All generated samples are between 0 and 9. The program displays a histogram (number of repetitions) of each value in that range. Users reported that the issue appears when they pass a large number to the program, such as 10^7 . Your task is to explain why this issue happens, and you must fix it.
- T1-2.c is a malfunctioning program. The `update` function has a bug, i.e., the `update` function takes two arguments: both are integer arrays. All items within the first array are initialized to -1. The second array contains indices that must be used to update their corresponding items of the first array if possible. Once you execute the code, it will give a segmentation fault. Your task is to explain why this happens and fix the `update` function.
- T1-3.c is a program that sorts a sequence of positive integers (using `count sort`). The program is working correctly and you may verify that by compiling and executing the program. However, users reported that sometimes when the `count_sort` function is called repeatedly in a loop, the program crashes unexpectedly. Your task is to read the code, identify the cause of this issue, and fix it.
Hint: you can use external tools such as Valgrind.

Task 2: Virtual vs. Physical Memory**(1 point)**

Explain the difference between virtual and physical memory for this exercise.

Task 3: File System**(2 points)**

Given the source file T2.c, you must implement the TODOs in this file. You have two functions to implement: `readLinesCount` and `writeLinesCount`. Once you see the `main` function, you will realize that the program accepts multiple arguments: The first argument represents the number of its following arguments. Each of these arguments is a path to a file. The program opens each file and counts its lines; it then writes each line count to an output file, called `output.txt`.

- implements `readLinesCount` that takes a path to a certain file and returns the count of the lines within the file.
- implements `writeLinesCount` that takes a path to an output file and an integer. The function then appends the integer to the given file as a newline.

Task 4: Containerization

(3 points)

Create and implement a Dockerfile in order to create a custom Docker Image that will run a simple Hello World code using Python. Follow the steps described below:

- First, create a Python code that prints "Hello World from Python".
- Create and implement a Dockerfile that will setup a Python base image, set a working directory of the container, copy your application code into the container, and setup the command to run your application.
- Build your Docker Image.
- Run the container which has your application.

Submit your application code, your Dockerfile, and the terminal output for all the steps. Document and explain your implementation.