# New York University
# Tandon School of Engineering
Department of Electrical & Computer Engineering

Introduction to Operating Systems (CS-GY6233)
Spring 2022

Assignment 6
(10 points)

A. (8 points) Repeat assignment 5 part 2, except that you shall now use a TCP/IP socket for communicating between the processes instead of a pipe.

Use the following socket functions in their default mode. You may use the `man` command in your Linux virtual machine for information about the parameters:

| **CLIENT** | **SERVER** |
|---|---|
| socket() – opens a socket (similar to pipe() ) | socket() |
| connect() – connects to a server | bind() – assigns a particular port number to the server<br>listen() – listens to connection requests from clients<br>accept() – accepts a connection from client |
| read() – reads a buffer from the socket, just as in file or pipe reading | write() – writes a buffer to the socket, just as in file or pipe writing |
| close() – closes the socket | close() |

You shall use sockets of type SOCK_STREAM and assign the parent (consumer) as the client and the child (producer) as the server.

Insert an initial random wait (1 to 5 seconds) at the child process (but not the parent) prior to it starting to listen and accept connections.

The parent process (client) may thus fail to connect if it tries to do so before the child process (server) has started to listen (which is after the random wait). As such, you should insert a loop in the parent that repeatedly attempts to connect, waiting 100 ms between attempts, till it succeeds eventually.

B. (2 points): Answer the following:
   a. Which of the calls above are blocking and which are not? Explain what that means?
   b. Is this a form of direct communications or indirect communications?
   c. What is the failure flag returned from connect() that indicates the server is not ready?
   d. How would you change your program to communicate between processes in a different machine?

## Submission file structure:

Please submit a **single .zip file** named **[Your Netid]_lab#.zip**. It shall have the following structure (replace # with the actual assignment number):

└── [Your Netid] hw# (Single folder includes all your submissions)

    ├── lab#_1.c (Source code for problem 1)

├── lab#_2a.c (Source code for problem 2a, and so on)

├── lab#_1.h (Source code header file, if any)

├── Makefile  (makefile used to build your program, if any)

├── lab#.pdf (images + Report/answers to short-answer questions)

## What to hand in (using Brightspace):

- Source files (.c or .h) with appropriate comments.
- Your Makefile if any.
- A .pdf file 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.

## RULES:

- 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.