

## VE482 — Introduction to Operating Systems

### Homework 3

Manuel — UM-JI (Fall 2019)

Non-programming exercises:

- Write in a neat and legible handwriting
- Clearly explain the reasoning process
- Write in a complete style (subject, verb and object)

Programming exercises:

- Write a README file for each program
- Upload an archive with all the programs onto Canvas

#### Ex. 1 — *Research on POSIX*

Write a few paragraphs about the POSIX standards. What are they, why do they exist, what kind of things are included in the norms.

#### Ex. 2 — *General questions*

1. Why would a thread voluntarily release the CPU?
2. What is the biggest advantage/disadvantage of user space threads?
3. If a multithreaded process forks, a problem occurs if the child gets copies of all the parent's threads. Suppose that one of the original threads was waiting for keyboard input. Now two threads are waiting for keyboard input, one in each process. Does this problem ever occur in single-threaded processes?
4. Many UNIX system calls have no Win32 API equivalents. For each such call, what are the consequences when porting a program from a UNIX system to a Windows system?

#### Ex. 3 — *C programming*

The goal of this exercise is to improve the programming skills and get more familiar with pointers and function pointers.

1. Implement a linked list structure containing two pointers of type `char` and `void`, respectively. It should be possible to at least add elements to the list.
2. Knowing that the `void` pointer in the structure could contain some `char*`, `int`, or `double`, write a search function for this linked list.
3. The linked list will store elements read from an ASCII file where each line is in the format `somestring=somedata`. The type of the data is defined in the filename; for instance a file containing unsorted integers will be named `rand_int.txt`. Implement the necessary functions to read and write such files.
4. Use a function pointers to compare and sort the elements from the structure with respect to the data field. Implement the following sorting orders: increasing, decreasing, and random. The filename is `sortingtype_datatype.txt`, where `sortingtype` is `rand`, `inc`, or `dec`.
5. Write a function to test the implementation.

Sample output:

```
$ cat rand_int.txt  
acb=148  
bac=234  
ghi=9
```

```
$ ./ex3 rand_int.txt inc  
reading rand_int.txt  
sorting elements  
writing inc_int.txt
```

```
$ cat inc_int.txt  
ghi=9  
acb=148  
bac=234
```