Operating Systems Programming Lab
Computer Engineering Department
Fall 2023/2024
Lab 2: UNIX I/O

Objectives

1. To understand the “forking” process and the “fork” system call.
2. To understand the “exec” system call and its variants from the unistd library.
3. To understand the IO operations in UNIX-based systems.

Prelab

1. Review chapter 4 of the textbook.
2. Read the manual pages of the following systems calls and functions.

```
ssize_t read(int fd, void* buf, size_t count);
ssize_t write(int fd, const void* buf, size_t count);
int open(const char* pathname, int flags);
int open(const char* pathname, int flags, mode_t mode);
int close(int fd);
pid_t fork();
int execvp(const char* file, char** argv);
pid_t wait(int* status);
```

Experiment

1. In class, we implemented two programs:

   - makeargv.c that converts a string to an array of strings (char**) with an extra NULL pointer at the end.
   - readline.c that reads an input file line by line.

   Use these two programs to implement an applications that takes a filename as an argument. The file contains a list of shell commands, one per line like this:

```
ls -l
pwd
touch f.txt
rm f.txt
```

   The application must read the file line by line. With every line, the parent process creates a child process to execute the command specified by that line. The parent waits for the child to finish. This process repeats until all commands in the file are executed.