

Операционные системы

Лабораторная работа №7

Гульдяев Тихон Дмитриевич

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	13
4	Ответы на контрольные вопросы	14
	Список литературы	18

Список таблиц

Список иллюстраций

2.1	Запись файлов из etc в file.txt	6
2.2	Дописывание файлов из домашней директории в file.txt	6
2.3	Вывод и запись в conf.txt файлов с расширением .conf из file.txt	7
2.4	Поиск файлов в домашней директории которые начинаются с символа с	7
2.5	Страница имен файлов каталога /etc начинающихся с символа h	8
2.6	Команда для вывода (по странично) имен файлов каталога /etc начинающихся с символа h	8
2.7	Запуск в фоновом режиме процесса, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log	9
2.8	Удаление файл ~/logfile	9
2.9	Запуск из консоли в фоновом режиме редактора gedit	9
2.10	Определение идентификатора процесса gedit	10
2.11	Более простой способ определения идентификатора процесса gedit	10
2.12	Справка по kill и завершение процесса gedit	10
2.13	Выполняю команду df, предварительно прочитав man	10
2.14	Выполняю команду du, предварительно прочитав man	11
2.15	man по find и вывод всех имен директорий в домашнем каталоге	12

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. Выполнение лабораторной работы

Записываю в файл file.txt названия файлов, содержащихся в каталоге /etc. Проверяя сколько файлов в /etc и сколько стало в file.txt. (рис. 2.1).

```
guldyaev-tikhon@guldyaevtikhon:~$ sudo find /etc -maxdepth 1 -type f -exec basename {} \; | wc -l
97
guldyaev-tikhon@guldyaevtikhon:~$ sudo find /etc -maxdepth 1 -type f -exec basename {} \; > file.txt
guldyaev-tikhon@guldyaevtikhon:~$ wc -l file.txt
97 file.txt
```

Рис. 2.1: Запись файлов из etc в file.txt

Затем, я дописываю в этот же файл названия файлов, содержащихся в моем домашнем каталоге. Проверяя сколько файлов в домашнем каталоге и сколько стало в file.txt. (рис. 2.2).

```
guldyaev-tikhon@guldyaevtikhon:~$ sudo find ~ -maxdepth 1 -type f -exec basename {} \; | wc -l
24
guldyaev-tikhon@guldyaevtikhon:~$ sudo find ~ -maxdepth 1 -type f -exec basename {} \; >> file.txt
guldyaev-tikhon@guldyaevtikhon:~$ wc -l file.txt
121 file.txt
guldyaev-tikhon@guldyaevtikhon:~$
```

Рис. 2.2: Дописывание файлов из домашней директории в file.txt

Вывожу имена всех файлов из file.txt, имеющих расширение .conf, после чего записываю их в новый текстовый файл conf.txt и проверяю их количество. (рис. 2.3).

```

guldyaev-tikhon@guldyaevtikhon:~$ grep '\.conf$' file.txt
/etc/appstream.conf
/etc/fuse.conf
/etc/ucf.conf
/etc/popularity-contest.conf
/etc/gai.conf
/etc/mtools.conf
/etc/adduser.conf
/etc/hdparm.conf
/etc/e2scrub.conf
/etc/vconsole.conf
/etc/ca-certificates.conf
/etc/kernel-img.conf
/etc/logrotate.conf
/etc/sysctl.conf
/etc/rygel.conf
/etc/libaudit.conf
/etc/kernelevents.conf
/etc/ltrace.conf
/etc/usb_modeswitch.conf
/etc/ld.so.conf
/etc/mke2fs.conf
/etc/rsyslog.conf
/etc/host.conf
/etc/libao.conf
/etc/brltty.conf
/etc/debconf.conf
/etc/xattr.conf
/etc/fprintd.conf
/etc/pnm2ppa.conf
/etc/deluser.conf
/etc/nsswitch.conf
/etc/pam.conf
/etc/sensors3.conf
/etc/apg.conf
guldyaev-tikhon@guldyaevtikhon:~$ grep '\.conf$' file.txt > conf.txt
guldyaev-tikhon@guldyaevtikhon:~$ wc -l conf.txt
34 conf.txt
guldyaev-tikhon@guldyaevtikhon:~$

```

Рис. 2.3: Вывод и запись в conf.txt файлов с расширением .conf из file.txt

Определяю, какие файлы в моем домашнем каталоге имеют имена, начинавшиеся с символа с. На выбор предоставляю 2 команды. (рис. 2.4).

```

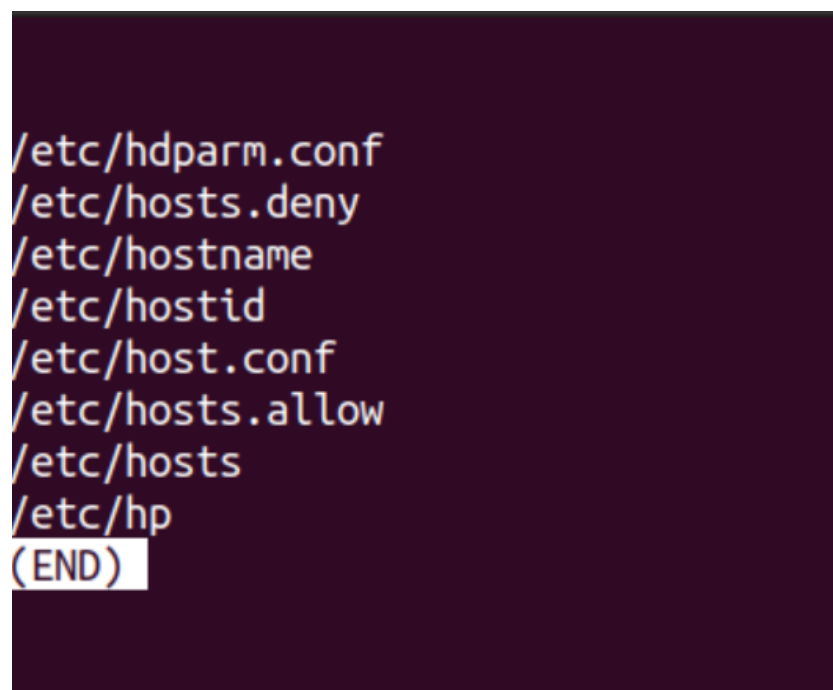
guldyaev-tikhon@guldyaevtikhon:~$ ls -p ~/ | grep -v / | grep ^c
change.c
conf.txt
guldyaev-tikhon@guldyaevtikhon:~$ find ~/ -maxdepth 1 -type f -name 'c*'
/home/guldyaev-tikhon/change.c
/home/guldyaev-tikhon/conf.txt
guldyaev-tikhon@guldyaevtikhon:~$

```

Рис. 2.4: Поиск файлов в домашней директории которые начинаются с символа с

Вывожу на экран (по странично) имена файлов из каталога /etc, начинающиеся с символа

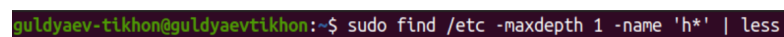
h.(рис. 2.5).



```
/etc/hdparm.conf
/etc/hosts.deny
/etc/hostname
/etc/hostid
/etc/host.conf
/etc/hosts.allow
/etc/hosts
/etc/hp
(END)
```

Рис. 2.5: Страница имен файлов каталога /etc начинающихся с символа h

Использую следующую команду.(рис. 2.6).



```
guldyaev-tikhon@guldyaevtikhon:~$ sudo find /etc -maxdepth 1 -name 'h*' | less
```

Рис. 2.6: Команда для вывода (по странично) имен файлов каталога /etc начинающихся с символа h

Запускаю в фоновом режиме процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log. И проверяю количество этих файлов(рис. 2.7).


```
guldyaev-tikhon@guldyaevtikhon:~$ sudo find / -name "log*" -type f -exec basename {} \; > logfile &
[1] 19611
guldyaev-tikhon@guldyaevtikhon:~$ find: '/run/user/1000/gvfs': Отказано в доступе
guldyaev-tikhon@guldyaevtikhon:~$

guldyaev-tikhon@guldyaevtikhon:~$ wc -l logfile
1072 logfile
guldyaev-tikhon@guldyaevtikhon:~$
```

Рис. 2.7: Запуск в фоновом режиме процесса, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log

Удаляю файл ~/logfile. (рис. 2.8).

```
guldyaev-tikhon@guldyaevtikhon:~$ rm logfile
```

Рис. 2.8: Удаление файл ~/logfile

Запускаю из консоли в фоновом режиме редактор gedit. (рис. 2.9).

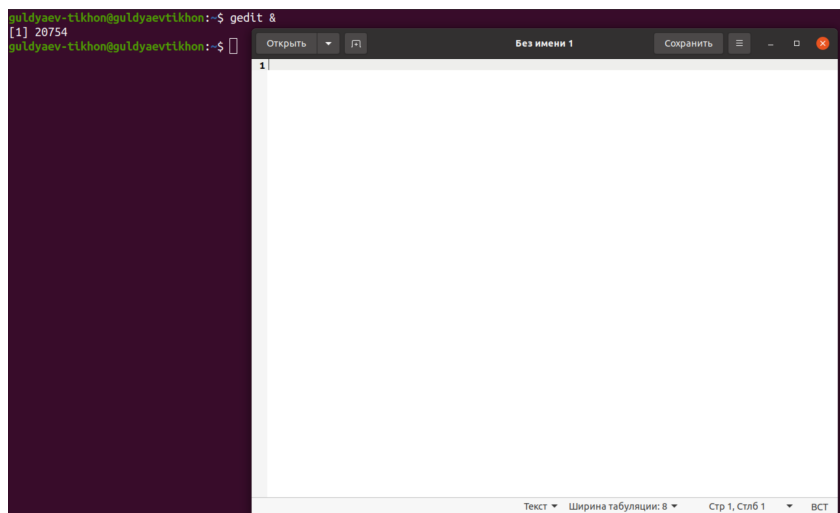


Рис. 2.9: Запуск из консоли в фоновом режиме редактора gedit

Определите идентификатор процесса gedit, используя команду ps, конвейер и фильтр grep. (рис. 2.10).

```
guldyaev-tikhon@guldyaevtikhon:~$ ps aux | grep gedit
guldyaev+ 20754  0.4  1.5 830956 60664 pts/0    Sl   20:24   0:00  gedit
guldyaev+ 20767  0.0  0.0 17564   724 pts/1    S+   20:25   0:00  grep --color=auto gedit
```

Рис. 2.10: Определение идентификатора процесса gedit

Более простой способ определения(рис. 2.11).

```
guldyaev-tikhon@guldyaevtikhon:~$ pgrep gedit
20754
guldyaev-tikhon@guldyaevtikhon:~$
```

Рис. 2.11: Более простой способ определения идентификатора процесса gedit

Читаю справку (man) команды kill, после чего использую её для завершения процесса gedit. (рис. 2.12).

```
guldyaev-tikhon@guldyaevtikhon:~$ man kill
guldyaev-tikhon@guldyaevtikhon:~$ kill 20754
guldyaev-tikhon@guldyaevtikhon:~$
```

Рис. 2.12: Справка по kill и завершение процесса gedit

Выполняю команду df, предварительно получив более подробную информацию с помощью команды man. (рис. 2.13).

```
guldyaev-tikhon@guldyaevtikhon:~$ man df
guldyaev-tikhon@guldyaevtikhon:~$ df
Файл. система 1K-блоков  Использовано  Доступно  Использовано%  Смонтировано в
udev          1968632      0      1968632      0% /dev
tmpfs         401824      1408      400416      1% /run
/dev/sda5     40454400    15222852  23144388    40% /
tmpfs         2009112      0      2009112      0% /dev/shm
tmpfs         5120         4        5116      1% /run/lock
tmpfs         2009112      0      2009112      0% /sys/fs/cgroup
/dev/loop0    128         128         0      100% /snap/bare/5
/dev/loop2    64896       64896      0      100% /snap/core20/1852
/dev/loop3    56960       56960      0      100% /snap/core18/2721
/dev/loop4    64896       64896      0      100% /snap/core20/1828
/dev/loop1    56960       56960      0      100% /snap/core18/2714
/dev/loop5    55552       55552      0      100% /snap/snap-store/558
/dev/loop6    37888       37888      0      100% /snap/gh/502
/dev/loop11   51072       51072      0      100% /snap/snapd/18596
/dev/loop9    358144     358144      0      100% /snap/gnome-3-38-2004/137
/dev/loop7    47104       47104      0      100% /snap/snap-store/638
/dev/loop8    51072       51072      0      100% /snap/snapd/18357
/dev/loop10   354688     354688      0      100% /snap/gnome-3-38-2004/119
/dev/loop12   93952       93952      0      100% /snap/gtk-common-themes/1535
/dev/sda1     523248      4      523244      1% /boot/efl
tmpfs         401820      28      401792      1% /run/user/1000
/dev/sr0      51716      51716      0      100% /media/guldyaev-tikhon/VBox_GAs_7.0.2
guldyaev-tikhon@guldyaevtikhon:~$
```

Рис. 2.13: Выполняю команду df, предварительно прочитав man

Выполняю команду `du`, предварительно получив более подробную информацию с помощью команды `man`. (рис. 2.14).

```
guldyaev-tikhon@guldyaev-tikhon:~$ man du
guldyaev-tikhon@guldyaev-tikhon:~$ du
4      ./cache/obexd
800    ./cache/krunner/qmlcache
804    ./cache/krunner
8      ./cache/ubuntu-report
14     ./cache/mozilla/firefox/tvlnhkr4.default-release/thumbnails
756    ./cache/mozilla/firefox/tvlnhkr4.default-release/cache2/defaults
112460 ./cache/mozilla/firefox/tvlnhkr4.default-release/cache2/entries
113336 ./cache/mozilla/firefox/tvlnhkr4.default-release/cache2
15424  ./cache/mozilla/firefox/tvlnhkr4.default-release/safebrowsing/google4
17436  ./cache/mozilla/firefox/tvlnhkr4.default-release/safebrowsing
22728  ./cache/mozilla/firefox/tvlnhkr4.default-release/startupCache
16     ./cache/mozilla/firefox/tvlnhkr4.default-release/settings/main/ns-language-packs/browser/newtab
20     ./cache/mozilla/firefox/tvlnhkr4.default-release/settings/main/ns-language-packs/browser
24     ./cache/mozilla/firefox/tvlnhkr4.default-release/settings/main/ns-language-packs
8      ./cache/mozilla/firefox/tvlnhkr4.default-release/settings/main/ns-images
36     ./cache/mozilla/firefox/tvlnhkr4.default-release/settings/main
40     ./cache/mozilla/firefox/tvlnhkr4.default-release/settings
153598 ./cache/mozilla/firefox/tvlnhkr4.default-release
4      ./cache/mozilla/firefox/guoyreka.default
153596 ./cache/mozilla/firefox
153600 ./cache/mozilla
4      ./cache/google-chrome/Default/Storage/ext/nmmhkkgccagldgltimedpiccngmteda/def/Cache/Cache_Data
8      ./cache/google-chrome/Default/Storage/ext/nmmhkkgccagldgltimedpiccngmteda/def/Cache
8      ./cache/google-chrome/Default/Storage/ext/nmmhkkgccagldgltimedpiccngmteda/def/Code Cache/wasn/index-dir
16     ./cache/google-chrome/Default/Storage/ext/nmmhkkgccagldgltimedpiccngmteda/def/Code Cache/wasn
8      ./cache/google-chrome/Default/Storage/ext/nmmhkkgccagldgltimedpiccngmteda/def/Code Cache/js/index-dir
16     ./cache/google-chrome/Default/Storage/ext/nmmhkkgccagldgltimedpiccngmteda/def/Code Cache/js
36     ./cache/google-chrome/Default/Storage/ext/nmmhkkgccagldgltimedpiccngmteda/def/Code Cache
48     ./cache/google-chrome/Default/Storage/ext/nmmhkkgccagldgltimedpiccngmteda/def
```

Рис. 2.14: Выполняю команду `du`, предварительно прочитав `man`

Воспользовавшись справкой команды `find`, вывожу имена всех директорий, имеющих в моем домашнем каталоге. (рис. 2.15).

```
guldyaev-tikhon@guldyaevtikhon:~$ man find
guldyaev-tikhon@guldyaevtikhon:~$ find ~ -maxdepth 1 -type d -exec basename {} \;
guldyaev-tikhon
.cache
play
Downloads
academic-laboratory-report-template-master
Видео
.thunderbird
Документы
.kde
.vscode
.mozilla
Шаблоны
snap
australia
study_2022-2023_os-intro
Зарядки
.gnupg
ski.places
src
test
Pictures
Desktop
Рабочий стол
Музыка
work
D04_LinuxMonitoring_v2.0-0
monthly
Изображения
labs_os
.config
.ssh
text_files
.pki
Общедоступные
.local
reports
guldyaev-tikhon@guldyaevtikhon:~$
```

Рис. 2.15: man по find и вывод всех имен директорий в домашнем каталоге

3. Выводы

Я ознакомился с инструментами поиска файлов и фильтрации текстовых данных. Приобрел практические навыки: по управлению процессами (и заданиями), по проверке использования диска и обслуживанию файловых систем.

4. Ответы на контрольные вопросы

1. Какие потоки ввода вывода вы знаете?

Потоки ввода/вывода, которые я знаю: стандартный ввод (stdin), стандартный вывод (stdout), стандартный вывод ошибок (stderr) и файловые потоки (например, чтение/запись в файлы).

2. Объясните разницу между операцией > и ».

Операция ">" используется для перенаправления вывода команды в файл, при этом существующий файл будет перезаписан. Операция ">" также используется для перенаправления вывода команды в файл, но при этом вывод добавляется в конец файла, не перезаписывая его содержимое.

3. Что такое конвейер?

Конвейер (pipeline) - это последовательность команд в командной оболочке, где вывод одной команды передается на вход следующей команде без сохранения на диск, что позволяет эффективно обрабатывать большие объемы данных путем их последовательной обработки.

4. Что такое процесс? Чем это понятие отличается от программы? Процесс - это экземпляр выполняющейся программы на компьютере, который имеет свое собственное адресное пространство, состояние и потоки выполнения. Программа - это набор инструкций, написанных на языке программирования, предназначенных для выполнения определенных задач на компьютере. Отличие между процессом и программой заключается в том, что процесс является экземпляром программы, который выполняется в оперативной памяти компьютера.

5. Что такое PID и GID?

PID (Process ID) - это уникальный числовой идентификатор, присвоенный операционной системой процессу, который позволяет идентифицировать и управлять процессами на компьютере. GID (Group ID) - это идентификатор группы, к которой принадлежит процесс в операционной системе, используется для управления доступом к ресурсам, которые могут быть общими для нескольких процессов, принадлежащих одной группе.

6. Что такое задачи и какая команда позволяет ими управлять?

Задачи - это процессы, которые выполняются в операционной системе. Команда, которая позволяет управлять задачами, в зависимости от операционной системы, может быть различной, например: "ps", "kill", "tasklist", "taskkill" и т. д.

7. Найдите информацию об утилитах top и htop. Каковы их функции?

Утилиты "top" и "htop" - это инструменты мониторинга процессов в операционной системе Linux/Unix. Они показывают информацию о процессах, такую как использование CPU, памяти, список процессов в реальном времени, и позволяют управлять процессами, такими как завершение или приостановка процессов.

8. Назовите и дайте характеристику команде поиска файлов. Приведите примеры использования этой команды.

Команда поиска файлов в Linux - "find". Она позволяет найти файлы и директории в файловой системе на основе различных критериев, таких как имя файла, размер, дата создания и другие. Некоторые основные опции команды "find" в Linux:

- -name: Искать файлы с определенным именем или шаблоном имени. Пример: `find /path/to/directory -name "*.txt"` - найдет все файлы с расширением ".txt" в указанной директории и ее поддиректориях.
- -size: Искать файлы по размеру. Пример: `find /path/to/directory -size +1M` - найдет все файлы размером больше 1 мегабайта в указанной директории и ее поддиректориях.

- `-mtime`: Искать файлы по времени последнего изменения. Пример: `find /path/to/directory -mtime -7` - найдет все файлы, измененные в течение последних 7 дней, в указанной директории и ее поддиректориях.
- `-type`: Искать файлы определенного типа (например, файлы, директории, символические ссылки и др.). Пример: `find /path/to/directory -type f` - найдет все обычные файлы в указанной директории и ее поддиректориях.

Это только некоторые из множества опций, которые можно использовать с командой “find” в Linux. Более подробную информацию о команде “find” и ее опциях можно найти в соответствующей документации.

9. Можно ли по контексту (содержанию) найти файл? Если да, то как? Да, в Linux можно найти файлы на основе их содержания с использованием команды “grep”. “grep” позволяет искать заданный текст или регулярное выражение в содержимом файлов. Пример использования: `grep “search_text” /path/to/file` - найдет все вхождения “search_text” в указанном файле.

10. Как определить объем свободной памяти на жёстком диске?

Для определения объема свободной памяти на жестком диске в Linux можно использовать команду “df”. Пример: `df -h` - отобразит информацию о доступном свободном пространстве на всех подключенных файловых системах в удобочитаемом формате.

11. Как определить объем вашего домашнего каталога?

Для определения объема вашего домашнего каталога в Linux можно использовать команду “du”. Пример: `du -sh /path/to/home_directory` - отобразит общий размер вашего домашнего каталога в удобочитаемом формате.

12. Как удалить зависший процесс?

Для удаления зависшего процесса в Linux можно использовать команду “kill”. Сначала необходимо определить идентификатор процесса (PID) зависшего процесса с помощью команды “ps” или “top”, затем выполнить команду “kill PID” (где PID - идентификатор

процесса). Если зависший процесс не реагирует на обычную команду “kill”, можно использовать “kill -9 PID” для принудительного завершения процесса.

Список литературы

<https://www.google.ru>

<https://chat.openai.com/chat>