МИНОБРНАУКИ РОССИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

«Череповецкий государственный университет»

**Лабораторная работа № 6**

**«Организация взаимодействия процессов через pipe и FIFO в ОС UNIX»**

**Выполнил:**

студент гр. 1ИВТпб-01-31оп

Климов А.Г.  
**Проверил:** преподаватель

Майтама Е.В.  
Отметка о зачете:

Череповец

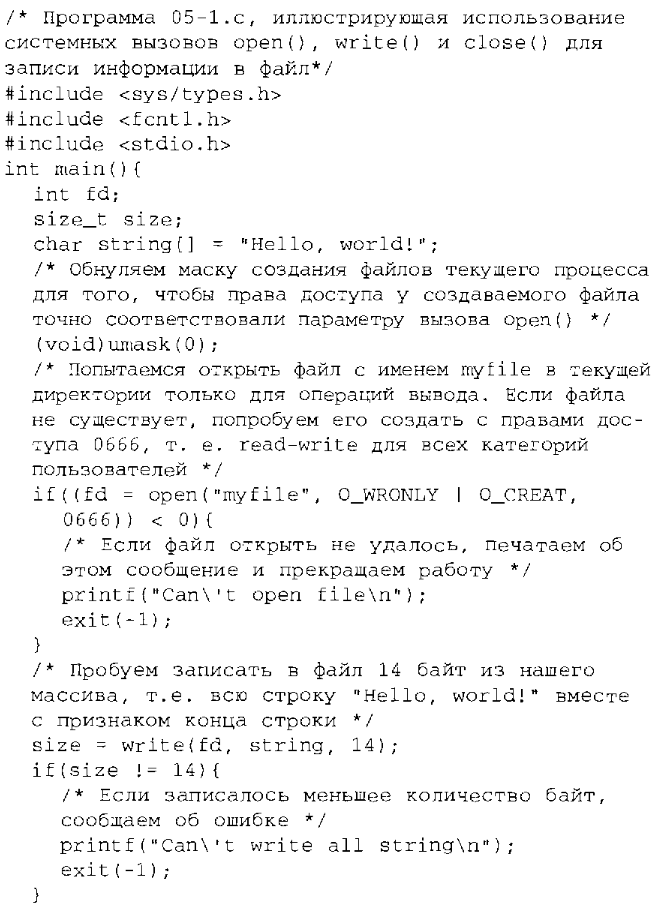
2018 год

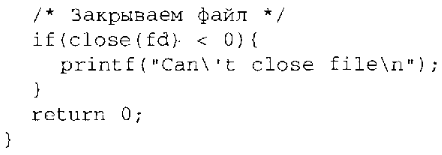
Наиболее простым способом для передачи информации с помощью потоковой модели между различными процессами или даже внутри одного процесса в операционной системе UNIX является **pipe** (канал, труба, конвейер).

Pipe можно представить себе в виде трубы ограниченной емкости, расположенной внутри адресного пространства операционной системы, доступ к входному и выходному отверстию которой осуществляется с помощью системных вызовов. В действительности pipe представляет собой область памяти, недоступную пользовательским процессам напрямую, зачастую организованную в виде кольцевого буфера (хотя существуют и другие виды организации). По буферу при операциях чтения и записи перемещаются два указателя, соответствующие входному и выходному потокам. При этом выходной указатель никогда не может перегнать входной и наоборот. Для создания нового экземпляра такого кольцевого буфера внутри операционной системы используется системный вызов pipe ().

Для организации потокового взаимодействия любых процессов в операционной системе UNIX применяется средство связи, получившее название **FIFO** (от First Input First Output) или именованный pipe . FIFO во всем подобен pip’у, за одним исключением: данные о расположении FIFO в адресном пространстве ядра и его состоянии процессы могут получать не через родственные связи, а через файловую систему. Для этого при создании именованного pip’а на диске заводится файл специального типа, обращаясь к которому процессы могут получить интересующую их информацию. Для создания FIFO используется системный вызов mknod() или существующая в некоторых версиях UNIX функция mkfifo().

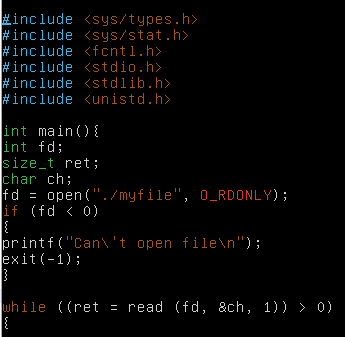
**Программа записи информации в файл**

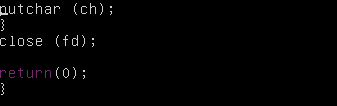
****

****

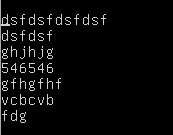
**Программа чтения информации из файла и вывода на экран**

Программа 1



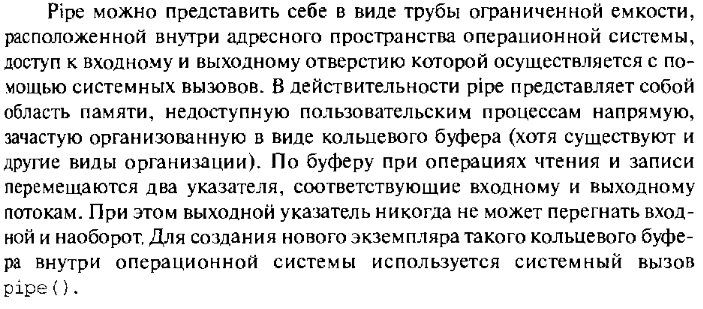


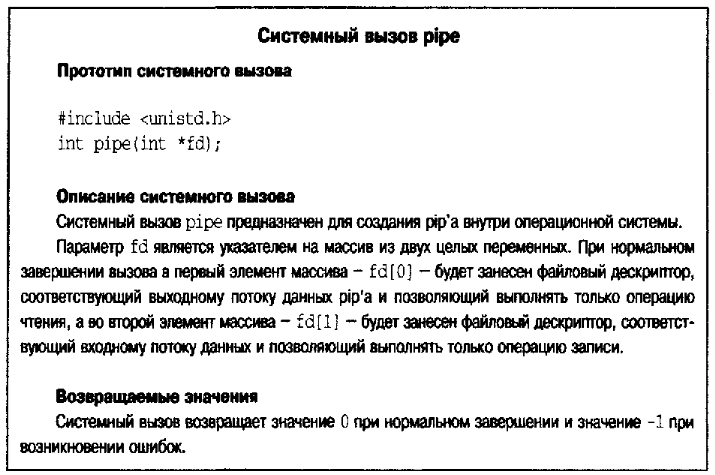
Содержимое myfile



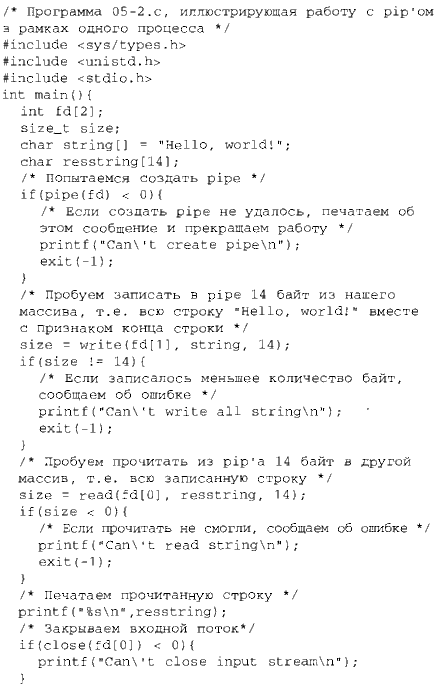
Результат 1

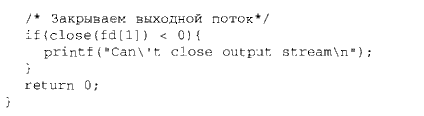




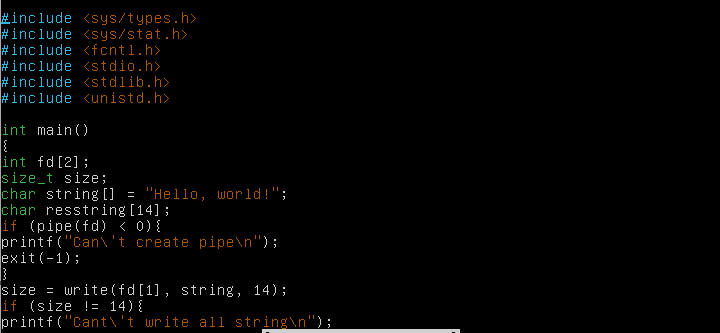


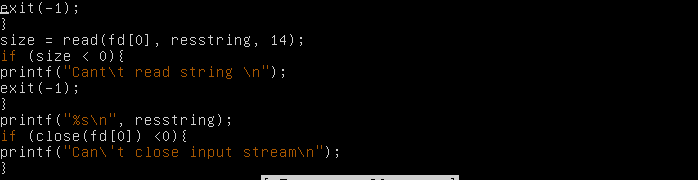
**Программа для pipe в одном процессе**

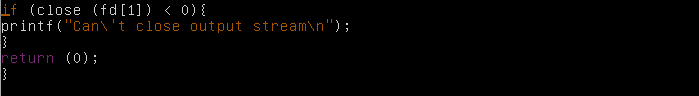




Программа 2



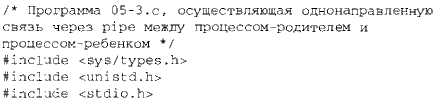


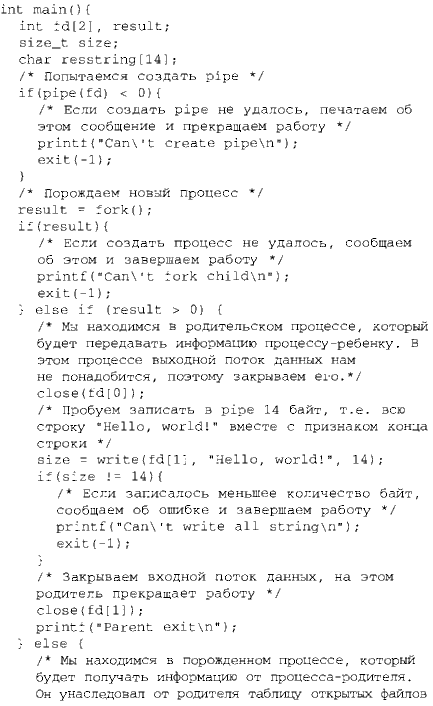


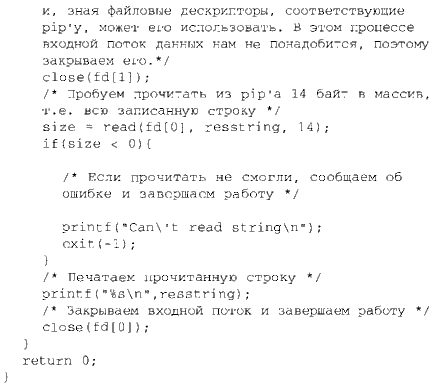
Результат 2



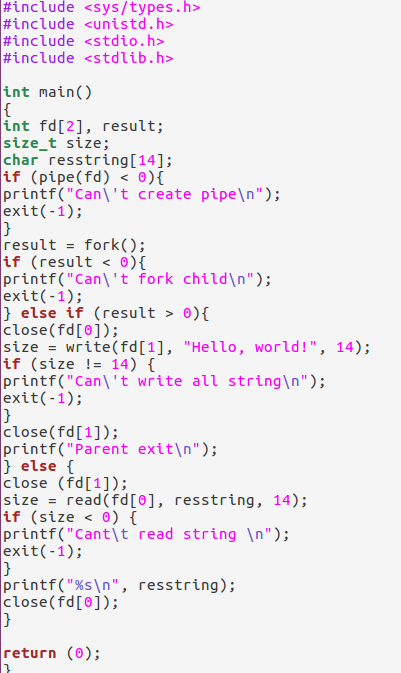
**Программа для организации однонаправленной связи между родственными процессами через pipe**







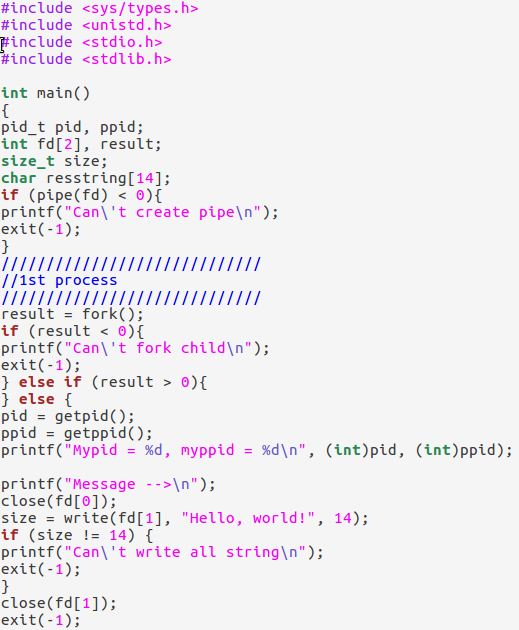
Программа 3

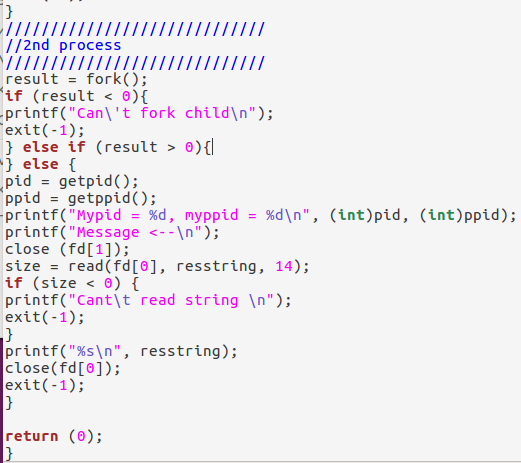


Результат 3

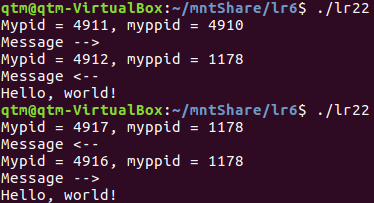


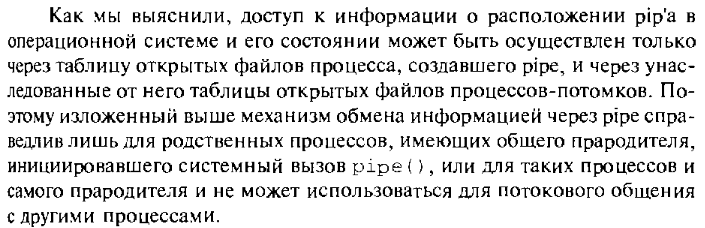
Программа 4

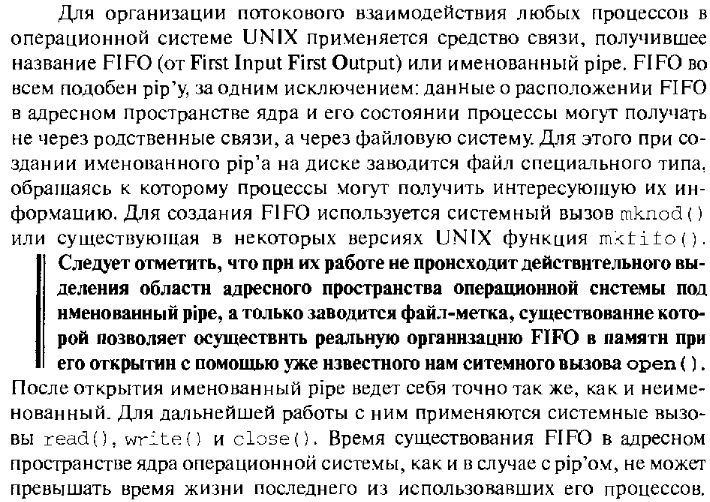


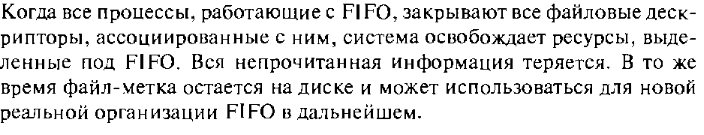


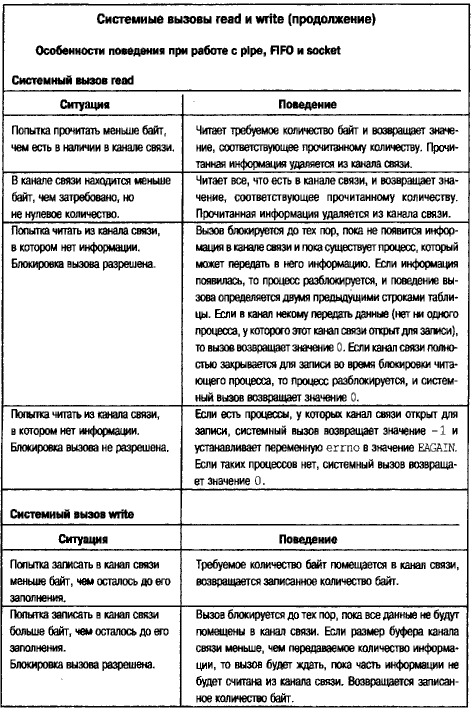
Результат 4

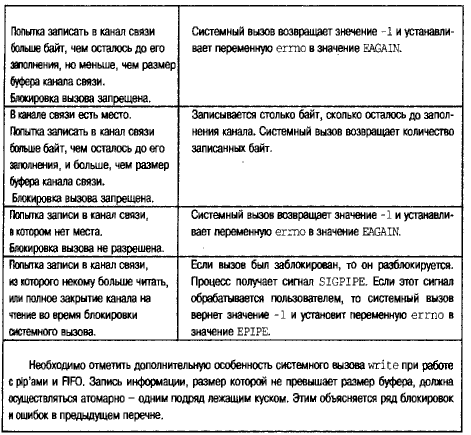




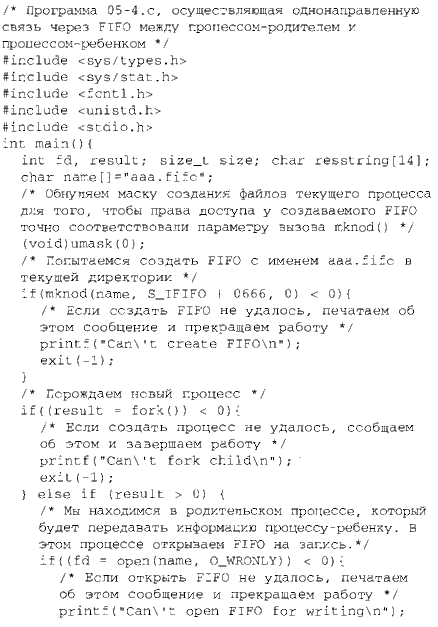


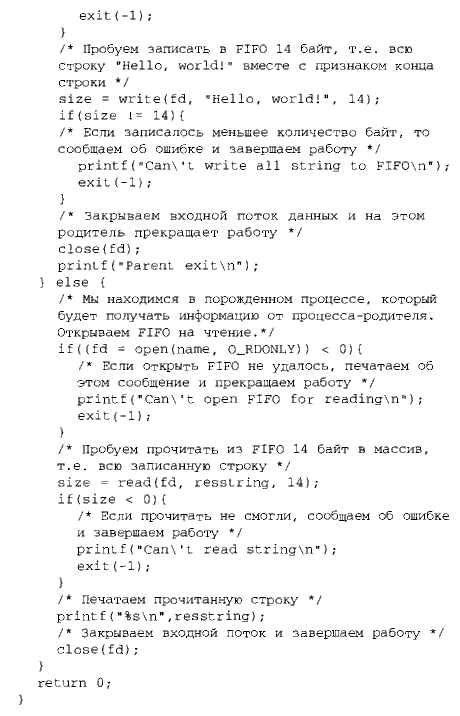


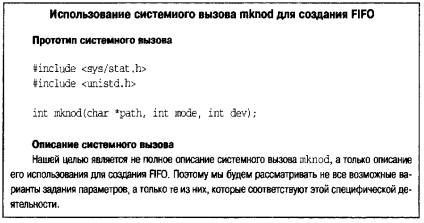


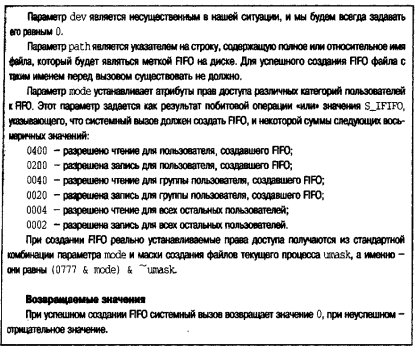


**Программа с FIFO в родственных процессах**





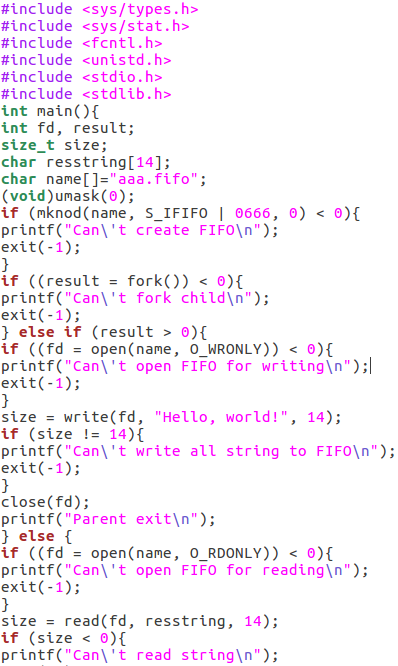


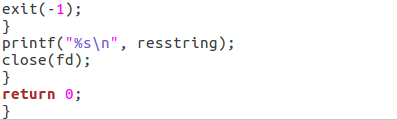


Исходные права доступа равны 0666 - для файлов.

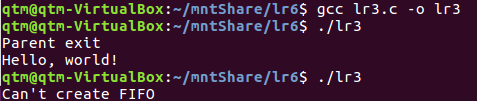
S\_IFIFO – системный вызов должен создать FIFO.

Программа 5

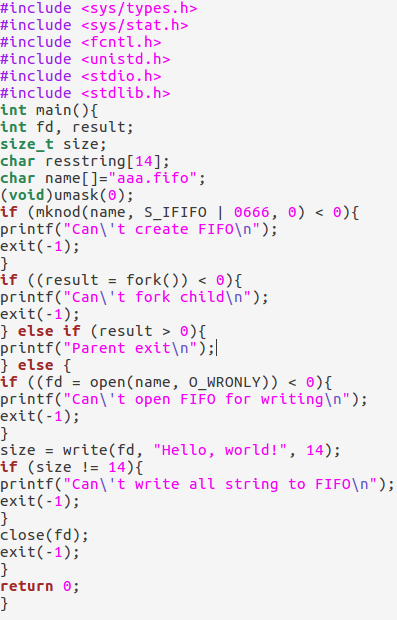




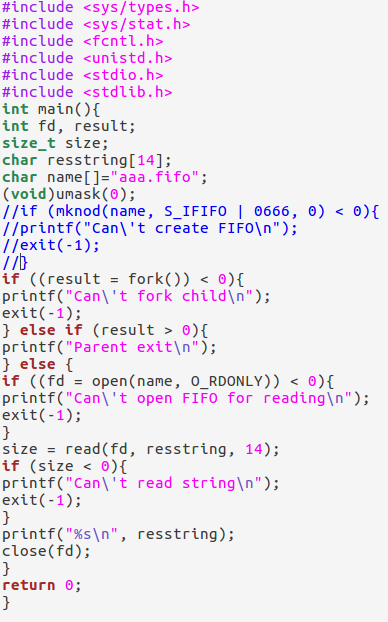
Результат 5



Программа 6\_1



Программа 6\_2



Результат 6

