

Laboratory #5 (PIPE, FIFO, SIGNAL)– May 31th 2021

Exercise 1 (PIPE)- Write a multi-process program that evaluates the following math series:

$$\sum_{i=1}^{N-1} \frac{e^i}{i}$$

The main process receives the N value as input (set max value for N to 10) and performs the final sum. The i -th process evaluates its internal term and sends the result to the father **through a pipe**. Please notice that each internal term generates a floating value.

Hint: please notice that the read function is, by default, a blocking function: if there are no (enough) data to be read, the reading process is moved to the waiting state. In this configuration, there is no 0 byte read return to check: a 0 / less than 0 value means error during the read operation!

Exercise 2 (FIFO)– Write a modified Linux shell based on a client/server architecture where the client shares with the server a sequence of strings, using a **FIFO**. The server scans the content of every string and, as soon as the string is equal to “**house**” or “**casa**”, it prints the message “**house detected**”. If the string contains the string “**exit**” both client and server must terminate.

Hint: remember that the FIFO can be defined and shared among processes by referring to the same fifo name, thus you do not need to fork the process but you must write two different programs: the server and the client. Run the server before the client.

Exercise 3 (SIGNAL) – Write a program that creates 10 child processes. The creating process has to:

- Display their **ID** and prompt the user to choose the process that has to end.
- Signal the end of the requested process through the **SIGQUIT** signal.
- Re-print the list of remaining open process **IDs** and prompts the user to choose a process to end.

Each child process has to:

- Manifest itself every X random seconds (between 1 and 10) by printing some messages including its **ID**.
- The main program ends when all processes terminate.