Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования «Сибирский государственный университет телекоммуникаций и информатики»

(СибГУТИ)

Кафедра ИКСС

Контрольная работа

по дисциплине: «Операционные системы»

Вариант 3

Выполнил: студент

группы *NN*

*ФИО*

Проверил:

*ФИО*

**Теория:**

**Конвейер** в терминологии [операционных систем](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) семейства [Unix](https://ru.wikipedia.org/wiki/Unix) — некоторое множество [процессов](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81_(%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)), для которых выполнено следующее [перенаправление ввода-вывода](https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D0%B5%D0%BD%D0%B0%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5)): то, что выводит на поток стандартного вывода предыдущий процесс, попадает в поток стандартного ввода следующего процесса. Запуск конвейера реализован с помощью системного вызова pipe().

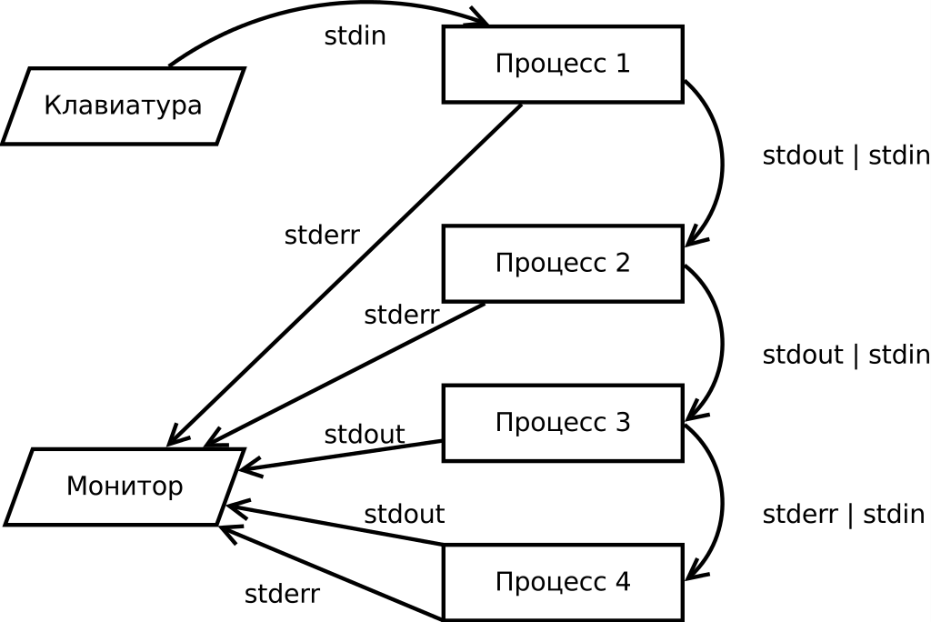


Рис.1 - Пример цепи процессов.

**Именованный канал** или **именованный конвейер** — один из методов [межпроцессного взаимодействия](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D0%B6%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D0%BD%D0%BE%D0%B5_%D0%B2%D0%B7%D0%B0%D0%B8%D0%BC%D0%BE%D0%B4%D0%B5%D0%B9%D1%81%D1%82%D0%B2%D0%B8%D0%B5" \o "Межпроцессное взаимодействие), расширение понятия [конвейера](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D0%B2%D0%B5%D0%B9%D0%B5%D1%80_(UNIX)) в [Unix](https://ru.wikipedia.org/wiki/Unix) и подобных [ОС](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0). Именованный канал позволяет различным процессам обмениваться данными, даже если программы, выполняющиеся в этих процессах, изначально не были написаны для взаимодействия с другими программами.

Это понятие также существует и в [Microsoft Windows](https://ru.wikipedia.org/wiki/Microsoft_Windows), хотя там его [семантика](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D0%BC%D0%B0%D0%BD%D1%82%D0%B8%D0%BA%D0%B0) существенно отличается. Традиционный канал — «безымянен», потому что существует анонимно и только во время выполнения процесса. Именованный канал — существует в [системе](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) и после завершения процесса. Он должен быть «отсоединён» или удалён, когда уже не используется. Процессы обычно подсоединяются к каналу для осуществления взаимодействия между ними.

В UNIX:

Вместо традиционного, безымянного конвейера оболочки, именованный канал создаётся явно с помощью [mknod](https://ru.wikipedia.org/w/index.php?title=Mknod&action=edit&redlink=1" \o "Mknod (страница отсутствует)) или [mkfifo](https://ru.wikipedia.org/wiki/Mkfifo" \o "Mkfifo), и два различных процесса могут обратиться к нему по имени.

Например, можно создать канал и настроить [gzip](https://ru.wikipedia.org/wiki/Gzip" \o "Gzip) на сжатие того, что туда попадает

В Windows:

В Windows дизайн именованных каналов смещён к взаимодействию [«клиент-сервер»](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B8%D0%B5%D0%BD%D1%82-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80), и они работают во многом как [сокеты](https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D0%BA%D0%B5%D1%82_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D1%8B%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81)): помимо обычных операций чтения и записи, именованные каналы в Windows поддерживают явный «пассивный» режим для серверных приложений.

Именованные каналы не существуют постоянно и не могут, в отличие от Unix, быть созданы как специальные файлы в произвольной доступной для записи файловой системе, но имеют временные имена, которые выделяются в корне файловой системы именованных каналов.

**Команды для управления процессами**

**Команда ps** - предназначена для вывода информации о выполняемых процессах.

**Программа top** - предназначена для вывода информации о процессах в реальном времени.

Команда **nice** - команда nice выполняет указанную команду с пониженным приоритетом, коэффициент понижения указывается в диапазоне 1..19 (по умолчанию он равен 10). Суперпользователь может повышать приоритет команды, для этого нужно указать отрицательный коэффициент, например --10.

**nohup** - игнорирование сигналов прерывания. Nohup выполняет запуск команды в режиме игнорирования сигналов

**kill** - принудительное завершение процесса

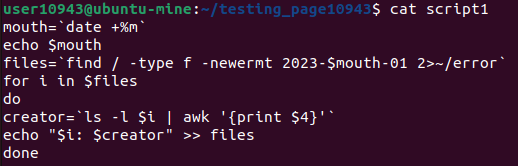
**Команды выполнения процессов в фоновом режиме - jobs, fg, bg**

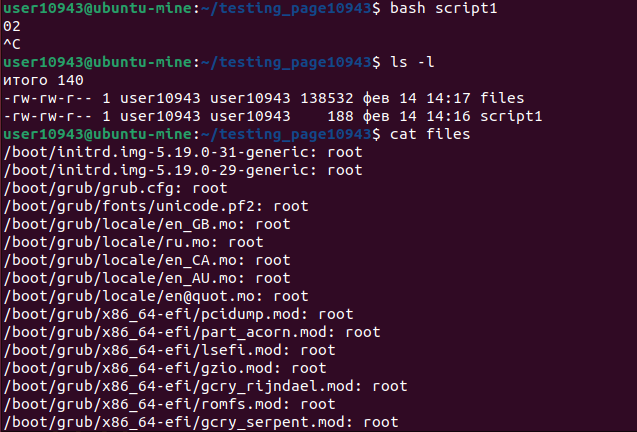
Команда jobs выводит список процессов, которые выполняются в фоновом режиме, fg - переводит процесс в нормальный режим ("на передний план" - foreground), а bg - в фоновый.

**Источники**

1. *Wikipedia «Конвейер (Unix)»* [электронный ресурс] // <https://is.gd/YKF14Y>
2. *Wikipedia «Именованный канал»* [электронный ресурс] // <https://is.gd/cBd46E>
3. *OpenNet «FIFO – именованные каналы»* [электронный ресурс] // <https://www.opennet.ru/docs/RUS/linux_parallel/node17.html>
4. *OpenNet «Управление процессами в Linux»* [электронный ресурс] // <http://www.opennet.ru/docs/RUS/lnx_process/process2.html>

Задание 1.



  
mouth=`date +%m` # берётся дата месяца в числовой форме

files=`find / -type f -newermt 2023-$mouth-01 2>~/error` # поиск файлов по всей системе, созданных позже 2023-$mouth-01 числа, ошибки отправляются в файле error

for i in $files # цикл по найденным файлам

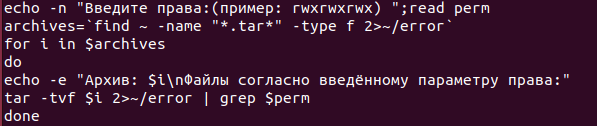
do

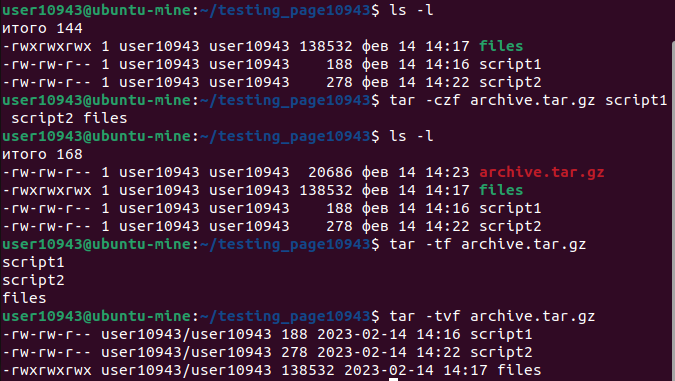
creator=`ls -l $i | awk '{print $4}'` # имя создателя (записывается 4 столбец из вывода команды ls -l название\_файла (абсолютный путь к файлу)

echo "$i: $creator" >> files # в файл files записывается название\_файла(абсолютный путь к файлу): создатель\_файла

done

Задание 2.





Изображение выглядит как текст

Автоматически созданное описание

echo -n "Введите права:(пример: rwxrwxrwx) ";read perm # вводятся права для поиска файлов в архивах

archives=`find ~ -name "\*.tar\*" -type f 2>~/error` # поиск всех архивов в домашней директории

for i in $archives # цикл по архивам

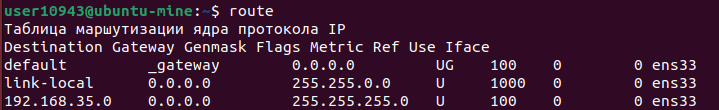
do

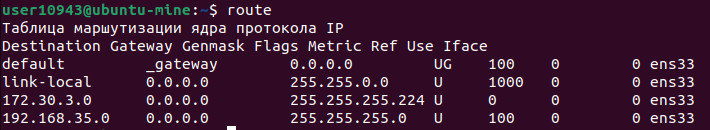
echo -e "Архив: $i\nФайлы согласно введённому параметру права:"

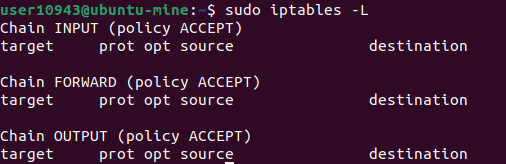
tar -tvf $i 2>~/error | grep $perm # выводится содержимое архива (-t – сформировать список содержимого, -v – подробная информация, -f – вывести информацию, после данный вывод форматируется согласно введённым правам с помощью команды grep (там где есть введённые права)

done

Задание 3.

route add -net 172.30.3.0/27 ens33  
  

2. iptables -A INPUT -s 172.30.3.0/27 -p tcp --dport 443 -j DROP  
  
Изображение выглядит как текст

Автоматически созданное описание

Запретить https-запросы (tcp port 443) для хостов 172.30.3.0/27 (DROP)