CSE 5031 Operating Systems
2020/21 Fall Term

Project: 2 – Part 3
Topic: ISAM update using Low-level I/O API
Date: 09 - 15.11.2020

Objectives:

- to implement an application to update an Indexed File
- to use low-level I/O **GNU C Library API**

References:

- **The GNU C Library Reference Manual** (http://www.gnu.org/software/libc/manual/pdf/libc.pdf)
- **Linux System Programming 2d ed., Robert Love, O'Reilly 2013** (course web site, or http://pdf-ebooks-for-free.blogspot.com.tr/2015/01/oreilly-linux-system-programming.html)

Section A. Project Definition

A.1 Project Coverage

In the previous project (project 2 Part 2) you were asked to develop a **query program** in **C** that queries the records of an **unsorted** data file (a copy of /etc/passwd) using **ISAM direct access** mechanism.

In part 3 you are asked to develop an **update program** in **C** that will:

- ✓ retrieve (read) a given account record from "***passwd***";
- ✓ change the contents of one of its fields in the memory buffer; and
- ✓ write the record back (updates it on disk) using **ISAM direct access** mechanism.

A.2 Creating the Index File

i) Follow the procedure defined in **Section B.4** of **Project 2 Part 2**, to create the **Index** file "***passwd.idx***" for the local copy of the ***"/etc/passwd***" file.

ii) Verify that"***passwd.idx***" has been created correctly by running the **ISAM Query program** you have developed in Project 2 Part 2.
In case you have not implemented this program, you may display the records of both text files and perform necessary visual checks.

Section B. Developing ISAM Update Program

B.1 Update Program Requirements

i) The **update** program is expected to change the contents of one specific field in the account record, the comment field (**5th field**). The update should be performed only if:

- ✓ the size comment field of the targeted record is **> 0**; and
- ✓ the size of the update data **==** the size record's comment field.

✓ You should be prepared to answer the questions:

- → "*what needs to be done if the update data length does not match the above criteria and we want still to register this change in the data file*" ?
- → "can we update record key field, and how"?

ii) Your **C** program should be implemented using the **pread** and **pwrite low-level I/O primitives** to perform direct **read** and **write** operations that are documented in:

- ✓ ***The GNU C Library Reference Manual*** section "3.2 Input and Output Primitives", and
- ✓ ***Linux System Programming 2d ed***, "Chapter 2 File I/O – Positional Reads Writes" section.

You are **strongly advised** to refer to theses programming resources, instead of wasting your valuable time in "**fishing junk**" over the Internet.

ii) The **update** program should load the file index "***passwd.idx***" in a dynamically allocated **Index Table**.
Note that you may derive the size of the **Index Table** from the size of the **Index File** using the "**stat**" function.

B.2 Update Program Algorithm

Write a **C** program that **updates** the local copy of the ***"/etc/passwd***" file implementing the following steps:

- ✓ read from **standard input** an account name, until an **end of file** is entered (ctrl+del keystrokes for the VM);
- ✓ retrieve the account record corresponding to the key from the ***"passwd"*** file;
- ✓ if the record is found:
  - + display the record, and the size of the comment field;
  - + if the size of record's comment field == 0 proceed with the next query;
    otherwise,
    - read from **stdin** the update text for the **comment** field;
    - if update the record if it meets the length constraints.

Section C. Project Report

C.1 Project Report

Write a brief that explains:

- ✓ The benefits of using the **pread** and **pwrite low-level I/O primitives** rather than **read** and **write** operations.
- ✓ What is to be changed in the update program if low level than **read** and **write** operations were used.
- ✓ Can we update or not the record key field, and how?

C.2 Project Report

If your update program **operates correctly,**

- ✓ add at the beginning of the C code a comment line consisting of your name and student-id**;**
- ✓ store update program **source code** and the brief file in the "**Prj2-Part3**" folder, located at the course web site under the tab **CSE5031 - OS Section -X/Assignment**; where "**X**" stands for (1,2,3,4) the laboratory session group you are registered in.

Warning

You are encouraged to discuss the implementation procedures and general concepts behind the projects with your fellow students. However, **plagiarism is strictly forbidden**! Submitted report should be the result of **your personal work**!

Be advised that you are **accountable** of your submission not only for this project, but also for the mid-term, and final examinations. Your project grade may be reevaluated retrospectively, had you fail to answer correctly the same or a similar examination questions that you have solved with success in your submissions.