# New York University
# Tandon School of Engineering
### Department of Computer Science and Engineering

## Introduction to Operating Systems
Spring 2025

## Assignment 2
(10 points)

a) (0 points) The `strace` utility allows us to trace the system calls a program makes. Use the man pages to read the documentation for the "`strace`" command, e.g. :

    `man strace`

Alternatively, you may google it by typing "man strace" in your browser's search window, which should send you to "man7.org". Find out the option that allows you to count the number of occurrences of each system call.

b) (10 points) Develop a C program, "mycat", whose main routine accepts an input parameter from the user and an input file name (passed to your program when it was invoked from the shell). The file is a text file.

Your program shall then print the following to the screen:

- In the first line, print the process ID of your (running) program, an integer (Hint: use function `getpid()`, you can man it of course)
- In the second and subsequent lines, print the contents of the input file. Use Unix calls for opening, closing and reading files, not the "standard C" library calls (i.e. use `open` instead of `fopen`, etc.).
- Your program shall sleep/wait for a random number of seconds (1 to 5) between printing the process ID and printing the content of the input file.

Below is an example of how your program should be invoked from the shell:

    `mycat input.txt`

Where input.txt is a file that already exists (you may create an input file with a few lines to test your code). Note that if a path is not provided in filenames, then it's assumed that a file is located at the same directory as the working directory of your program, i.e. the directory where your program was invoked from.

After developing your program, invoke using `strace` and then answer the following questions:

1) What are the system call names for getting the process ID, opening a file, closing a file, reading a file, printing to the console and sleeping?
2) What are the number of system calls for opening, closing and reading the file(s) (i.e. how many times each was called).
3) What are the number of system calls for printing to the screen? (count each individually. You may either use strace options to aid you in doing so, or you may use grep).
4) What was the value of the file descriptor of your read file (please review the lecture slides before asking what this means)?

**<u>Notes and hints:</u>**

- Please include your answers to questions and the strace log in your submitted .pdf file.
- Create a text file and use it to test your program, e.g. type:
  ```
  touch input.txt
  echo "Hello world" > input.txt
  echo "This is lab 2" >> input.txt
  ```
- Use the man pages to learn how to use POSIX API library functions (and the necessary include files) and/or UNIX commands and its various optional arguments (e.g. strace, especially for counting), e.g.:
  ```
  man strace              // gets info from section 1, user's manual
  man getpid.2            // section 2 is programmer's manual
  ```

# <u>What to submit to</u> <span style="color:blue">gradescope</span>:

Please submit the following files individually:

1) Source file(s) with appropriate comments.
   The naming should be similar to "**lab#_$.c**" (**#** is replaced with the assignment number and **$** with the question number within the assignment, e.g. lab4_b.c, for lab 4, question b OR lab5_1a for lab 5, question 1a).
2) A single pdf file (for images + report/answers to questions), named **"lab#.pdf"** (**#** is replaced by the assignment number), containing:
   - Screen shot(**s**) of your terminal window showing the current directory, the command used to compile your program, the command used to run your program and the output of your program.
3) Your Makefile, if any. This is applicable only to kernel modules.

# <u>RULES:</u>

- You shall **use kernel version 4.x.x or above**. You shall not use kernel version 3.x.x.
- You may consult with other students about GENERAL concepts or methods but copying code (or code fragments) or algorithms is NOT ALLOWED and is considered cheating (whether copied form other students, the internet or any other source).
- If you are having trouble, please ask your teaching assistant for help.
- You must submit your assignment prior to the deadline.