Лабораторная работа №3 «ПРОГРАММНЫЕ КАНАЛЫ»

**Цель работы.**

Изучение конвейеров (pipes, программных каналов), как простейшего средства коммуникации запущенных процессов. Исследование различных способов организации каналов и их сопоставление.

**Последовательность выполнения работы:**

1. Войдите в систему и скопируйте с разделяемого ресурса в свой HOME-каталог набор исходных файлов для третьего занятия.

2. Скомпилируйте и выполните программу whosortpipe.cpp . Сопоставьте результат выполнения программы с выполнением этих же двух команд из shell в конвейерном режиме ( | ).

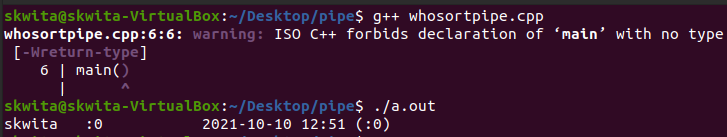


Рис. 3-1 Компиляция и запуск программы «whosortpipe.cpp» последовательно

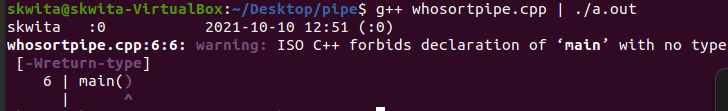


Рис. 3-2 Компиляция и запуск программы «whosortpipe.cpp» в конвейерном режиме

Выполнение двух команд из shell в конвейерном режиме выдаст тот же результат.

3. Программу cmdpipe.cpp запускайте после компиляции, задавая ей при стартах в качестве параметров командной строки пары команд shell для конвейеризации (who и sort ; last и sort ; last и more ; pstree и more). Сопоставьте результаты запусков программы с выполнением тех же пар команд из shell в конвейерном режиме. Можно ли с помощью вызова popen() создать программу, организующую конвейер из трех команд shell , передаваемых ей в качестве параметров командной строки при запуске? Если да, то создайте такую программу, если нет, дайте обоснованный ответ, почему нельзя.

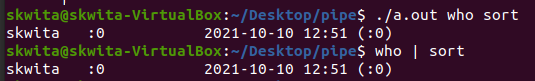


Рис. 3-3 Сравнение результата запуска программы с выполнением тех же пар команд в конвейерном режиме

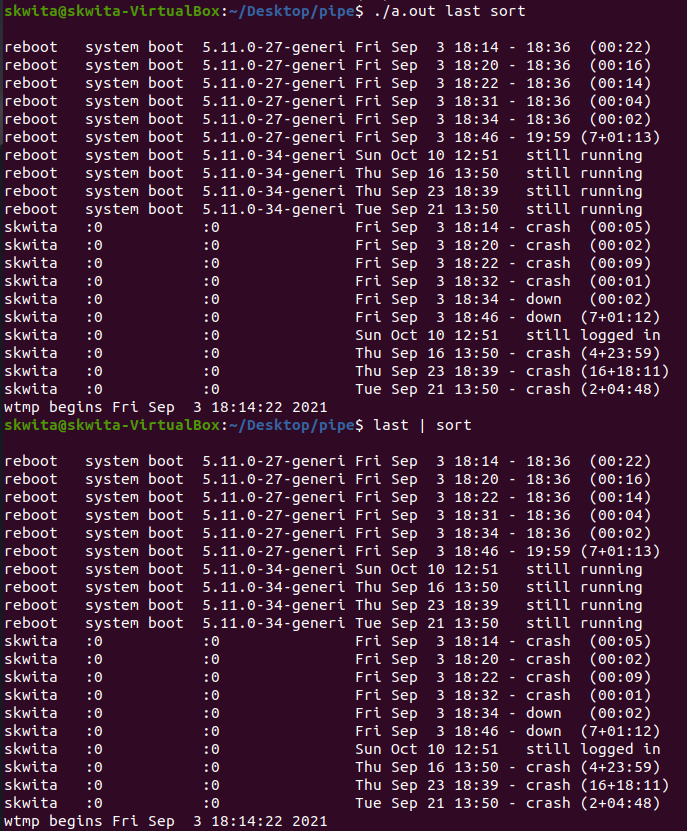


Рис. 3-4 Сравнение результата запуска программы с выполнением тех же пар команд в конвейерном режиме

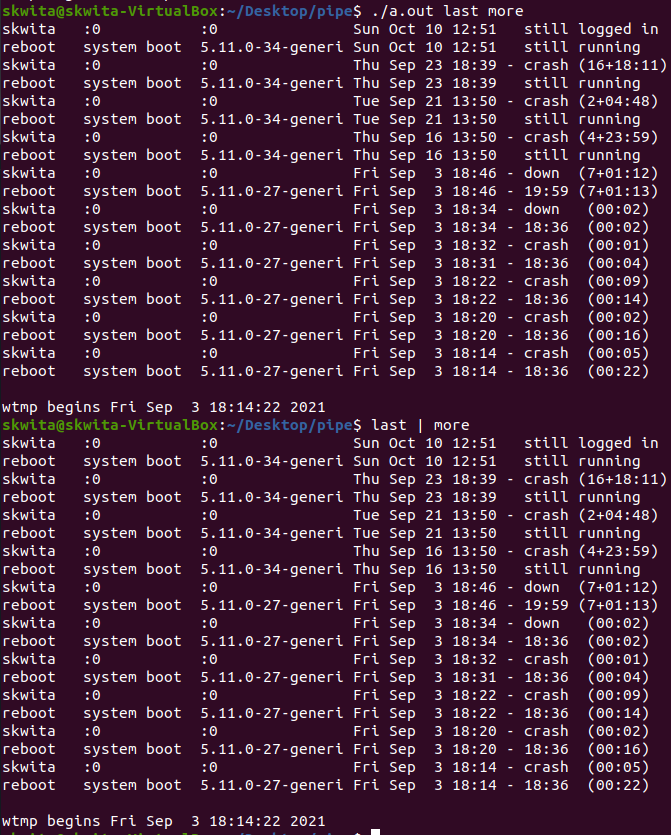


Рис. 3-5 Сравнение результата запуска программы с выполнением тех же пар команд в конвейерном режиме

|  |  |
| --- | --- |
|  |  |
| Рис. 3-6 Результат запуска программы | Рис. 3-7 Результат выполнения команд в конвейерном режиме |

Вызов popen() генерирует дочерний процесс, который запускает exec()-ом на исполнение команду shell, указанную первым параметром popen(). При вызове popen: fin = popen(argv[1], “r”) передается один элемент, значит можно сделать вывод, что организовать конвейер из трех команд не получится.

4. Напишите программу (например, на основе вызовов pipe()), воспринимающую варьируемое количество команд, передаваемых ей при запуске в качестве параметров. Каждая последующая команда должна быть соединена с предыдущей с помощью конвейера. Так, при запуске программы

$ ./a.out last sort more

должны выполняться действия, эквивалентные запуску команд из shell :

$ last | sort | more.

Созданная программа:

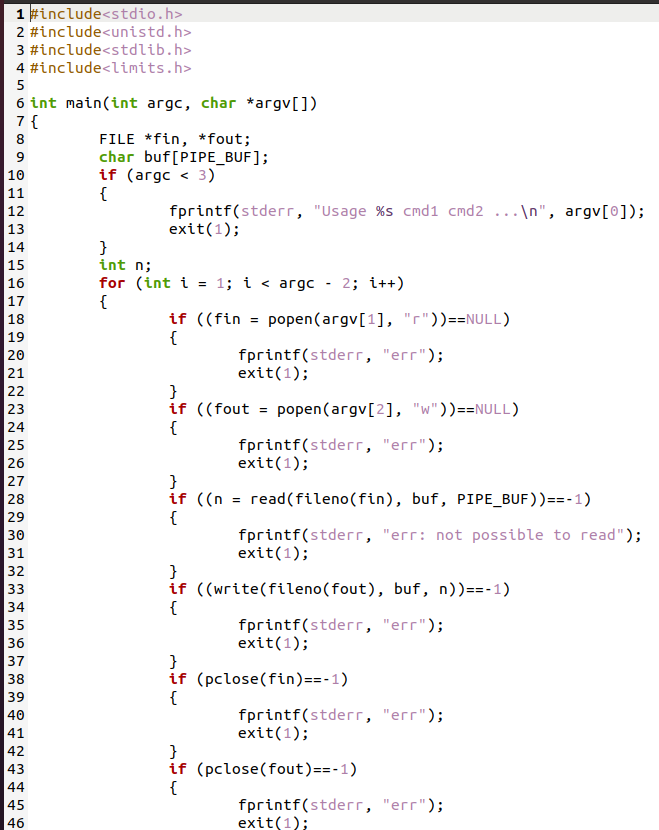
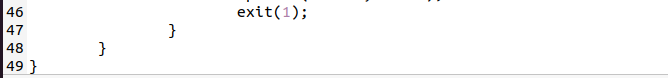
 

Рис. 4-1 Программа.

Данная программа может создать конвейер из любого числа команд (больше 2) с использованием вызова **pipe().**

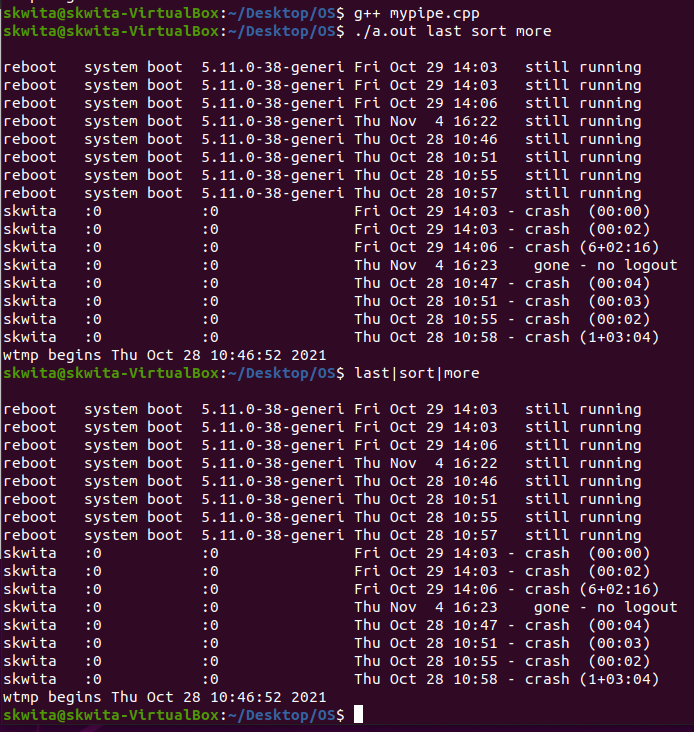


Рис. 4-2 Результат работы программы и выполнением команд в конвейерном режиме.

5. Разберите и выполните пример клиент-серверного взаимодействия, организованного на конвейерах различного типа. Исходный текст примера содержится в файлах pipe\_server.cpp, pipe\_client.cpp и pipe\_local.h и разобран в материалах лекций. Сервер запускается в фоновом режиме. Проанализируйте результаты функционирования данной системы и ее недостатки. Программа-сервер этого примера исполняет каждый командный запрос поочередно. Если какой-либо запрос потребует много времени, все остальные клиентские процессы будут ожидать обслуживания.

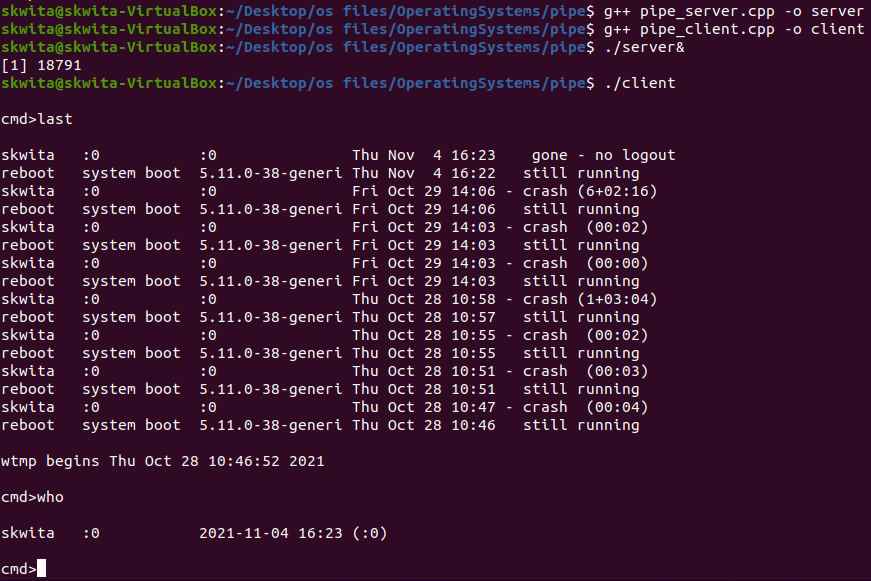


Рис. 5-1 Пример работы

Поскольку сервер выполняет команды последовательно, скорость выполнения каждого отдельного процесса влияет на общую скорость работы программы. Из-за этого если нагрузка будет слишком большой при трудоемком запросе, то время отклика сервера будет тоже довольно большой.

6. Модифицируйте программу pipe\_server.cpp так, чтобы при получении нового сообщения от очередного клиента сервер порождал очередной дочерний процесс для выполнения задачи обслуживания данного запроса (выполнения переданной от клиента команды и переправки клиенту результата).



Рис 6-1 Измененная программа.

Вывод

В ходе лабораторной работы были исследованы способы организации каналов. Изучено понятие конвейер и его свойства. Получены навыки конвейерного запуска команд и была написана программа для запуска конвейером нескольких команд. Также разобран пример клиент-серверного взаимодействия.