

Лабораторная работа №8

Операционные системы

Тойчубекова Асель Нурлановна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	8
4	Выполнение лабораторной работы	10
4.1	Ответы на вопросы	19
5	Выводы	21

Список иллюстраций

4.1	Запись названий файлов	10
4.2	Запись названий файлов из корневого каталога	10
4.3	Содержимое файла file.txt	11
4.4	Сортировка и вывод названий файлов с расширением .conf	11
4.5	Редактирование файла conf.txt	12
4.6	Поиск файлов, начинающихся с определенного элемента	12
4.7	Поиск файлов, начинающихся с определенного элемента	12
4.8	Поиск файлов, начинающихся с определенного элемента	13
4.9	Поиск файла по заданному символу	13
4.10	Запуск процесса в фоновом режиме	13
4.11	Удаление файла	14
4.12	Запуск процесса в фоновом режиме	14
4.13	Запуск процесса в фоновом режиме	14
4.14	Идентификатор процесса gedit	14
4.15	Идентификатор процесса gedit	15
4.16	Справка команды kill	15
4.17	Завершение процесса gedit	15
4.18	Справка команды df	16
4.19	Справка команды du	16
4.20	Выполнение команды df	17
4.21	Выполнение команды du	17
4.22	Выполнение команды du	17
4.23	Справка команды find	18
4.24	Вывод имен директорий	18
4.25	Вывод имен директорий	18

Список таблиц

1 Цель работы

Целью данной лабораторной работы является ознакомление с инструментами поиска файлов и фильтрации текстовых данных. Также приобретение практических навыков: по управлению процессами (и заданиями), по проверке использования диска и обслуживанию файловых систем.

2 Задание

Выполнить все указания к лабораторной работе:

1. Осуществите вход в систему, используя соответствующее имя пользователя.
2. Запишите в файл `file.txt` названия файлов, содержащихся в каталоге `/etc`. Допишите в этот же файл названия файлов, содержащихся в вашем домашнем каталоге.
3. Выведите имена всех файлов из `file.txt`, имеющих расширение `.conf`, после чего запишите их в новый текстовый файл `conf.txt`.
4. Определите, какие файлы в вашем домашнем каталоге имеют имена, начинающиеся с символа `s`? Предложите несколько вариантов, как это сделать.
5. Выведите на экран (по странично) имена файлов из каталога `/etc`, начинающиеся с символа `h`.
6. Запустите в фоновом режиме процесс, который будет записывать в файл `~/logfile` файлы, имена которых начинаются с `log`.
7. Удалите файл `~/logfile`.
8. Запустите из консоли в фоновом режиме редактор `gedit`.
9. Определите идентификатор процесса `gedit`, используя команду `ps`, конвейер и фильтр `grep`. Как ещё можно определить идентификатор процесса?
10. Прочтите справку (`man`) команды `kill`, после чего используйте её для завершения процесса `gedit`.
11. Выполните команды `df` и `du`, предварительно получив более подробную информацию об этих командах, с помощью команды `man`.
12. Воспользовавшись справкой команды `find`, выведите имена всех директо-

рий, имеющихся в вашем домашнем каталоге.

3 Теоретическое введение

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 **>**, **>>**, **<**, **<<**.

Конвейер (**pipe**) служит для объединения простых команд или утилит в цепочки, в которых результат работы предыдущей команды передается поседующей. Для этого используется символ **|**.

Команда find используется для поиска и отображения на экран имён файлов, соответствующих заданной строке символов. Путь определяет каталог, начиная с которого по всем подкаталогам будет вестись поиск.

Найти в текстовом файле указанную строку символов позволяет **команда grep**.

Команда df показывает размер каждого смонтированного раздела диска.

Команда du показывает число килобайт, используемое каждым файлом или каталогом.

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

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**

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.

Команда ps используется для получения информации о процессах.

4 Выполнение лабораторной работы

Для начала осуществляю вход в систему, используя соответствующее имя пользователя.

Записываю в файл file.txt названия файлов, содержащихся в каталоге /etc. С помощью утилиты head, которая выводит первые 10 строчек содержания файла на экран проверяю, что все имена файлов были записаны. (рис. 4.1).

```
[antoyjchubekova@antoyjchubekova ~]$ touch file.txt
[antoyjchubekova@antoyjchubekova ~]$ sudo ls -lR /etc > file.txt
[antoyjchubekova@antoyjchubekova ~]$ head file.txt
/etc:
итого 1356
drwxr-xr-x. 1 root root    126 ноя  1 04:07 abrt
-rw-r--r--. 1 root root     16 фев 20 14:12 adjtime
-rw-r--r--. 1 root root  1529 июл 25  2023 aliases
drwxr-xr-x. 1 root root     70 янв 29 03:00 alsa
drwxr-xr-x. 1 root root   1400 мар  7 12:16 alternatives
drwxr-xr-x. 1 root root     56 ноя  1 04:07 anaconda
-rw-r--r--. 1 root root    541 июл 19  2023 anacrontab
-rw-r--r--. 1 root root    833 фев 10  2023 appstream.conf
[antoyjchubekova@antoyjchubekova ~]$
```

Рис. 4.1: Запись названий файлов

Затем добавляю в этот же файл названия файлов, содержащихся в моем домашнем каталоге (рис. 4.2). Используя команду cat вывожу на экран содержимое файла file.txt и вижу, что все названия файлов из домашнего каталога были записаны в файл (рис. 4.3).

```
[antoyjchubekova@antoyjchubekova ~]$ sudo ls -lR ~/ > file.txt
[antoyjchubekova@antoyjchubekova ~]$ cat file.txt
```

Рис. 4.2: Запись названий файлов из корневого каталога

```

/home/antoychubekova/work/study/2023-2024/Операционные системы/report/pandoc/filters/pandocxnos:
итого 64
-rw-r--r--. 1 antoychubekova antoychubekova 47950 фев 25 20:28 core.py
-rw-r--r--. 1 antoychubekova antoychubekova 121 фев 25 20:28 __init__.py
-rw-r--r--. 1 antoychubekova antoychubekova 1502 фев 25 20:28 main.py
-rw-r--r--. 1 antoychubekova antoychubekova 6855 фев 25 20:28 pandocattributes.py

/home/antoychubekova/Видео:
итого 0

/home/antoychubekova/Документы:
итого 0

/home/antoychubekova/Загрузки:
итого 56672
-rw-r--r--. 1 antoychubekova antoychubekova 362 мар 6 16:20 git-extended-main(1).zip
-rw-r--r--. 1 antoychubekova antoychubekova 362 мар 6 16:20 git-extended-main.zip
-rw-r--r--. 1 antoychubekova antoychubekova 22009044 мар 2 17:22 hugo_extended_0.123.7_linux-amd64.tar.gz
-rw-r--r--. 1 antoychubekova antoychubekova 11347 мар 1 19:15 LICENSE
-rw-r--r--. 1 antoychubekova antoychubekova 28685814 фев 23 00:32 pandoc-3.1.3-linux-amd64.tar.gz
-rw-r--r--. 1 antoychubekova antoychubekova 43256 июн 10 2023 pandoc-crossref.1
-rw-r--r--. 1 antoychubekova antoychubekova 7249656 фев 22 23:52 pandoc-crossref-Linux.tar.xz
-rw-r--r--. 1 antoychubekova antoychubekova 12891 мар 1 19:15 README.md

/home/antoychubekova/Изображения:
итого 0

/home/antoychubekova/Музыка:
итого 0

/home/antoychubekova/Общедоступные:
итого 0

/home/antoychubekova/Рабочий стол:
итого 0

/home/antoychubekova/Шаблоны:
итого 0
[antoychubekova@antoychubekova ~]$

```

Рис. 4.3: Содержимое файла file.txt

Вывожу имена всех файлов из file.txt, имеющих расширение .conf, с помощью утилиты grep. (рис. 4.4).

```

[antoychubekova@antoychubekova ~]$ grep .conf file.txt
drwxr-xr-x. 1 antoychubekova antoychubekova 16 мар 2 21:24 config
-rw-r--r--. 1 antoychubekova antoychubekova 227 мар 16 15:05 jsconfig.json
/home/antoychubekova/work/blog/config:
/home/antoychubekova/work/blog/config/_default:
drwxr-xr-x. 1 antoychubekova antoychubekova 96 мар 2 21:24 conference-paper
/home/antoychubekova/work/blog/content/publication/conference-paper:
-rw-r--r--. 1 antoychubekova antoychubekova 15043 мар 2 21:24 conference-paper.pdf
-rw-r--r--. 1 antoychubekova antoychubekova 653 мар 16 15:05 mathjax-config.js
drwxr-xr-x. 1 antoychubekova antoychubekova 628 мар 2 22:15 conference-paper
/home/antoychubekova/work/blog/public/publication/conference-paper:
-rw-r--r--. 1 antoychubekova antoychubekova 15043 мар 16 14:54 conference-paper.pdf
drwxr-xr-x. 1 antoychubekova antoychubekova 46 мар 2 22:15 paper-conference
/home/antoychubekova/work/blog/public/publication-type/paper-conference:
/home/antoychubekova/work/blog/public/publication-type/paper-conference/page:
/home/antoychubekova/work/blog/public/publication-type/paper-conference/page/1:
drwxr-xr-x. 1 antoychubekova antoychubekova 1016 мар 2 21:29 conference-paper
/home/antoychubekova/work/blog/resources/_gen/images/publication/conference-paper:
drwxr-xr-x. 1 antoychubekova antoychubekova 24 мар 7 10:52 config
/home/antoychubekova/work/study/2023-2024/Операционные системы/os-intro/config:
/home/antoychubekova/work/study/2023-2024/Операционные системы/os-intro/config/course:
/home/antoychubekova/work/study/2023-2024/Операционные системы/os-intro/config/script:
drwxr-xr-x. 1 antoychubekova antoychubekova 24 фев 25 20:22 config
/home/antoychubekova/work/study/2023-2024/Операционные системы/os-intro-/config:
/home/antoychubekova/work/study/2023-2024/Операционные системы/os-intro-/config/course:
/home/antoychubekova/work/study/2023-2024/Операционные системы/os-intro-/config/script:
[antoychubekova@antoychubekova ~]$

```

Рис. 4.4: Сортировка и вывод названий файлов с расширением .conf

Далее записываю их в новый текстовый файл conf.txt, используя символ “>”.
С помощью команды head проверяю их наличия в файле. (рис. 4.5).

```
[antoychubekova@antoychubekova ~]$ grep .conf file.txt > conf.txt
[antoychubekova@antoychubekova ~]$ head conf.txt
drwxr-xr-x. 1 antoychubekova antoychubekova 16 map 2 21:24 config
-rw-r--r--. 1 antoychubekova antoychubekova 227 map 16 15:05 jsconfig.json
/home/antoychubekova/work/blog/config:
/home/antoychubekova/work/blog/config/_default:
drwxr-xr-x. 1 antoychubekova antoychubekova 96 map 2 21:24 conference-paper
/home/antoychubekova/work/blog/content/publication/conference-paper:
-rw-r--r--. 1 antoychubekova antoychubekova 15043 map 2 21:24 conference-paper.pdf
-rw-r--r--. 1 antoychubekova antoychubekova 653 map 16 15:05 mathjax-config.js
drwxr-xr-x. 1 antoychubekova antoychubekova 628 map 2 22:15 conference-paper
/home/antoychubekova/work/blog/public/publication/conference-paper:
[antoychubekova@antoychubekova ~]$
```

Рис. 4.5: Редактирование файла conf.txt

Определяю, какие файлы в домашнем каталоге начинаются с символа “с” с помощью утилиты find, прописываю ей в аргументы домашнюю директорию, также опции -name и -print для того, чтобы команда искала файлы по имени и выводила их на экран. (рис. 4.6 и рис. 4.7)

```
[antoychubekova@antoychubekova ~]$ find ~ -name "с*" -print
```

Рис. 4.6: Поиск файлов, начинающихся с определенного элемента

```
/home/antoychubekova/.local/share/pnpm/store/v3/files/fb/c70aee344045e6243303221be6f185c34a88724333a6d1f450ff9c6057
74c2ee61ac23179706aed2e2949f8b19dd98c25e72634790e47d63e821596a24-index.json
/home/antoychubekova/.local/share/pnpm/store/v3/files/34/cf3f3fd9f75e35e12199f594b86415a0024ce5114178d6855e0103f467
f31be0aadaa9017f483b89914314b1d51968e2dab37aa6f4b0e96bb9a3b2dddba-index.json
/home/antoychubekova/.local/share/pnpm/store/v3/files/34/c89327a9a369a48f5f772b688d67f4978f550c7f5d6b78120b9349d977
9966088195fb587c88eaa396ae5e52dab02abc99795c9b07a03263f4aaf7dd2
/home/antoychubekova/.local/share/pnpm/store/v3/files/34/ce995f086cf2435826183ba1e0cb14dda2e4980de00b1037e625c061fc
c925dda32150772490b382011f505dcb7de2c6b2c89f2627470d0fa23ca4dedb
/home/antoychubekova/.local/share/pnpm/store/v3/files/50/c2ba1f00ea760c542672c39099f49cbcd587fc16070bd7dbbabe6bfd00
3861845895b419baa3bcf26c303a98819cf9abad1c91d6940f6658e65735e65b5
/home/antoychubekova/.local/share/pnpm/store/v3/files/96/c599b361c22575426db0cf98fc89b8cf47c27635dc4240c7ed7ef65560
bbf77eabff2fae031e584ee9ce1deaf54c0886925168a1c5da332873258ccb0657
/home/antoychubekova/.local/share/pnpm/store/v3/files/96/cdb14a708a5582fb4864ae12dca3f894685fdf7350b08e82b172ea50c2
37c584688559f294e81398d6ed96d6d0c7fe92dc0e3089b073a9496bef4f36e29
/home/antoychubekova/.local/share/pnpm/store/v3/files/96/c03de2fe6884e1045ea60faadaab9144d0bc4fba76b192336c6e886b1
c08cfb0173d5229c4fc5abb513baf4528a59ceb34278c83809bcf701ac22b23c50
/home/antoychubekova/.local/share/pnpm/store/v3/files/7c/c422d5a25c7d8e8ecff78dfa69d1257a64b97f376b4c3199491cb2c528
4af999b2ca60f57f723d44c20a6b47371a32ea1f6e69499f8d3e24c2cc21865b3
/home/antoychubekova/.local/share/pnpm/store/v3/files/ea/c08a474742c09923d38d77705e6501ec174859d5c88805332de032f9224
6c94419f517cea43d8bb4674055168ba6a3c2f4730bd0ed2aef0b8d66fb146e0
/home/antoychubekova/.local/share/pnpm/store/v3/files/ea/cd63b05dee388d71af71178bd5e9f70e2c2662d1940cdbf5aea4b11b16
ed597641599dfc1ab71f69a3137a78b2d009a808c5e74268610fecbb1061494c1
/home/antoychubekova/.local/share/pnpm/store/v3/files/ea/c5fc976eed27e6228274e89e1688641a4e36ec7f1d9743b2fd4d102e8
75b4baf184f08e09efe25125ad61d62ebfe1f46986515d6887c62cc52c5ff6447
/home/antoychubekova/.local/share/pnpm/store/v3/files/c8
/home/antoychubekova/.local/share/pnpm/store/v3/files/26/cf29e7076256ba5ec3cc3a4fa82061ec4e0b34e0991eac5126c7f5a1ca
3f8c4f4322c43e3eda41e7b05b0c4d59e9e07be01cbf2eb48b7498f5b58cb86
/home/antoychubekova/.local/share/pnpm/store/v3/files/fa/c2825be1dc5f1e0cf89c2a8215f11d45ec081b8d2507fd40199d5df0cf
eb7e4aedba6067d02204712e310f79acb83664a162feade7c4254d4771fa6996
/home/antoychubekova/.local/share/pnpm/store/v3/files/fa/c7ab0fc3d73c16e1c384510829ecfdd6fcd0c200f9e58c55e1eaa2826
38064a98f7e7d642899ef13d64c8775913ea461d72241e6b7a8bf1deddee06f5
/home/antoychubekova/.local/share/pnpm/store/v3/files/86/cad360013bf350477fe4b26e9ced8f448d8f924d63ff666a52539720c8
```

Рис. 4.7: Поиск файлов, начинающихся с определенного элемента

Второй способ заключается в использовании утилиты `ls -lR` и использовать `grep`, чтобы найти элементы с первым символом `c`. Однако этот способ не работает для поиска файлов из подкаталогов каталога. (рис. 4.8).

```
[antoychubekova@antoychubekova ~]$ ls -lR | grep c*
-rw-r--r--. 1 antoychubekova antoychubekova  2084 мар 30 03:54 conf.txt
```

Рис. 4.8: Поиск файлов, начинающихся с определенного элемента

С помощью утилиты `find` выведу на экран имена файлов из каталога `etc`, начинающиеся с символом `h`. (рис. 4.9).

```
[antoychubekova@antoychubekova ~]$ sudo find /etc -name "h*" -print
[sudo] пароль для antoychubekova:
/etc/avahi/hosts
/etc/firewalld/helpers
/etc/libibverbs.d/hfi1verbs.driver
/etc/libibverbs.d/hns.driver
/etc/systemd/homed.conf
/etc/udev/hwdb.d
/etc/udev/hwdb.bin
/etc/host.conf
/etc/hosts
/etc/hostname
/etc/mercurial/hgrc.d
[antoychubekova@antoychubekova ~]$
```

Рис. 4.9: Поиск файла по заданному символу

Запускаю в фоновом режиме процесс, который будет записывать в файл `~/logfile` файлы, имена которых начинаются с `log`, с помощью утилиты `find`, амперсанта и символа перенаправления. (рис. 4.10).

```
[antoychubekova@antoychubekova ~]$ find ~ -name "log*" -print > logfile &
[1] 3511
[antoychubekova@antoychubekova ~]$ head logfile
/home/antoychubekova/.mozilla/firefox/rq0b6ssr.default-release/logins-backup.json
/home/antoychubekova/.mozilla/firefox/rq0b6ssr.default-release/logins.json
/home/antoychubekova/.cache/pnpm/metadata/registry.npmjs.org/log-symbols.json
/home/antoychubekova/.local/share/pnpm/keyrings/login.keyring
/home/antoychubekova/.local/share/pnpm/global/5/.pnpm/commitizen@4.3.0_@types+node@20.11.24_typescript@5.3.3/node_modules/commitizen/dist/git/log.js
/home/antoychubekova/.local/share/pnpm/global/5/.pnpm/commitizen@4.3.0_@types+node@20.11.24_typescript@5.3.3/node_modules/commitizen/logo
/home/antoychubekova/.local/share/pnpm/global/5/.pnpm/cra@5.4.1/node_modules/log-symbols
/home/antoychubekova/.local/share/pnpm/global/5/.pnpm/log-symbols@4.1.0
/home/antoychubekova/.local/share/pnpm/global/5/.pnpm/log-symbols@4.1.0/node_modules/log-symbols
/home/antoychubekova/.local/share/pnpm/global/5/.pnpm/node_modules/log-symbols
[1]+  Завершён      find ~ -name "log*" -print > logfile
[antoychubekova@antoychubekova ~]$
```

Рис. 4.10: Запуск процесса в фоновом режиме

Удаляю файл ~/logfile. С помощью команды ls, мы видим, что все было удалено. (рис. 4.11).

```
[antoyjchubekova@antoyjchubekova ~]$ ls
abc1  directory  file.txt  LICENSE  monthly  reports  Видео  Изображения  'Рабочий стол'
australia  Downloads  fun      logfile  my_os    ski.places  Документы  Музыка  Шаблоны
conf.txt  feathers  git-extended  may      play     work      Загрузки  Общедоступные
[antoyjchubekova@antoyjchubekova ~]$ rm logfile
[antoyjchubekova@antoyjchubekova ~]$ ls
abc1  directory  file.txt  LICENSE  my_os    ski.places  Документы  Музыка  Шаблоны
australia  Downloads  fun      may      play     work      Загрузки  Общедоступные
conf.txt  feathers  git-extended  monthly  reports  Видео  Изображения  'Рабочий стол'
[antoyjchubekova@antoyjchubekova ~]$
```

Рис. 4.11: Удаление файла

Запускаю из консоли в фоновом режиме редактор gedit, используя амперсанта в конце. Мы видим, что редактор запустился, так что можно продолжать работу как в консоли так и в редакторе. (рис. 4.12 и рис. 4.13)

```
[antoyjchubekova@antoyjchubekova ~]$ gedit &
[1] 3521
[antoyjchubekova@antoyjchubekova ~]$
```

Рис. 4.12: Запуск процесса в фоновом режиме

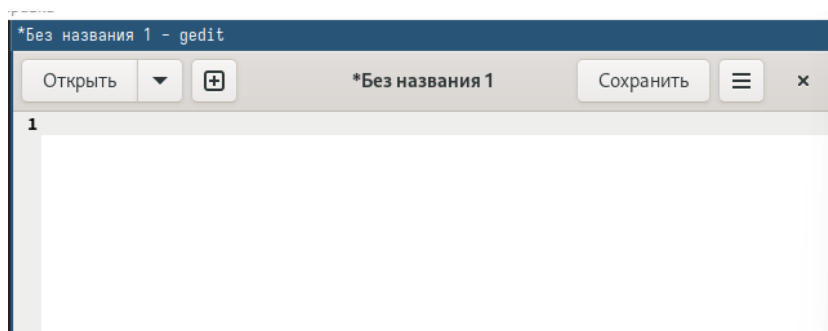


Рис. 4.13: Запуск процесса в фоновом режиме

Используя команду ps, также конвейер и фильтр grep определяю идентификатор процесса gedit. Мы видим, что идентификатор равен 3572. (рис. 4.14).

```
[antoyjchubekova@antoyjchubekova ~]$ ps aux | grep gedit
antoyjc+  3572  0.0  0.1 222456 2304 pts/0    S+   04:18   0:00 grep --color=auto gedit
[1]+  Завершён      gedit
[antoyjchubekova@antoyjchubekova ~]$
```

Рис. 4.14: Идентификатор процесса gedit

Также идентификатор процесса можно определить с помощью команды `pgrep`. (рис. 4.15).

```
[antoyjchubekova@antoyjchubekova ~]$ pgrep gedit
3606
[antoyjchubekova@antoyjchubekova ~]$
```

Рис. 4.15: Идентификатор процесса `gedit`

С помощью команды `man` просматриваю справку о команде `kill`, которая отвечает за прерывание процесса. (рис. 4.16).

```
KILL(1) User Commands KILL(1)
NAME
    kill - terminate a process

SYNOPSIS
    kill [-signal|-s signal|-p] [-q value] [-a] [--timeout milliseconds signal] [--] pid[name...]
    kill -l [number] | -L

DESCRIPTION
    The command kill sends the specified signal to the specified processes or process groups.

    If no signal is specified, the TERM signal is sent. The default action for this signal is to terminate the process. This signal should be used in preference to the KILL signal (number 9), since a process may install a handler for the TERM signal in order to perform clean-up steps before terminating in an orderly fashion. If a process does not terminate after a TERM signal has been sent, then the KILL signal may be used; be aware that the latter signal cannot be caught, and so does not give the target process the opportunity to perform any clean-up before terminating.

    Most modern shells have a builtin kill command, with a usage rather similar to that of the command described here. The --all, --pid, and --queue options, and the possibility to specify processes by command name, are local extensions.
```

Рис. 4.16: Справка команды `kill`

Используя команду `kill` завершаю процесс `gedit`, указывая его идентификатор процесса. (рис. 4.17).

```
[antoyjchubekova@antoyjchubekova ~]$ gedit &
[4] 3713
[antoyjchubekova@antoyjchubekova ~]$ kill 3713
[antoyjchubekova@antoyjchubekova ~]$
```

Рис. 4.17: Завершение процесса `gedit`

С помощью команды `man` получаю более подробную информацию про команды `df` (рис. 4.18) и `du` (рис. 4.19).


```
foot
DE(1) User Commands

NAME
    df - report file system space usage

SYNOPSIS
    df [OPTION]... [FILE]...

DESCRIPTION
    This manual page documents the GNU version of df.  df displays the amount of space available on each file system containing each file name argument.  If no file name is given, the space available on all mounted file systems is shown.  Space is shown in 1K blocks by default, unless the environment variable POSIXLY_CORRECT is set, in which case 512-byte blocks are used.

    If an argument is the absolute file name of a device node containing a mounted file system, df displays the space available on that file system rather than on the file system containing the device node.  df cannot show the space available on unmounted file systems, because on most kinds of file systems this requires very nonportable intimate knowledge of file system structures.

OPTIONS
    Show information about the file system on which each FILE resides, or all file systems by default.

    Mandatory arguments to long options are mandatory for short options too.

    -a, --all
```

Рис. 4.18: Справка команды df

```
foot
DU(1) User Commands

NAME
    du - estimate file space usage

SYNOPSIS
    du [OPTION]... [FILE]...
    du [OPTION]... --files0-from=F

DESCRIPTION
    Summarize device usage of the set of FILES, recursively for directories.

    Mandatory arguments to long options are mandatory for short options too.

    -0, --null
        end each output line with NUL, not newline

    -a, --all
        write counts for all files, not just directories

    --apparent-size
        print apparent sizes rather than device usage; although the apparent size is used
```

Рис. 4.19: Справка команды du

Выполняю команду df, которая показывает размер каждого смонтированного раздела диска. (рис. 4.20).


```

[antoyjchubekova@antoyjchubekova ~]$ df -vi
Файловая система  Инодов  ИИспользовано  ИСвободно  ИИспользовано%  Смонтировано в
/dev/sda3          0          0          0          - /
devtmpfs           243799      569      243230          1% /dev
tmpfs              249014       9      249005          1% /dev/shm
efivarfs           0           0           0          - /sys/firmware/efi/efivars
tmpfs              819200      934      818266          1% /run
tmpfs             1048576      32     1048544          1% /tmp
/dev/sda3          0           0           0          - /home
/dev/sda2          65536       37      65499          1% /boot
/dev/sda1          0           0           0          - /boot/efi
tmpfs              49802      113      49689          1% /run/user/1000
[antoyjchubekova@antoyjchubekova ~]$

```

Рис. 4.20: Выполнение команды df

Выполняю команду du, которая показывает число килобайт, используемое каждым файлом или каталогом. (рис. 4.21 и рис. 4.22)

```

[antoyjchubekova@antoyjchubekova ~]$ du -a /etc

```

Рис. 4.21: Выполнение команды du

```

8  /etc/java/java-17-openjdk/java-17-openjdk-17.0.9.0-3.fc39.x86_64/conf/net.properties
4  /etc/java/java-17-openjdk/java-17-openjdk-17.0.9.0-3.fc39.x86_64/conf/sound.properties
144 /etc/java/java-17-openjdk/java-17-openjdk-17.0.9.0-3.fc39.x86_64/conf
4  /etc/java/java-17-openjdk/java-17-openjdk-17.0.9.0-3.fc39.x86_64/lib/security/blocked.certs
4  /etc/java/java-17-openjdk/java-17-openjdk-17.0.9.0-3.fc39.x86_64/lib/security/cacerts
112 /etc/java/java-17-openjdk/java-17-openjdk-17.0.9.0-3.fc39.x86_64/lib/security/cacerts.upstream
12  /etc/java/java-17-openjdk/java-17-openjdk-17.0.9.0-3.fc39.x86_64/lib/security/default.policy
228 /etc/java/java-17-openjdk/java-17-openjdk-17.0.9.0-3.fc39.x86_64/lib/security/public_suffix_list.dat
360 /etc/java/java-17-openjdk/java-17-openjdk-17.0.9.0-3.fc39.x86_64/lib/security
360 /etc/java/java-17-openjdk/java-17-openjdk-17.0.9.0-3.fc39.x86_64/lib
504 /etc/java/java-17-openjdk/java-17-openjdk-17.0.9.0-3.fc39.x86_64
504 /etc/java/java-17-openjdk

```

Рис. 4.22: Выполнение команды du

С помощью утилиты man нахожу опцию команды find, которая выведет имена всех директорий в домашнем каталоге. (рис. 4.23).

```
FIND(1)                                General Commands Manual

NAME
    find - search for files in a directory hierarchy

SYNOPSIS
    find [-H] [-L] [-P] [-D debugopts] [-Olevel] [starting-point...] [expression]

DESCRIPTION
    This manual page documents the GNU version of find. GNU find searches the directory tree
    starting-point by evaluating the given expression from left to right, according to the
    precedence (see section OPERATORS), until the outcome is known (the left hand side is false for
    true for or), at which point find moves on to the next file name. If no starting-point is
    assumed.

    If you are using find in an environment where security is important (for example if you
    search directories that are writable by other users), you should read the 'Security Considerations'
    of the findutils documentation, which is called Finding Files and comes with findutils. This
    includes a lot more detail and discussion than this manual page, so you may find it a more
    information.

OPTIONS
    The -H, -L and -P options control the treatment of symbolic links. Command-line arguments
    are taken to be names of files or directories to be examined, up to the first argument that
```

Рис. 4.23: Справка команды find

Вывожу на экран имена всех директорий в домашнем каталоге, используя команду find и опцию -type d. (рис. 4.24) и du (рис. 4.25).

```
./fun
[antoyjchubekova@antoyjchubekova ~]$ find -type d
```

Рис. 4.24: Вывод имен директорий

```
./work/study/2023-2024/Операционные системы/os-intro/project-personal/stage1
./work/study/2023-2024/Операционные системы/os-intro/project-personal/stage1/presentation
./work/study/2023-2024/Операционные системы/os-intro/project-personal/stage1/presentation/image
./work/study/2023-2024/Операционные системы/os-intro/project-personal/stage1/report
./work/study/2023-2024/Операционные системы/os-intro/project-personal/stage1/report/bib
./work/study/2023-2024/Операционные системы/os-intro/project-personal/stage1/report/image
./work/study/2023-2024/Операционные системы/os-intro/project-personal/stage1/report/pandoc
./work/study/2023-2024/Операционные системы/os-intro/project-personal/stage1/report/pandoc/csl
./work/study/2023-2024/Операционные системы/os-intro/project-personal/stage1/report/pandoc/filters
./work/study/2023-2024/Операционные системы/os-intro/project-personal/stage1/report/pandoc/filters/pandoc
./work/study/2023-2024/Операционные системы/os-intro/project-personal/stage2
./work/study/2023-2024/Операционные системы/os-intro/project-personal/stage2/presentation
./work/study/2023-2024/Операционные системы/os-intro/project-personal/stage2/presentation/image
./work/study/2023-2024/Операционные системы/os-intro/project-personal/stage2/report
./work/study/2023-2024/Операционные системы/os-intro/project-personal/stage2/report/bib
./work/study/2023-2024/Операционные системы/os-intro/project-personal/stage2/report/image
./work/study/2023-2024/Операционные системы/os-intro/project-personal/stage2/report/pandoc
./work/study/2023-2024/Операционные системы/os-intro/project-personal/stage2/report/pandoc/csl
./work/study/2023-2024/Операционные системы/os-intro/project-personal/stage2/report/pandoc/filters
./work/study/2023-2024/Операционные системы/os-intro/project-personal/stage2/report/pandoc/filters/pandoc
```

Рис. 4.25: Вывод имен директорий

4.1 Ответы на вопросы

1. Потоки ввода-вывода (I/O streams) включают стандартный ввод (stdin), стандартный вывод (stdout) и стандартный вывод ошибок (stderr).
2. Операция “>” используется для перенаправления вывода команды в файл, при этом файл будет перезаписан, если уже существует. Операция “>>” также используется для перенаправления вывода команды в файл, но в этом случае вывод будет добавлен в конец файла, не перезаписывая его содержимое.
3. Конвейер (pipeline) - это механизм в Unix-подобных операционных системах, который позволяет передавать вывод одной команды как ввод следующей команде, образуя цепочку команд.
4. Процесс - это экземпляр программы, который выполняется на компьютере. Отличие от программы заключается в том, что программа - это набор инструкций и данных, а процесс - это исполняемая копия этой программы, которая имеет свой собственный уникальный идентификатор (PID), область памяти и состояние выполнения.
5. PID (Process ID) - уникальный идентификатор процесса. GID (Group ID) - идентификатор группы, к которой принадлежит процесс.
6. Задачи (jobs) - это команды, которые выполняются в фоновом режиме в командной оболочке Unix. Команда для управления задачами - это “jobs”.
7. Утилиты “top” и “htop” предоставляют информацию о текущих процессах на компьютере, такую как использование CPU, память и другие ресурсы. “htop” предоставляет более детальную информацию и имеет более удобный интерфейс по сравнению с “top”.
8. Команда поиска файлов - “find”. Она позволяет найти файлы и каталоги

по различным критериям, таким как имя, тип, размер и т. д. Примеры использования:

- “find /home/user -name”*.txt”” - поиск всех файлов с расширением .txt в домашнем каталоге пользователя.
- “find /var/log” - поиск всех файлов в каталоге /var/log.

9. Да, можно. Для этого можно использовать команду “grep”, чтобы выполнить поиск по содержимому файлов. Например, чтобы найти файлы, содержащие определенную строку, можно использовать следующую команду: “grep -r”строка” /путь/к/каталогу”.
10. Объем свободной памяти на жестком диске можно определить, используя команду “df”. Например: “df -h” покажет размеры файловых систем, включая информацию о доступном и используемом месте.
11. Объем вашего домашнего каталога можно определить, используя команду “du” (от “disk usage”). Например: “du -sh ~” покажет общий объем использованного места в вашем домашнем каталоге.
12. Для удаления зависшего процесса можно использовать команду “kill”. Сначала нужно определить PID зависшего процесса, а затем отправить сигнал завершения процесса. Например: “kill PID”.

5 Выводы

В ходе выполнения лабораторной работы №8 я ознакомилась с инструментами поиска файлов и фильтрации текстовых данных. Также приобрела практические навыки: по управлению процессами (и заданиями), по проверке использования диска и обслуживанию файловых систем.