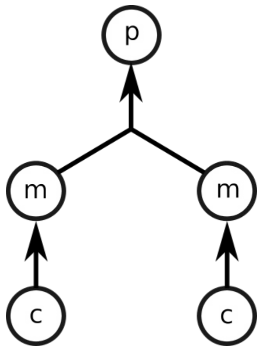
Operating Systems 2 - Laboratory 1

FIFO and pipes



Write a multiprocess program called **`pipefork`** which handles communication between processes via named pipes. The program takes three positional arguments: `t` in range `[100,1000]`, `n` in range `[1,10]`, `r` in range `[0,100]`, `a` in range `[1,PIPE_BUF]` and `b` in range `[a,PIPE_BUF]` – `b` is always greater or equal to `a`.

The main process p creates a single FIFO and two child processes m as shown on the diagram. Each m process creates a single FIFO and it's child process c. You may choose the FIFO names as you wish.

Each c process is sending buffers filled with random number chars (`'0' - '9'`) via FIFO. Each write size is randomized between a and b bytes. After each write process waits for `t` milliseconds.

Each m process reads data via pipe from process c. With `r%` probability it doubles the random buffer size by appending 'X' characters at the end of the buffer. Then it sends the buffer via next pipe to the main process p.

The p process reads the data from the single pipe and displays each read buffer on stdout in format: [n]: [size]: [text] where text is the received character buffer, n is the incremented read call number and size is the length of the text.

After `n` writers process c processes terminate. Also, when user sends SIGINT to the program processes c should close their pipes and exit immediately. Other processes should ignore this signal. In both cases, as a result, all processes detect closed pipes and the program terminates.

STAGES (TOTAL OF 14 POINTS)

	Stage	**Points**	**Requirements**
Laboratory part 105minutes	1	2	Parsing **all** arguments and creating the child processes (all 4). All processes wait for children (if any). Processes c sleep for 1s before exiting. All processes print messages containing PID at start and at exit.
	2	4	Creating FIFOs. Processes c send 1 byte to parent before exiting. Processes m forward this byte to parent and also exit. Reading processes exit when the other end of the pipe is closed. Before exiting FIFOs are removed from the filesystem.
	3	1	Processes c send n separate bytes sleeping t ms inbetween.
Homework	4	2	Processes c generate a radom character buffer for each write call. The main process prints this buffer in format given in the task description (handling of a and b parameters).
	5	2	Random extension of the buffer size in the m processes (handling of r parameter).
	6	3	Clean interruption of the program with C-c.

UPLOAD

Please upload your solution to: **`/home2/samba/sobotkap/unix/`**

You have to upload each stage immediately after finishing it. You upload to a network share via ssh.mini.pw.edu.pl server:

```
scp user.etapX.tar.bz2 user@ssh.mini.pw.edu.pl:/home2/samba/sobotkap/unix/
```

Please name your stages files according to the schema: `LOGIN.etapN.tar.bz2(.gz)`

THE STATEMENT

By decree 27/2020 of University Rector you must add the following statement to the uploads:

```
-----------------------------------------------------------------------
I declare that this piece of work which is the basis for recognition of
achieving learning outcomes in the OPS course was completed on my own.
[First and last name] [Student record book number (Student ID number)]
-----------------------------------------------------------------------
```

Please add it as comment at the beginning of each source file you upload.
Replace square brackets with your data.