

## Семинар 5

### Организация взаимодействия процессов через `pipe` и `FIFO` в UNIX

(68 баллов)

1. Откомпилируйте программу 05-0.c (в учебнике отсутствует, есть в архиве) и запустите ее на исполнение. Обратите внимание на несоответствие прав доступа у созданного файла правам, указанным в параметрах системного вызова `open()`. Объясните причину расхождений.

(3 балла)

2. Откомпилируйте программу 05-1.c и запустите ее на исполнение. Обратите внимание на использование системного вызова `umask()` с параметром 0 для того, чтобы права доступа к созданному файлу точно соответствовали указанным в системном вызове `open()`.

(3 балла)

3. Измените программу 05-1.c из предыдущего раздела так, чтобы она читала записанную ранее в файл информацию и печатала ее на экране. Все лишние операторы желательно удалить, оставив нужные.

(5 баллов – за правильное решение,  
при наличии лишних операторов — -1 балл за каждый,  
при удалении нужного — -1 за каждое лишнее удаление).

4. Откомпилируйте программу 05-2.c (работа с **pipe** в рамках одного процесса), запустите на исполнение. Объясните результаты.

(5 баллов)

5. Откомпилируйте программу 05-3.c (однонаправленная связь между предком и потомком), запустите на исполнение. Объясните результаты.

(5 баллов)

6. **Задача повышенной сложности:** модифицируйте пример 05-3.c для связи между собой двух родственных процессов, выполняющих разные программы.

Первому, кто предложил и реализовал новый для группы вариант, – 15 баллов, повторение уже предложенной идеи – 8 баллов.

7. Модифицируйте программу 05-3.c для организации **двусторонней** связи, откомпилируйте ее и запустите на исполнение.

(7 баллов)

8. **Задача повышенной сложности:** определите размер `pipe` для вашей операционной системы.

(Оригинальная идея - 15 баллов).

(Повтор идеи и объяснение происходящего — 8 баллов).

9. Откомпилируйте программу 05-4.c, запустите её на исполнение. В этой программе информацией через *именованный программный канал* обмениваются процесс-родитель и процесс-ребенок.

(3 балла)

10. Напишите на базе предыдущего примера две **не** родственные программы, одна из которых пишет информацию в FIFO, а вторая – читает из него.

(7 баллов)