**System programming**

**Lab 2**

**Buffered I/O**

**Task #1**

Using the *time* built-in command of the shell, try timing the operation of the program *cp*   
(from Lab 1) on your system. Experiment with different file and buffer sizes. Create a table of your results. Analyze.

**Task #2**

The command *tail [ –n num ] file* prints the last *num* lines (ten by default) of the named file. Implement this command using I/O system calls (*lseek(),* *read(), write(),* and so on). Keep in mind the buffering issues described in the lecture 2, in order to make the implementation efficient.

**Task #3**

Look at the following pseudocode that is used for copying data from standard input into standard output using functions *fgets* and *fputs*.

//include necessary header files

int main(void)

{

char buf[MAXLINE];

while (fgets(buf, MAXLINE, stdin) != NULL)

if (fputs(buf, stdout) == EOF)

// print about output error

if (ferror(stdin))

// print about input error

exit(0);

}

Instead of MAXLINE use the value 4. What happens if you try to copy a file in which the lengths of the lines exceeds 4 bytes? Explain.