

Отчёт по лабораторной работе №8

**Поиск файлов. Перенаправление ввода-вывода. Просмотр
запущенных процессов**

Нухова Камилла Руслановна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Вывод	18
4	Контрольные вопросы	19

Список иллюстраций

2.1	Запись в файл	7
2.2	Поиск расширения .conf	8
2.3	Поиск файлов	9
2.4	Поиск файлов	10
2.5	Фоновый запуск процесса	11
2.6	Фоновый запуск и завершение процесса	12
2.7	Справка по команде df	13
2.8	Запуск команды df	14
2.9	Справка по команде du	15
2.10	Запуск команды du	16
2.11	Поиск директорий	17

Список таблиц

1 Цель работы

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

2 Выполнение лабораторной работы

1 Включаем компьютер, и заходим в учетную запись.

2 Запишем в файл file.txt названия файлов, содержащихся в каталоге /etc. Допишем в этот же файл названия файлов, содержащихся в нашем домашнем каталоге.

```
krnuhova@krnuhova:~$ ls /etc/ > file.txt
krnuhova@krnuhova:~$ ls >> file.txt
krnuhova@krnuhova:~$ cat file.txt
abrt
adjtime
aliases
alsa
alternatives
anaconda
anthy-unicode.conf
asound.conf
audit
authselect
avahi
bash_completion.d
bashrc
bindresvport.blacklist
binfmt.d
bluetooth
brlapi.key
brltty
brltty.conf
ceph
chkconfig.d
chromium
chrony.conf
cifs-utils
colord
containers
```

Рис. 2.1: Запись в файл

3 Выведем имена всех файлов из file.txt, имеющих расширение .conf, после чего запишем их в новый текстовый файл conf.txt.

```
krnuhova@krnuhova:~$ grep .conf file.txt > conf.txt
krnuhova@krnuhova:~$ cat conf.txt
anthy-unicode.conf
asound.conf
brltty.conf
chkconfig.d
chrony.conf
dconf
dleyna-server-service.conf
dnsmasq.conf
dracut.conf
dracut.conf.d
fprind.conf
fuse.conf
host.conf
idmapd.conf
kdump.conf
krb5.conf
krb5.conf.d
ld.so.conf
ld.so.conf.d
libaudit.conf
locale.conf
logrotate.conf
makedumpfile.conf.sample
man_db.conf
mk2fs.conf
```

Рис. 2.2: Поиск расширения .conf

4 Определили, какие файлы в нашем домашнем каталоге имеют имена, начинавшиеся с символа с?


```
/home/krunhova/work/study/2024-2025/Операционные системы/os-intro/project-personal/stage3/report/bib/cite.bib
/home/krunhova/work/study/2024-2025/Операционные системы/os-intro/project-personal/stage3/report/pandoc/csl
/home/krunhova/work/study/2024-2025/Операционные системы/os-intro/project-personal/stage3/report/pandoc/filters/pa
ndocxnos/core.py
/home/krunhova/work/study/2024-2025/Операционные системы/os-intro/project-personal/stage4/report/bib/cite.bib
/home/krunhova/work/study/2024-2025/Операционные системы/os-intro/project-personal/stage4/report/pandoc/csl
/home/krunhova/work/study/2024-2025/Операционные системы/os-intro/project-personal/stage4/report/pandoc/filters/pa
ndocxnos/core.py
/home/krunhova/work/study/2024-2025/Операционные системы/os-intro/project-personal/stage5/report/bib/cite.bib
/home/krunhova/work/study/2024-2025/Операционные системы/os-intro/project-personal/stage5/report/pandoc/csl
/home/krunhova/work/study/2024-2025/Операционные системы/os-intro/project-personal/stage5/report/pandoc/filters/pa
ndocxnos/core.py
/home/krunhova/work/study/2024-2025/Операционные системы/os-intro/project-personal/stage6/report/bib/cite.bib
/home/krunhova/work/study/2024-2025/Операционные системы/os-intro/project-personal/stage6/report/pandoc/csl
/home/krunhova/work/study/2024-2025/Операционные системы/os-intro/project-personal/stage6/report/pandoc/filters/pa
ndocxnos/core.py
/home/krunhova/work/study/2024-2025/Операционные системы/os-intro/presentation/report/_resources/csl
/home/krunhova/work/study/2024-2025/Операционные системы/os-intro/presentation/report/bib/cite.bib
/home/krunhova/git-extended/.git/hooks/commit-msg.sample
/home/krunhova/git-extended/.git/objects/07/c4060b36e02f21d48091feb69990ff312c9030
/home/krunhova/git-extended/.git/objects/cb
/home/krunhova/git-extended/.git/config
/home/krunhova/conf.txt
krunhova@krunhova:~$
```

Рис. 2.3: Поиск файлов

5 Выведем на экран (постранично) имена файлов из каталога /etc, начинающи-
еся с символа h.

```
find /etc -name "h*" -print | less
```

```
krnuhova@krnuhova:~  
find: '/etc/libvirt': Отказано в доступе  
/etc/hp  
/etc/hp/hplip.conf  
/etc/httpd  
/etc/httpd/conf/httpd.conf  
/etc/libibverbs.d/hfi1verbs.driver  
/etc/libibverbs.d/hns.driver  
find: '/etc/lvm/archive': Отказано в доступе  
find: '/etc/lvm/backup': Отказано в доступе  
find: '/etc/lvm/cache': Отказано в доступе  
find: '/etc/lvm/devices'/etc/logrotate.d/httpd  
: Отказано в доступе  
find: '/etc/nftables': Отказано в доступе  
find: '/etc/openvpn/client': Отказано в доступе  
/etc/nvme/hostnqn  
/etc/nvme/hostid  
find: '/etc/openvpn/server': Отказано в доступе  
find: '/etc/polkit-1/localauthority': Отказано в доступе  
find: '/etc/polkit-1/rules.d': Отказано в доступе  
find: '/etc/sos/cleaner': Отказано в доступе  
/etc/sane.d/dll.d/hpaio  
/etc/sane.d/hp.conf  
/etc/sane.d/hp3900.conf  
/etc/sane.d/hp4200.conf  
/etc/sane.d/hp5400.conf  
/etc/sane.d/hpsj5s.conf  
/etc/sane.d/hs2p.conf  
find: '/etc/ssh/ssh_config.d': Отказано в доступе  
find: '/etc/sss': Отказано в доступе  
find: '/etc/sudoers.d': Отказано в доступе  
/etc/sysconfig/htcacheclean  
/etc/systemd/system/httpd.service.d  
/etc/udev/hwdb.d  
/etc/udev/hwdb.bin  
/etc/host.conf  
/etc/hosts  
/etc/hostname  
(END)
```

Рис. 2.4: Поиск файлов

- 6 Запустили в фоновом режиме процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log. Процесс выполнен
- 7 Удалили файл ~/logfile. Но сначала убили процесс в нем.

```
krnuhova@krnuhova:~$ find /etc -name "ht*" -print | less
krnuhova@krnuhova:~$
krnuhova@krnuhova:~$
krnuhova@krnuhova:~$ find ~ -name "log*" > logfile &
[1] 12236
krnuhova@krnuhova:~$
[1]+  Завершён      find ~ -name "log*" > logfile
krnuhova@krnuhova:~$ rm logfile
krnuhova@krnuhova:~$
```

Рис. 2.5: Фоновый запуск процесса

- 8 Запустили из консоли в фоновом режиме редактор gedit.
- 9 Определили идентификатор процесса gedit, используя команду ps, конвейер и фильтр grep
- 10 Прочитали справку (man) команды kill, после чего используйте её для завершения процесса gedit.

```

krnuhova@krnuhova:~$
krnuhova@krnuhova:~$ find /etc -name "h*" -print | less
krnuhova@krnuhova:~$
krnuhova@krnuhova:~$
krnuhova@krnuhova:~$ find ~ -name "log*" > logfile &
[1] 12236
krnuhova@krnuhova:~$
[1]+  Завершён      find ~ -name "log*" > logfile
krnuhova@krnuhova:~$ rm logfile
krnuhova@krnuhova:~$
krnuhova@krnuhova:~$
krnuhova@krnuhova:~$ gedit &
[1] 12257
krnuhova@krnuhova:~$ ps | grep gedit
  12257 pts/0    00:00:00 gedit
krnuhova@krnuhova:~$ kill 12257
krnuhova@krnuhova:~$
[1]+  Завершено    gedit
krnuhova@krnuhova:~$ █

```

Рис. 2.6: Фоновый запуск и завершение процесса

11 Выполним команды `df` и `du`, предварительно получив более подробную информацию об этих командах, с помощью команды `man`.

```
krnuhova@krnuhova:~ -- man df

df(1)                                     Команды пользователя                                     df(1)

ИМЯ
df – вывести информацию об использовании пространства файловой системы

СИНТАКСИС
df [ПАРАМЕТР]... [ФАЙЛ]...

ОПИСАНИЕ
Данная страница руководства описывает версию df от GNU. df отображает объём доступного пространства в
каждой файловой системе, содержащей файлы, имена которых переданы в качестве аргументов. Если имена файлов
не указаны, будет отображено доступное пространство во всех смонтированных в настоящий момент файловых
системах. По умолчанию объём пространства отображается в блоках размером 1K, однако если задана переменная
среды POSIXLY_CORRECT, будут использоваться блоки размером 512 байт.

Если аргумент представляет собой абсолютное имя файла устройства, на котором расположена смонтированная
файловая система, то df отобразит информацию о пространстве, доступном в этой файловой системе, а не в
файловой системе, содержащей файл устройства. Данная версия df не может отображать доступное пространство
в размонтированных файловых системах, поскольку в большинстве случаев это требует глубокого понимания
структур файловой системы и ухудшает переносимость программы.

ПАРАМЕТРЫ
Отобразить информацию о каждой файловой системе, содержащей ФАЙЛЫ, или обо всех файловых системах (по
умолчанию).

Аргументы, обязательные для длинных параметров, обязательны и для коротких.

-a, --all
    включить информацию о псевдо-, повторяющихся и недоступных файловых системах

-B, --block-size=РАЗМЕР
    привести размеры к величине РАЗМЕР перед выводом; например, «-BM» выводит размеры в единицах
    измерения, кратных 1 048 576 байт; см. формат РАЗМЕРА ниже

--direct
    отобразить статистику о файле, а не точке монтирования

Manual page df(1) line 1 (press h for help or q to quit)
```

Рис. 2.7: Справка по команде df

```
krnuhova@krnuhova:~ — man du

DU(1)                                     Команды пользователя                                     DU(1)

ИМЯ
du — оценить используемое файлами пространство

СИНТАКСИС
du [ПАРАМЕТР]... [ФАЙЛ]...
du [ПАРАМЕТР]... --files0-from=F

ОПИСАНИЕ
Вывести сводную информацию об использовании устройств набором ФАЙЛов, выполнять рекурсивно для каталогов.

Аргументы, обязательные для длинных параметров, обязательны и для коротких.

-b, --null
    завершать каждую выводимую строку символом конца строки NUL вместо перевода на новую строку

-a, --all
    выводить результаты подсчёта для всех файлов, а не только для каталогов

--apparent-size
    выводить действительные размеры вместо занимаемого пространства на устройстве; как правило,
    действительный размер меньше занимаемого места, но он может быть больше из-за «дыр» в
    («разрежённых») файлах, внутренней фрагментации, блоков косвенной адресации (indirect blocks) и
    тому подобного

-B, --block-size=РАЗМЕР
    привести размеры к величине РАЗМЕР перед выводом; например, «-BM» выводит размеры в единицах
    измерения, кратных 1 048 576 байт; см. формат РАЗМЕРа ниже

-b, --bytes
    то же, что и «--apparent-size --block-size=1»

-c, --total
    подвести общий итог

-D, --dereference-args
Manual page du(1) line 1 (press h for help or q to quit)
```

Рис. 2.8: Запуск команды df

```
4      ./git-extended/.git/objects/35
4      ./git-extended/.git/objects/b9
4      ./git-extended/.git/objects/5f
76     ./git-extended/.git/objects
8      ./git-extended/.git/logs/refs/heads
8      ./git-extended/.git/logs/refs/remotes/origin
8      ./git-extended/.git/logs/refs/remotes
16     ./git-extended/.git/logs/refs
20     ./git-extended/.git/logs
212    ./git-extended/.git
220    ./git-extended
0      ./monthly
0      ./reports/monthly/monthly
0      ./reports/monthly
0      ./reports
4      ./ski.plases/equipment
0      ./ski.plases/plans
4      ./ski.plases
0      ./australia
0      ./play/games/play
0      ./play/games
0      ./play
456560 .
krnuhova@krnuhova:~$
```

Рис. 2.9: Справка по команде du

```
/home/krunhova/git-extended/.git/objects/2d
/home/krunhova/git-extended/.git/objects/61
/home/krunhova/git-extended/.git/objects/70
/home/krunhova/git-extended/.git/objects/6e
/home/krunhova/git-extended/.git/objects/35
/home/krunhova/git-extended/.git/objects/b9
/home/krunhova/git-extended/.git/objects/5f
/home/krunhova/git-extended/.git/logs
/home/krunhova/git-extended/.git/logs/refs
/home/krunhova/git-extended/.git/logs/refs/heads
/home/krunhova/git-extended/.git/logs/refs/remotes
/home/krunhova/git-extended/.git/logs/refs/remotes/origin
/home/krunhova/monthly
/home/krunhova/reports
/home/krunhova/reports/monthly
/home/krunhova/reports/monthly/monthly
/home/krunhova/ski.places
/home/krunhova/ski.places/equipment
/home/krunhova/ski.places/plans
/home/krunhova/australia
/home/krunhova/play
/home/krunhova/play/games
/home/krunhova/play/games/play
krunhova@krunhova:~$
```

Рис. 2.10: Запуск команды du

12 Воспользовавшись справкой команды find, вывести имена всех директорий, имеющих в нашем домашнем каталоге.

```
find ~ -type d
```



```
/home/krunhova/git-extended/.git/objects/2d
/home/krunhova/git-extended/.git/objects/61
/home/krunhova/git-extended/.git/objects/70
/home/krunhova/git-extended/.git/objects/6e
/home/krunhova/git-extended/.git/objects/35
/home/krunhova/git-extended/.git/objects/b9
/home/krunhova/git-extended/.git/objects/5f
/home/krunhova/git-extended/.git/logs
/home/krunhova/git-extended/.git/logs/refs
/home/krunhova/git-extended/.git/logs/refs/heads
/home/krunhova/git-extended/.git/logs/refs/remotes
/home/krunhova/git-extended/.git/logs/refs/remotes/origin
/home/krunhova/monthly
/home/krunhova/reports
/home/krunhova/reports/monthly
/home/krunhova/reports/monthly/monthly
/home/krunhova/ski.places
/home/krunhova/ski.places/equipment
/home/krunhova/ski.places/plans
/home/krunhova/australia
/home/krunhova/play
/home/krunhova/play/games
/home/krunhova/play/games/play
krunhova@krunhova:~$
```

Рис. 2.11: Поиск директорий

3 Вывод

В данной работе мы ознакомились с инструментами поиска файлов и фильтрации текстовых данных. А также приобрели практические навыки по управлению процессами.

4 Контрольные вопросы

1. Какие потоки ввода вывода вы знаете? Ответ:
 - a) `stdin` — стандартный поток ввода (клавиатура),
 - b) `stdout` — стандартный поток вывода (консоль),
 - c) `stderr` — стандартный поток вывод сообщений об ошибках на экран
2. Объясните разницу между операцией `>` и `>>` Ответ: Разница заключается в том, что Символ `>` используется для переназначения стандартного ввода команды, а символ `>>` используется для присоединения данных в конец файла стандартного вывода команды.
3. Что такое конвейер? Ответ: Конвейер – это способ связи между двумя программами. Например: конвейер `pipe` служит для объединения простых команд или утилит в цепочки, в которых результат работы предыдущей команды передается последующей. Синтаксис у конвейера следующий:
`команда1 | команда 2`
4. Что такое процесс? Чем это понятие отличается от программы? Ответ: Процесс - это программа, которая выполняется в отдельном виртуальном адресном пространстве независимо от других программ или их пользованию по необходимости.

5. Что такое PID и GID? Ответ: Во первых id — UNIX-утилита, выводящая информацию об указанном пользователе USERNAME или текущем пользователе, который запустил данную команду и не указал явно имя пользователя.
- 1) GID – (Group ID) - идентификатор группы
- 2) UID – (User ID) - идентификатор группы Обычно UID является — положительным целым числом в диапазоне от 0 до 65535, по которому в системе однозначно отслеживаются действия пользователя
6. Что такое задачи и какая команда позволяет ими управлять? Ответ: Запущенные фоновые программы называются задачами(процессами) (jobs). Ими можно управлять с помощью команды jobs, которая выводит список запущенных в данный момент процессов. Для завершения процесса необходимо выполнить команду : kill % номер задачи
7. Найдите информацию об утилитах top и htop. Каковы их функции? Ответ: Top это консольная команда, которая выводит список работающих в системе процессов и информации о них. По умолчанию она в реальном времени сортирует их по нагрузке на процессор. Htop же является альтернативой программе top она предназначена для вывода на терминал списка запущенных процессов и информации о них.
8. Назовите и дайте характеристику команде поиска файлов. Приведите примеры использования этой команды. Ответ: Команда find используется для поиска и отображения имен файлов, соответствующих заданной строке символов. Синтаксис: find trek [-options] Пример: Задача - Вывести на экран имена файлов из каталога /etc и его подкаталогов, Заканчивающихся на k:
find ~ -name "*k" -print
9. Можно ли по контексту (содержанию) найти файл? Если да, то как? Ответ: Можно, команда ggrep способна обрабатывать вывод других файлов. Для этого надо использовать конвейер, связав вывод команды с вводом ggrep.

Пример: Задача - показать строки в каталоге /dreams с именами начинающимися на t, в которых есть фраза: I like of Operating systems grep I like of Operating systems t*

10. Как определить объем свободной памяти на жёстком диске? Ответ: Команда df показывает размер каждого смонтированного раздела диска. Например команда: df -h
11. Как определить объем вашего домашнего каталога? Ответ: Команда du показывает число килобайт, используемое каждым файлом или каталогом. Например команда: du -sh
12. Как удалить зависший процесс? Ответ: Перед тем, как выполнить остановку процесса, нужно определить его PID. Когда известен PID , мы можем убить его командой kill. Команда kill принимает в качестве параметра PID процесса. PID можно узнать с помощью команд ps, grep, top или htop