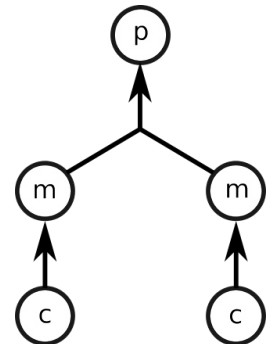


Operating Systems 2 - Laboratory 1

FIFO and pipes

Write a multiprocess program called **pipefork** which handles communication between processes via unnamed pipes. The program takes three positional arguments: *t* in range [50,500], *n* in range [3,30] and *r* in range [0,100] and *b* in range [1,PIPE_BUF-6].

The main process *p* creates a single pipe and two child processes *m* as shown on the diagram. Each *m* process creates a single pipe and it's child process *c*.



Each *c* process is sending random character data ('a' - 'z') via pipe. Each write size is randomized between *b* and (PIPE_BUF - 6) bytes. After each write process waits for *t* milliseconds.

Each *m* process reads data via pipe from process *c*. With *r*% probability it appends 8 bytes of additional data: "injected" at the end of the packet. 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 arguments and creating the child processes (all 4). All processes wait for children (if any). Processes <i>c</i> sleep for 1 second before exiting. All processes print messages containing PID at start and at exit.
	2	4	Creating pipes and closing unused descriptors. Processes <i>c</i> send 1 byte to parent before exiting. Processes <i>m</i> forward this byte to parent and also exit. Reading processes exit when the other end of the pipe is closed.
	3	1	Processes <i>c</i> send <i>n</i> separate bytes sleeping <i>t</i> ms inbetween.
Homework 18.03.2021	4	2	Processes <i>c</i> generate a radom character buffer for each write call. The main process prints this buffer in format given in the task description.
	5	2	Random injection of "injected" test in the <i>m</i> processes.
	6	3	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.