MYY502 – SYSTEMS PROGRAMMING

Lab #6
24/11/2015

I. FILES

In file `data.txt` lies a set of integers. The first one denotes the total number of the rest integers. If for example the first integer equals to 4 then 4 more integers follow in the file.

a) Write a program that opens this file, reads the data (integers) and stores them to an array of appropriate size, created dynamically using `malloc()` function. Next the program will separate the data into two different files: file `positive.txt` and file `negative.txt`. It will store all the positive numbers in file `positive.txt` and all negative numbers in `negative.txt`. Both `positive.txt` and `negative.txt` must retain the format of the original file, that is the first number must denote the total number of integers that follow in the file.

b) Add some code that restores the file pointer of `positive.txt` in the beginning of the file, re-reads the numbers of `positive.txt` and calculates their sum.

II. STRUCTS

Write a program that stores the grades of a class for high-school students. For every student we must keep data regarding the name, the registration number, and the grade for the particular class. So you are asked to:

1. Read the number of students as well as their data from standard input.
2. Store the input data using a (`struct`) for each student.
3. Calculate the grade average, using only the structs from step 2.
4. Print the names of the students that have failed the class (grade less than 10). Again using only the structs from step 2.
5. Change the names of failed students adding the word «(failed)». e.g. «Vasilios Vasiliou» must be altered to «Vasilios Vasiliou (failed)».

Implement the above steps using an **array of structs**. You must declare the struct, then dynamically allocate the array of the structs and, finally, store the data of each student in that array. You should implement the functionality following the skeleton code given in file `skel1.c`.

III. STRUCTS + FILES

Repeat the previous exercise II, but this time the students data must be kept in file. More specifically:

- When the program exits it stores the array of structs in a file `students.txt` (store each field of the struct individually in separate lines using fprintf()).*Remember to first store the number of the students.*
- During the next run of the program you the program must check whether a file named `students.txt` exists. If so, the program must read it and initialize the struct array data from the file's contents. (Remember that the total number of students is stored at the beginning of the file. If no file exists then the data will be read from standard input, same as exercise II.

IV. ADDITIONAL EXERCISE: STRUCTS + DYNAMIC LIST 1

Implement the exercise II using a **list of structs** instead of an array. You must declare a struct and include a field called `next`, that will point to the next node of the list. You must also declare a `head` variable (which is a pointer) with an initial value of NULL, denoting an empty list. Then this variable will point to the last entered node of the list. Each list node must be allocated dynamically and must be entered where the `head` points. You should implement the functionality following the skeleton code given in file `skel2.c`.

V. ADDITIONAL EXERCISE: STRUCTS + DYNAMIC LIST 2

Re-implement the exercise IV, but this time each list node *must be entered at the end of list instead of the beginning*.