

Шаблон отчёта по лабораторной работе

Простейший вариант

Хань Цзянтао

Содержание

1	Цель работы.....	1
2	Задание.....	1
3	Теоретическое введение	2
4	Выполнение лабораторной работы.....	3
	4.0.1 Перенаправление ввода-вывода.....	3
	4.0.2 Конвейер.....	3
	4.0.3 Поиск файла	4
	4.0.4 Фильтрация текста	5
	4.0.5 Фильтрация текста	5
	4.0.6 Управление задачами	6
	4.0.7 Управление процессами	7
	4.0.8 Получение информации о процессах Команда ps используется для получения информации о процессах. Формат команды:	7
	Список литературы.....	8

1 Цель работы

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

2 Задание

1. Поиск файлов

2. Перенаправление ввода-вывода:
3. Просмотр запущенных процессов:

3 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 навык работы в командной строке Linux.

В ходе выполнения лабораторной работы студенты обычно сталкиваются с различными сценариями использования командной строки, что позволяет им понять принц

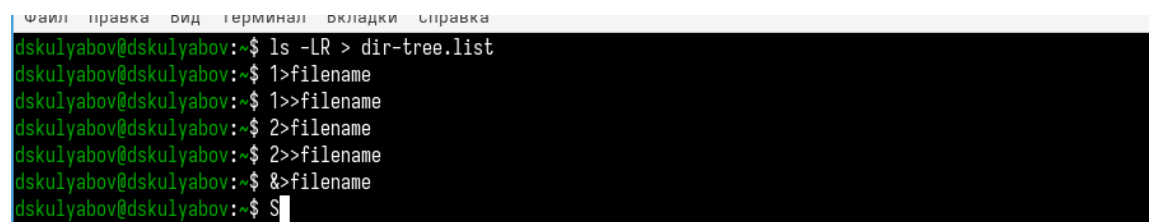
ипы работы операционной системы Linux и углубить свои навыки в области администрирования и разработки.

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 стандартный 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
```



```
dskulyabov@dskulyabov:~$ ls -lR > dir-tree.list
dskulyabov@dskulyabov:~$ 1>filename
dskulyabov@dskulyabov:~$ 1>>filename
dskulyabov@dskulyabov:~$ 2>filename
dskulyabov@dskulyabov:~$ 2>>filename
dskulyabov@dskulyabov:~$ &>filename
dskulyabov@dskulyabov:~$ $
```

4.0.2 Конвейер

- Конвейер (pipe) служит для объединения простых команд или утилит в цепочки, в которых 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
dskulyabov@dskulyabov:~$ ls -la |sort > sorting_list
dskulyabov@dskulyabov:~$ $
```

4.0.3 Поиск файла

Команда `find` используется для поиска и отображения на экран имён файлов, соответствующих 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`

```
dskulyabov@dskulyabov:~$ find ~ -name "f*" -print
/home/dskulyabov/.mozilla/firefox
/home/dskulyabov/.mozilla/firefox/syhmh5m5.default-release/favicons.sqlite
/home/dskulyabov/.mozilla/firefox/syhmh5m5.default-release/favicons.sqlite-wal
/home/dskulyabov/.mozilla/firefox/syhmh5m5.default-release/formhistory.sqlite
/home/dskulyabov/.cache/mozilla/firefox
/home/dskulyabov/.cache/mesa_shader_cache/68/fe42557be71d66efe2b0194675844540c54e02
/home/dskulyabov/.cache/mesa_shader_cache/a7/f92cb9eed4bb4b5993dd826f17697bcacc28ca
/home/dskulyabov/.cache/mesa_shader_cache/49/f4783dd7dbf837d4ff299e7b70f4959216fc91
/home/dskulyabov/.cache/fontconfig
/home/dskulyabov/.config/fontconfig
/home/dskulyabov/.config/fontconfig/fonts.conf
/home/dskulyabov/.config/foot
/home/dskulyabov/.config/foot/foot.ini
/home/dskulyabov/.config/sway/other/fuzzel
/home/dskulyabov/.config/sway/other/fuzzel/fuzzel.ini
/home/dskulyabov/.local/share/mc/filepos
/home/dskulyabov/.local/share/flatpak
/home/dskulyabov/.local/share/chezmoi/.git/hooks/fsmonitor-watchman.sample
/home/dskulyabov/.local/share/chezmoi/.dot_bashrc.d/fzf.bash
/home/dskulyabov/.local/share/chezmoi/.dot_config/fontconfig
/home/dskulyabov/.local/share/chezmoi/.dot_config/fontconfig/fonts.conf
/home/dskulyabov/.local/share/chezmoi/.dot_config/foot
/home/dskulyabov/.local/share/chezmoi/.dot_config/foot/foot.ini
```

13 Здесь `~` — обозначение вашего домашнего каталога, `-name` — после этой опции указывается 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`

```
dskulyabov@dskulyabov:~$ find /etc -name "p*" -print
/etc/ImageMagick-7/policy.xml
/etc/NetworkManager/dispatcher.d/pre-down.d
/etc/NetworkManager/dispatcher.d/pre-up.d
/etc/abrt/plugins
/etc/abrt/plugins/python3.conf
/etc/alternatives/print
/etc/alternatives/print-cancel
/etc/alternatives/print-cancelman
/etc/alternatives/print-lp
/etc/alternatives/print-lpc
/etc/alternatives/print-lpcman
/etc/alternatives/print-lpman
/etc/alternatives/print-lpq
/etc/alternatives/print-lpqman
/etc/alternatives/print-lprm
/etc/alternatives/print-lprman
/etc/alternatives/print-lprman
/etc/alternatives/print-lpstat
/etc/alternatives/print-lpstatman
/etc/anaconda/profile.d
find: '/etc/audit': Отказано в доступе
/etc/authselect/password-auth
/etc/authselect/postlogin
/etc/bash_completion.d/python-argcomplete
/etc/bash_completion.d/pass-otp
find: '/etc/credstore.encrypted': Отказано в доступе
```

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

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

```
find /etc/passwd -name "*" -exec rm "{}" \;
```

lab5 Здесь опция -exec rm "{}" ; задаёт применение команды rm ко всем файлам, имена 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

Кроме 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, начинающимися на f, в которых есть слово begin:

grep begin f*

```
Файл  Правка  Вид  Терминал  Вкладки  Справка
dskulyabov@dskulyabov:~$ grep begin f*
dskulyabov@dskulyabov:~$ ls -l | grep лаб
```

2. 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

```
dskulyabov@dskulyabov:~$ df -vi
Файловая система  Инодов  ИИспользовано  ИСвободно  ИИспользовано%  Смонтировано в
/dev/sda3          0         0           0             0%              - /
devtmpfs           998431      523       997908          1% /dev
tmpfs              1003670       2     1003668          1% /dev/shm
tmpfs              819200      868     818332          1% /run
tmpfs              1048576      34     1048542          1% /tmp
/dev/sda3           0         0           0             0%              - /home
/dev/sda2           65536      395     65141           1% /boot
tmpfs              200734      87     200647           1% /run/user/1000
dskulyabov@dskulyabov:~$ du -a ~/
```

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

du [-опции] [имя_файла...]

Пример.

```
du -a ~/
```

```
4 /home/dskulyabov/.bashrc.d/nodejs.bash
4 /home/dskulyabov/.bashrc.d/path.bash
4 /home/dskulyabov/.bashrc.d/qt.bash
4 /home/dskulyabov/.bashrc.d/term.bash
4 /home/dskulyabov/.bashrc.d/wayland.bash
32 /home/dskulyabov/.bashrc.d
4 /home/dskulyabov/.gtkrc-2.0
4 /home/dskulyabov/.vimrc
20 /home/dskulyabov/LICENSE
0 /home/dskulyabov/titlework
0 /home/dskulyabov/abc1
0 /home/dskulyabov/may
0 /home/dskulyabov/example.txt
4 /home/dskulyabov/qiuqiu
0 /home/dskulyabov/[-3]
0 /home/dskulyabov/[-3]qiuqiu
0 /home/dskulyabov/monthly/april
0 /home/dskulyabov/monthly/may
0 /home/dskulyabov/monthly/june
0 /home/dskulyabov/monthly
0 /home/dskulyabov/reports/monthly/monthly/april
0 /home/dskulyabov/reports/monthly/monthly/may
0 /home/dskulyabov/reports/monthly/monthly/june
0 /home/dskulyabov/reports/monthly/monthly
0 /home/dskulyabov/reports/monthly/monthly/july
0 /home/dskulyabov/reports/monthly
0 /home/dskulyabov/reports
40 /home/dskulyabov/dir-tree.list
0 /home/dskulyabov/filename
8 /home/dskulyabov/sorting_list
80 /home/dskulyabov/dir-tree.lis
4 /home/dskulyabov/lab08
8 /home/dskulyabov/sorting_list
4 /home/dskulyabov/.xsession-errors.old
4 /home/dskulyabov/.xsession-errors
4 /home/dskulyabov/.vboxclient-clipboard-tty1-service.pid
4 /home/dskulyabov/.vboxclient-seamless-tty1-service.pid
4 /home/dskulyabov/.vboxclient-draganddrop-tty1-service.pid
4 /home/dskulyabov/.vboxclient-seamless-tty1-service.pid
```

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 &
```

```
dskulyabov@dskulyabov:~$ gedit &
[1] 2490
dskulyabov@dskulyabov:~$ kill %1
dskulyabov@dskulyabov:~$
```

Будет запущен текстовый редактор gedit в фоновом режиме. Консоль при этом не будет заблокирована. Запущенные фоновые 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 идентификатором 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 (работающих или остановленных) на вашем терминале, используйте опцию aux.

Пример:

ps aux

```
dskulyabov@dskulyabov:~$ ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.2  0.3 72852 27360 ?        Ss   18:15   0:01 /usr/lib/systemd/systemd --switched-root --system --deserialize=36 rhgb
root         2  0.0  0.0      0     0 ?        S    18:15   0:00 [kthreadd]
root         3  0.0  0.0      0     0 ?        S    18:15   0:00 [pool_workqueue_release]
root         4  0.0  0.0      0     0 ?        I<   18:15   0:00 [kworker/R-rcu_g]
root         5  0.0  0.0      0     0 ?        I<   18:15   0:00 [kworker/R-rcu_p]
root         6  0.0  0.0      0     0 ?        I<   18:15   0:00 [kworker/R-slab_]
root         7  0.0  0.0      0     0 ?        I<   18:15   0:00 [kworker/R-netns]
root        10  0.0  0.0      0     0 ?        I<   18:15   0:00 [kworker/0:0H-events_highpri]
root        11  0.0  0.0      0     0 ?        I    18:15   0:00 [kworker/u4:0-events_unbound]
root        12  0.0  0.0      0     0 ?        I    18:15   0:00 [kworker/u4:1-ext4-rsv-conversion]
root        13  0.0  0.0      0     0 ?        I<   18:15   0:00 [kworker/R-mm_pe]
root        14  0.0  0.0      0     0 ?        I    18:15   0:00 [rcu_tasks_kthread]
root        15  0.0  0.0      0     0 ?        I    18:15   0:00 [rcu_tasks_rude_kthread]
root        16  0.0  0.0      0     0 ?        I    18:15   0:00 [rcu_tasks_trace_kthread]
root        17  0.0  0.0      0     0 ?        S    18:15   0:00 [ksoftirqd/0]
root        18  0.0  0.0      0     0 ?        I    18:15   0:00 [rcu_preempt]
root        19  0.0  0.0      0     0 ?        S    18:15   0:00 [migration/0]
root        20  0.0  0.0      0     0 ?        S    18:15   0:00 [idle_inject/0]
root        21  0.0  0.0      0     0 ?        S    18:15   0:00 [cpuhp/0]
root        22  0.0  0.0      0     0 ?        S    18:15   0:00 [cpuhp/1]
root        23  0.0  0.0      0     0 ?        S    18:15   0:00 [idle_inject/1]
root        24  0.1  0.0      0     0 ?        S    18:15   0:00 [migration/1]
root        25  0.0  0.0      0     0 ?        S    18:15   0:00 [ksoftirqd/1]
root        26  0.0  0.0      0     0 ?        I    18:15   0:00 [kworker/1:0-ata_sff]
root        27  0.0  0.0      0     0 ?        I<   18:15   0:00 [kworker/1:0H-kblockd]
root        29  0.1  0.0      0     0 ?        I    18:15   0:00 [kworker/u6:0-btrfs-endio-write]
```

Для запуска команды в фоновом режиме необходимо в конце командной строки указать 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 &

```
[1]+ Выход 1
dskulyabov@dskulyabov:~$ find /var/log -name "*.log" -print > l.log &
[1] 2565
dskulyabov@dskulyabov:~$ find: '/var/log/audit': Отказано в доступе
find: '/var/log/chrony': Отказано в доступе
find: '/var/log/ppp': Отказано в доступе
find: '/var/log/private': Отказано в доступе
find: '/var/log/samba': Отказано в доступе
find: '/var/log/sss': Отказано в доступе
[1]+ Выход 1
dskulyabov@dskulyabov:~$
```

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://midnightcommander.org/>.
4. NASM Assembly Language Tutorials. — 2021. — URL: <https://asmtutor.com/>.
5. 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>.
6. Robbins A. Bash Pocket Reference. — O'Reilly Media, 2016. — 156 с. — ISBN 978-1491941591.
7. The NASM documentation. — 2021. — URL: <https://www.nasm.us/docs.php>.
8. Zarrelli G. Mastering Bash. — Packt Publishing, 2017. — 502 с. — ISBN
9. Колдаев В. Д., Лупин С. А. Архитектура ЭВМ. — М. : Форум, 2018.
10. Куляс О. Л., Никитин К. А. Курс программирования на ASSEMBLER. — М. : Солон-Пресс, 2017.
11. Новожилов О. П. Архитектура ЭВМ и систем. — М. : Юрайт, 2016.
12. Расширенный ассемблер: NASM.— 2021.—URL: <https://www.opennet.ru/docs/RUS/nasm/>.
13. Робачевский А., Немнюгин С., Стесик О. Операционная система UNIX. — 2-е изд. — БХВПетербург, 2010. — 656 с. — ISBN 978-5-94157-538-1.
14. 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.:::