Шаблон отчёта по лабораторной работе

Простейший вариант

Хань Цзянтао

Содержание

[1 Цель работы 1](#_Toc13074)

[2 Задание 1](#_Toc30319)

[3 Теоретическое введение 2](#_Toc9281)

[4 Выполнение лабораторной работы 2](#_Toc32641)

[4.0.1 Перенаправление ввода-вывода 2](#_Toc6730)

[4.0.2 Конвейер 3](#_Toc29758)

[4.0.3Поиск файла 3](#_Toc6055)

[4.0.4 Фильтрация текста 4](#_Toc18700)

[4.0.5 Фильтрация текста 5](#_Toc5035)

[4.0.6 Управление задачами 6](#_Toc16980)

[4.0.7 Управление процессами 6](#_Toc18935)

[4.0.8 Получение информации о процессах Команда ps используется для получения информации о процессах. Формат команды: 7](#_Toc31398)

[Список литературы 8](#_Toc19076)

# 1 Цель работы

Ознакомление с инструментами поиска файлов и фильтрации текстовых данных.Приобретение практических навыков: по управлению процессами (и заданиями), попроверке использования диска и обслуживанию файловых систем.

# 2 Задание

1. Поиск файлов
2. Перенаправление ввода-вывода:
3. Просмотр запущенных процессов:

# Теоретическое введение

В ходе выполнения этой лабораторной работы студенты обычно изучают следующие темы:

Поиск файлов с использованием команды find:

Команда find в Linux предоставляет мощный инструмент для поиска файлов и каталогов в файловой системе. Она может быть использована для поиска файлов по различным критериям, таким как имя файла, тип, размер и т. д.

Перенаправление ввода-вывода:

В Linux перенаправление ввода-вывода позволяет управлять потоками ввода, вывода и ошибок командной строки. Оно позволяет перенаправить вывод одной команды как ввод другой команды или в файл, а также использовать файлы в качестве ввода или вывода для команд.

Просмотр запущенных процессов с помощью команды ps:

Команда ps позволяет просматривать информацию о процессах, запущенных на компьютере. Она выводит список процессов с их идентификаторами, статусами, использованием ресурсов и другой полезной информацией.

Применение практических навыков:

Студенты выполняют ряд задач, используя команды find, перенаправление ввода-вывода и команду ps, что позволяет им закрепить полученные знания и развить навыки работы в командной строке Linux.

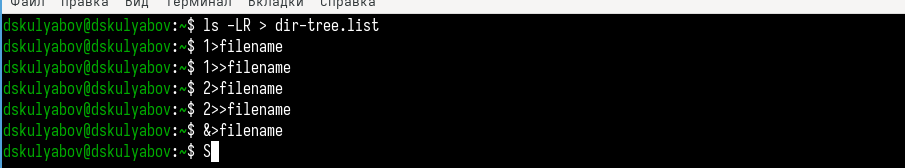
В ходе выполнения лабораторной работы студенты обычно сталкиваются с различными сценариями использования командной строки, что позволяет им понять принципы работы операционной системы Linux и углубить свои навыки в области администрирования и разработки.

# 4 Выполнение лабораторной работы

### 4.0.1 **Перенаправление ввода-вывода**

В системе по умолчанию открыто три специальных потока: – stdin — стандартный поток ввода (по умолчанию: клавиатура), файловый дескриптор 0; – stdout — стандартный поток вывода (по умолчанию: консоль), файловый дескриптор 1; – stderr — стандартный поток вывод сообщений об ошибках (по умолчанию: консоль), файловый дескриптор 2. Большинство используемых в консоли команд и программ записывают результаты своей работы в стандартный поток вывода stdout. Например, команда ls выводит в стан- дартный поток вывода (консоль) список файлов в текущей директории. Потоки вывода и ввода можно перенаправлять на другие файлы или устройства. Проще всего это делается с помощью символов >, >>, <, <<. Рассмотрим пример.

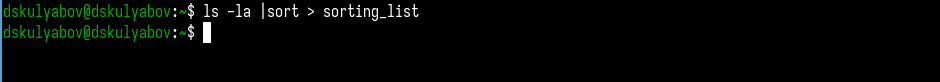
ls -lR > dir-tree.list  
1>filename  
1>>filename  
2>filename  
2>>filename  
&>filename



### 4.0.2 Конвейер

* Конвейер (pipe) служит для объединения простых команд или утилит в цепочки, в ко- торых результат работы предыдущей команды передаётся последующей. Синтаксис следующий: команда 1 | команда 2 Конвейеры можно группировать в цепочки и выводить с помощью перенаправления в файл, например:

ls -la |sort > sortilg\_list 10



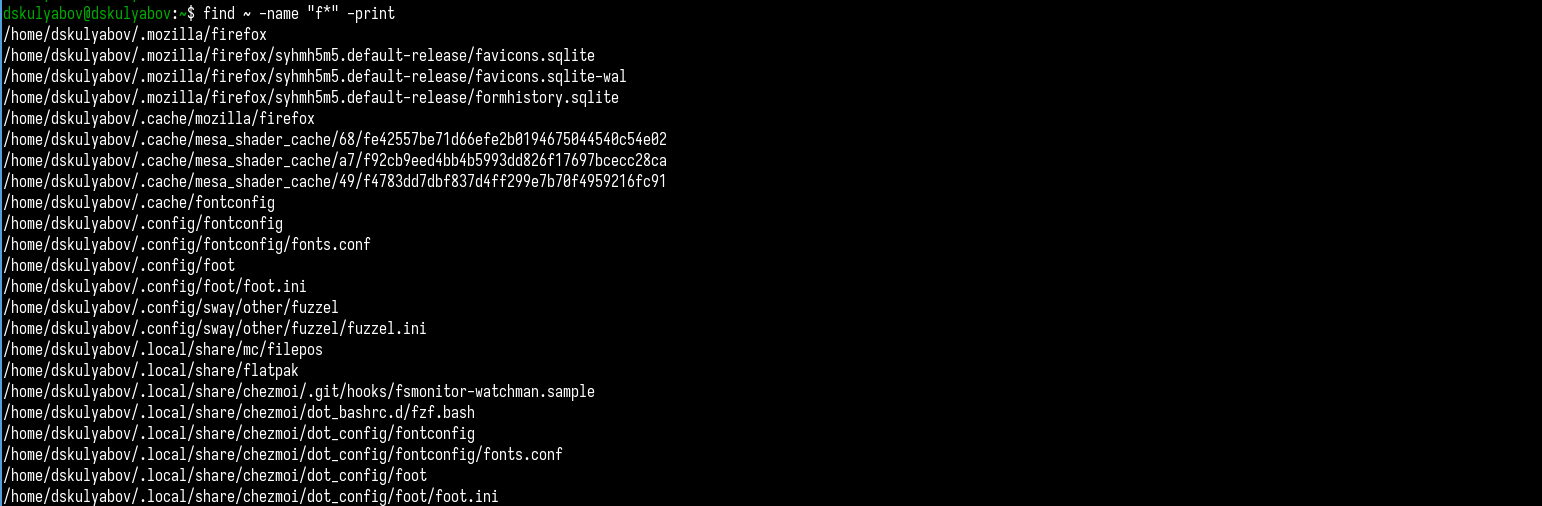
### 4.0.3Поиск файла

​Команда find используется для поиска и отображения на экран имён файлов, соответ- ствующих заданной строке символов. Формат команды:

find путь [-опции]

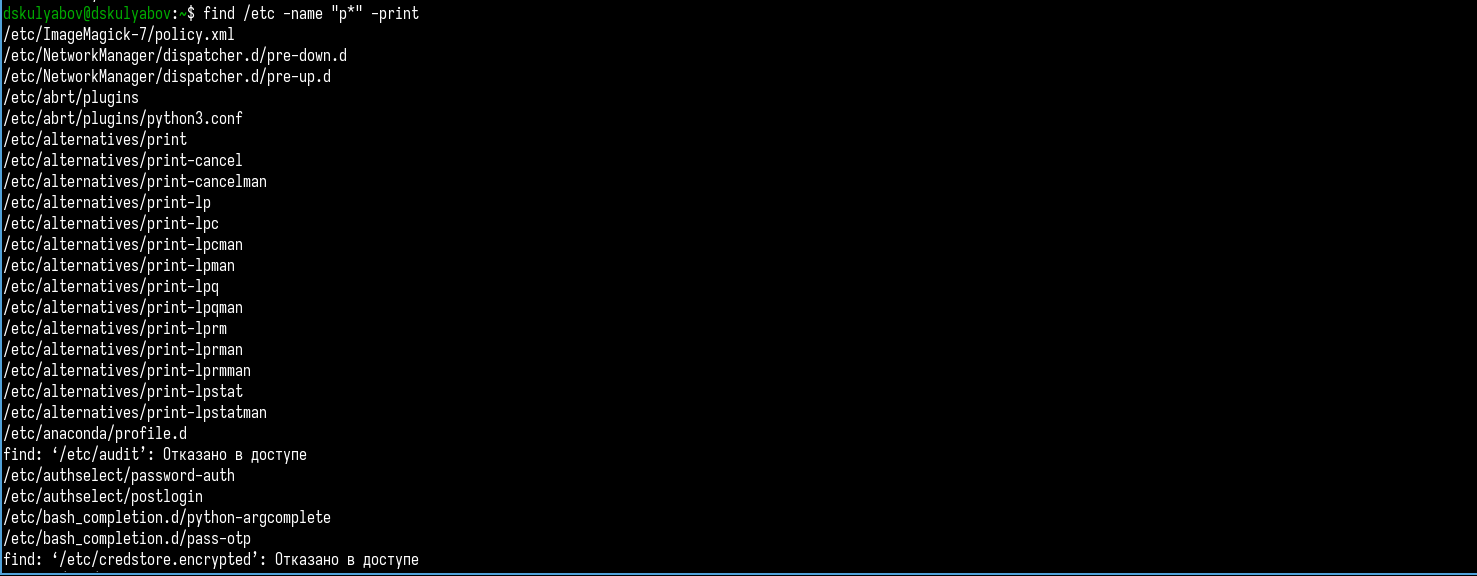
12 Путь определяет каталог, начиная с которого по всем подкаталогам будет вестись поиск. Примеры: 1. Вывести на экран имена файлов из вашего домашнего каталога и его подкаталогов, начинающихся на f:

find ~ -name "f\*" -print



13 Здесь ~ — обозначение вашего домашнего каталога, -name — после этой опции указы- вается имя файла, который нужно найти, “f\*” — строка символов, определяющая имя файла, -print — опция, задающая вывод результатов поиска на экран. 2. Вывести на экран имена файлов в каталоге /etc, начинающихся с символа p:

find /etc -name "p\*" -print



lab4 3. Найти в Вашем домашнем каталоге файлы, имена которых заканчиваются символом ~ и удалить их:

find ~ -name "\*~" -exec rm "{}" \;



lab5 Здесь опция -exec rm “{}” ; задаёт применение команды rm ко всем файлам, име- на которых соответствуют указанной после опции -name строке символов. Для просмотра опций команды find воспользуйтесь командой man.

### 4.0.4 Фильтрация текста

​ Найти в текстовом файле указанную строку символов позволяет команда grep. Формат команды:

grep строка имя\_файла

роме того, команда grep способна обрабатывать стандартный вывод других команд (любой текст). Для этого следует использовать конвейер, связав вывод команды с вводом grep. Примеры: 1. Показать строки во всех файлах в вашем домашнем каталоге с именами, начинающи- мися на f, в которых есть слово begin:

grep begin f\*

06

1. Найти в текущем каталоге все файлы, содержащих в имени «лаб»:

ls -l | grep лаб

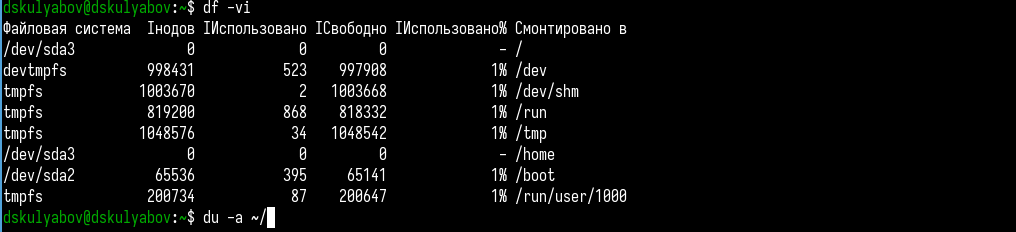
### 4.0.5 Фильтрация текста

Команда df показывает размер каждого смонтированного раздела диска. Формат команды:

df [-опции] [файловая\_система]

Пример:

df -vi

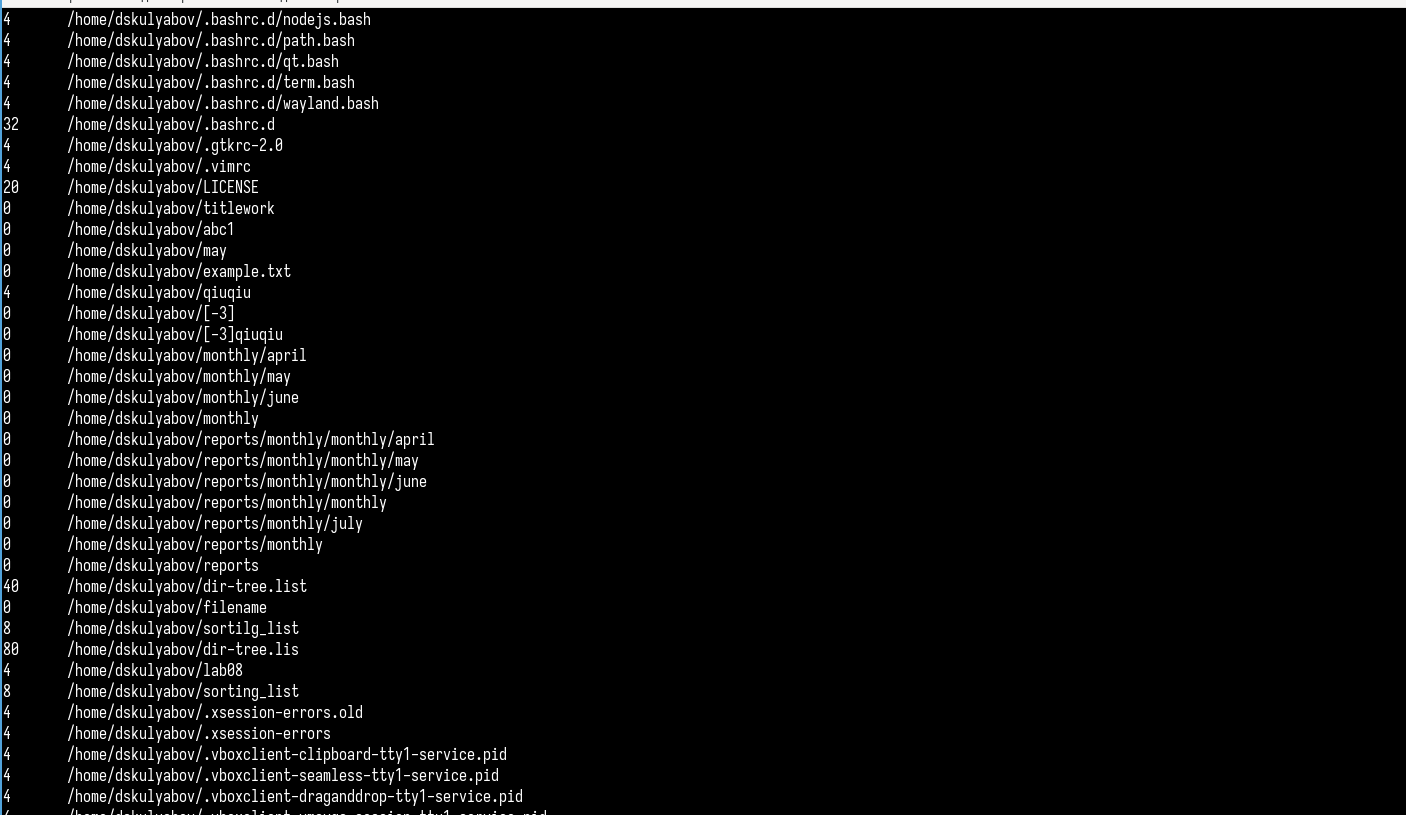


lab7 Команда du показывает число килобайт, используемое каждым файлом или каталогом. Формат команды:

du [-опции] [имя\_файла...]

Пример.

du -a ~/



На afs можно посмотреть использованное пространство командой

fs quota

### 4.0.6 Управление задачами

Любую выполняющуюся в консоли команду или внешнюю программу можно запустить в фоновом режиме. Для этого следует в конце имени команды указать знак амперсанда &. Например:

gedit &



Будет запущен текстовой редактор gedit в фоновом режиме. Консоль при этом не будет заблокирована. Запущенные фоном программы называются задачами (jobs). Ими можно управлять с помощью команды jobs, которая выводит список запущенных в данный момент задач. Для завершения задачи необходимо выполнить команду

kill %номер задачи

### 4.0.7 Управление процессами

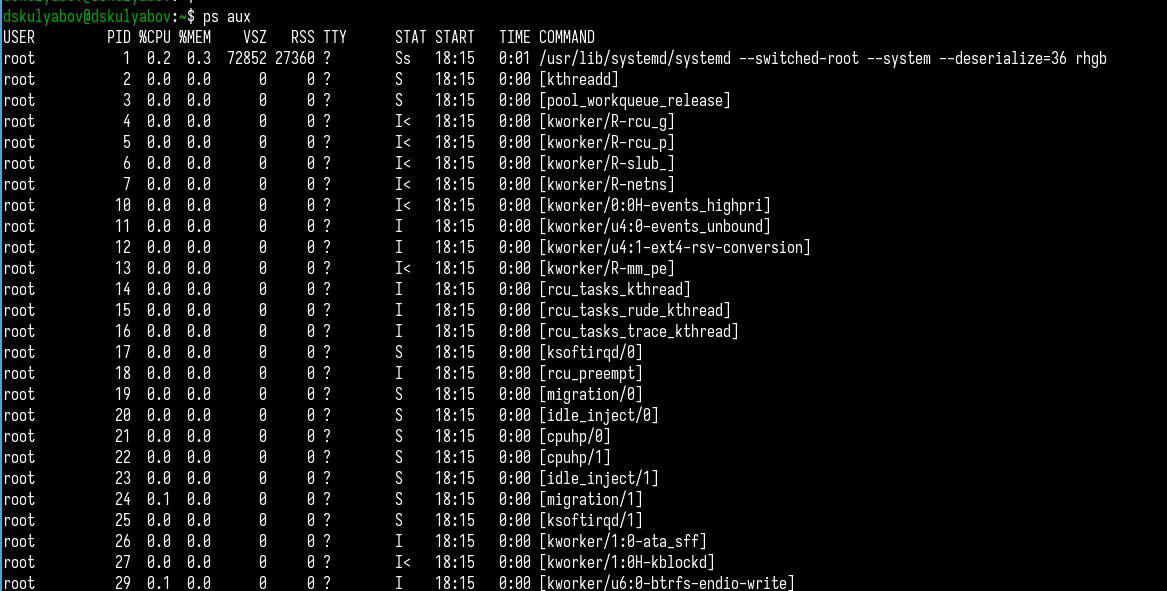
Любой команде, выполняемой в системе, присваивается идентификатор процесса (process ID). Получить информацию о процессе и управлять им, пользуясь идентифи- катором процесса, можно из любого окна командного интерпретатора.

### 4.0.8 Получение информации о процессах Команда ps используется для получения информации о процессах. Формат команды:

ps [-опции]

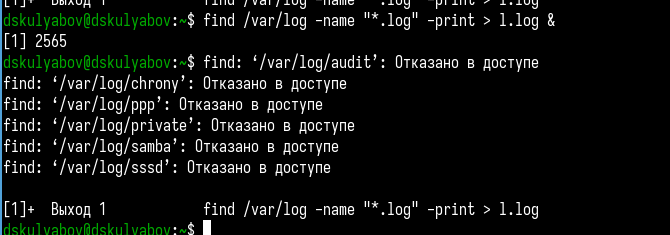
Для получения информации о процессах, управляемых вами и запущенных (работаю- щих или остановленных) на вашем терминале, используйте опцию aux. Пример:

ps aux



Для запуска команды в фоновом режиме необходимо в конце командной строки ука- зать знак & (амперсанд). Пример работы, требующей много машинного времени для выполнения, и которую целесообразно запустить в фоновом режиме:

find /var/log -name "\*.log" -print > l.log &



# Список литературы

::: {#refs}

1. GDB: The GNU Project Debugger. — URL: https://www.gnu.org/software/gdb/.
2. GNU Bash Manual. — 2016. — URL: https://www.gnu.org/software/bash/manual/.
3. Midnight Commander Development Center. — 2021. — URL: https://midnight

commander. org/.

1. NASM Assembly Language Tutorials. — 2021. — URL: https://asmtutor.com/.
2. Newham C. Learning the bash Shell: Unix Shell Programming. — O’Reilly

Media, 2005. — 354 с. — (In a Nutshell). — ISBN 0596009658. — URL:

http://www.amazon.com/Learningbash-Shell-Programming-Nutshell/dp/0596009658.

1. Robbins A. Bash Pocket Reference. — O’Reilly Media, 2016. — 156 с. — ISBN

978-1491941591.

1. The NASM documentation. — 2021. — URL: https://www.nasm.us/docs.php.
2. Zarrelli G. Mastering Bash. — Packt Publishing, 2017. — 502 с. — ISBN
3. Колдаев В. Д., Лупин С. А. Архитектура ЭВМ. — М. : Форум, 2018.
4. Куляс О. Л., Никитин К. А. Курс программирования на ASSEMBLER. — М. :

Солон-Пресс, 2017.

1. Новожилов О. П. Архитектура ЭВМ и систем. — М. : Юрайт, 2016.
2. Расширенный ассемблер: NASM.— 2021.—URL: https://www.opennet.ru/docs/RUS/nasm/.
3. Робачевский А., Немнюгин С., Стесик О. Операционная система UNIX. — 2-е

изд. — БХВПетербург, 2010. — 656 с. — ISBN 978-5-94157-538-1.

1. Столяров А. Программирование на языке ассемблера NASM для ОС Unix.— 2-

е изд.—М. : МАКС Пресс, 2011.—URL: http://www.stolyarov.info/books/asm\_unix.:::