Humboldt-Universität zu Berlin
Institut für Informatik
Prof. Dr. Ulf Leser
Arik Ermshaus

Berlin, the 24.10.2023

Tutorials for the lecture
Implementation of Databases (DBS2)

Exercise sheet 1

Submission: By Monday, on 13.11.2023, until 23:59 o'clock via Moodle. The exercise sheets are to be completed in groups of three (in exceptional cases two) students. Unless otherwise specified, the solutions must be submitted on separate **PDFs** via Moodle. All subtasks of a task must be uploaded in one PDF. It is sufficient for one person per group to submit the group's solution. For evaluation, the last uploaded version will be considered. Please include your **names**, your **CMS usernames**, and your **submission group (e.g., Gruppe 123)** from Moodle on all submissions. Name the uploaded PDF files according to the scheme: A<Task>-<Person1>-<Person2>-<Person3>.pdf, for example, A3-Musterfrau-Mustermann-Beispiel.pdf for Task 3 by Lisa Musterfrau, Peter Mustermann, and Karla Beispiel. The listing of names may be in any order. Please refer to the information in the Moodle course https://hu.berlin/dbs223.

Task 1 (Hard drives) — 2 + 2 + 2 + 3 = 9 Points

In this task you should answer questions about properties of an example hard drive. Consider a magnetic disk that consists of 5 platters, having a diameter of 9 cm. Each platter has two surfaces, each containing 10.000 tracks. A given track contains 1.000 sectors for simplicity, each having a capacity of 1024 byte, and a block comprises 5 sectors. Moving the head over n tracks takes $(1 + 0,001 \cdot n)$ ms. The rotation speed is given by 5.000 revolutions per minute. The probability that a disk crashes on a given day is $\frac{1}{10.000}$.

(a) Calculate the entire storage capacity of the magnetic disk in gigabytes (GB).

(b) Determine the number of blocks contained in a single cylinder of the disk.

(c) Calculate the average time needed to sequentially read the contents of an entire track. *Hint:* The head first has to move to the correct spot to read the track.

(d) How many magnetic disks can you operate at most simultaneously, while keeping the probability of at least one disk crashing per day below 1%?

Provide your calculation paths for all subtasks (a) to (d).

Task 2 (RAID) **2 + 2 + 2 = 6 Points**

You are tasked with configuring a RAID (Redundant Array of Independent Disks) system that consists of 5 individual magnetic disks, each with a storage capacity of 1 TB. The configuration must adhere to the following criteria: The system should tolerate the failure of a single disk without losing any data. Among configurations that meet the first criterion, choose the one that maximizes the total storage capacity. If multiple configurations offer equal storage capacity, prioritize the one that maximizes read/write performance.

(a) Which RAID level would be the most appropriate for this setup? Justify your choice.

(b) Calculate the total effective storage capacity of the RAID system, based on your chosen configuration.

(c) Assuming each disk can read or write at a rate of 100 blocks per second, determine the maximum number of blocks that can be read from the entire RAID system per second.

Provide your calculation paths for subtasks (b) and (c).

Task 3 (External sorting) **15 Points**

In Moodle, you find a file named "large_file.txt". This file contains 50 million unique integers, each listed on a separate line. The total file size is approximately 439 MB. Your assignment is to sort the integers in this file in ascending order while adhering to a memory usage constraint of 50 MB.

To complete this task, implement the function "sort_external_file" in the provided C++ template "task3.cpp", also available in Moodle. This template contains tests to verify the correctness of your code. Do not modify the function signatures already defined in the template. You are, however, free to implement additional helper functions.

Your code should rely solely on the C++ standard libraries; usage of third-party libraries is not permitted. Also, ensure that your code is executable on the gruenau2-6 system using the provided "CMakeLists.txt" file.