

# Лабораторная работа № U11

## Взаимодействие между процессами – именованные и неименованные каналы (однонаправленные)

Для однонаправленного обмена данными между процессами можно использовать механизм именованных каналов, при этом одна программа будет вычитывать из него данные, а другая записывать, однако при этом не будет происходить физической записи данных на диск.

- Создать новый именованный канал – *mkfifo имя\_канала*
- Записать/считать данные – обычные файловые операции или перенаправления

Неименованные каналы создаются с помощью системного вызова *pipe*, который возвращает массив из двух указателей – для каждой стороны именованного канала.

*int pipefd[2];* - указатель на «стороны» именованного канала, где элемент 0 – указатель для считывания, 1 – для записи.

*pipe(pipefd);* – создать неименованный канал, где *pipefd* – это массив из двух элементов типа *int*

*close(pipe\_ref)* – закрыть неименованный канал, где *pipe\_ref* – дескриптор закрываемого канала

*int read(int pipe\_ref, \*buf, int strlen)* – читает данные (из канала), где *\*buf* указатель на буфер, куда будут вычитываться данные, а *strlen* – это максимальный размер считываемых за 1 вызов данных.

Возвращаемое значение – количество реально прочитанных данных

*int write(int pipe\_ref, \*buf, int strlen)* – записывает данные в канал, возвращаемое значение – количество реально записанных данных

Задание:

### 1. Именованные каналы

- а. Создать именованный канал в домашнем каталоге
- б. С помощью *echo* записать в него какой – либо текст. (выяснить завершится ли *echo* сразу или будет «висеть»? )
  - и. Примечание: с помощью команды *stat имя\_канала* убедиться, что работаете с именованным каналом, а не обычным файлом
- в. Запустить ещё один «терминал» и с помощью *cat* вычитать содержимое именованного канала.
- г. Повторить п3 и далее п2. (Определить – завершится ли *echo* сразу или будет «висеть»?)
- д. Написать программу, которая будет вычитывать данные из именованного канала (с помощью системного вызова *read*) и выводить «на экран».
  - и. Примечание: логику вычитывания данных (*printf*, *open*, *read* и *close*) надо внутри вечного цикла делать
- е. Запустить программу и в другом «терминале» поместить текст в именованный канал.

### 2. Неименованные каналы: написать программу, где сначала создаётся канал, а затем создаётся новый процесс, при этом:

- процесс потомок будет вычитывать данные из канала;
- процесс-родитель записывать текст за 5 попыток, между которыми секунд 10-20 (для этого можно использовать *sleep(XXX)*, где *XXX* это количество секунд).

Прим:

- По завершении работы не забыть удалить «старый» именованный канал и ненужную уже программу
- Для компилирования можно использовать *gcc -o имя\_исполняемого\_модуля \_имя\_исходного\_файла.c*
- Для *pipe* нужна библиотека: *stdlib.h*, для *printf* и работы с файлами – *stdio.h*, для *sleep* – *unistd.h*