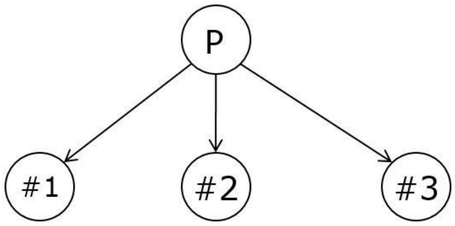
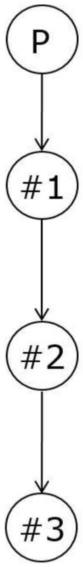
Laboratory #2 – fork(), sleep(), exit(), wait(), waitpid() – March 16th 2021

Exercise 1: write a C program that creates a child process using the fork() system call. The child process must print its PID, whereas the parent process must print both its PID and the PID of the child.

Exercise 2: write a C program that creates 3 processes as the following model.



Exercise 3: write a C program that creates 3 processes as the following model.



Exercise 4. Write a *C program* that creates 2 child processes with the following characteristics:

- the child processes wait for 5 seconds and terminate
- the parent process terminates when the second created child terminates.

Exercise 5. Write a *C program* that creates 6 child processes with the following characteristics:

- all 6 child processes are always running
- first 3 processes continuously generate a random number and print it on the standard output
- second 3 processes generate 50 random numbers and print it and its square on the standard output
- every time a child process terminates it should be replaced by a new child of the same type.

Exercise 6: Write a multi-process program that evaluates the following math series:

$$\sum_{i=0}^{n} 2^{\mathrm{i}}$$

The main process receives the n value as input (set max value for n to 5) and performs the final sum. Each 2i is evaluated by the i-th process and sent to the father for the final sum.

Hint: remember that you cannot share variables among processes right now but you can always send back some value by the exit status.