

Отчёта по лабораторной работе № 7

Операционные системы

Ильина Любовь Александровна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	8
4	Выводы	21

Список иллюстраций

3.1	Вход в систему	8
3.2	Создание файла, запись и добавление в него названий файлов . .	9
3.3	Поиск в файле с фильтром, вывод и запись результатов поиска в новый файл	9
3.4	Вывод <code>ls -d c*</code> и <code>ls grep '^c*</code>	10
3.5	Вывод <code>find -name 'c*' -print</code>	10
3.6	Постраничный вывод имен файлов	10
3.7	Отбор файлов и запись их имен в файл в фоновом режиме	11
3.8	Удаление файла	11
3.9	Запуск <code>gedit</code>	11
3.10	Запуск <code>gedit</code>	12
3.11	Просмотр справки о команде <code>kill</code>	13
3.12	Завершение <code>gedit</code>	13
3.13	Просмотр справки о команде <code>df</code>	14
3.14	Просмотр справки о команде <code>du</code>	15
3.15	Выполнение команды <code>df</code>	15
3.16	Выполнение команды <code>du</code>	15
3.17	Просмотр справки о команде <code>find</code>	16
3.18	Вывод всех директорий с помощью <code>find</code>	17

Список таблиц

1 Цель работы

Ознакомиться 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. 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`. Можно ли определить этот идентификатор более простым способом?
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 Выполнение лабораторной работы

1. Осуществим вход в систему, используя соответствующее имя пользователя.
(рис. 3.1)

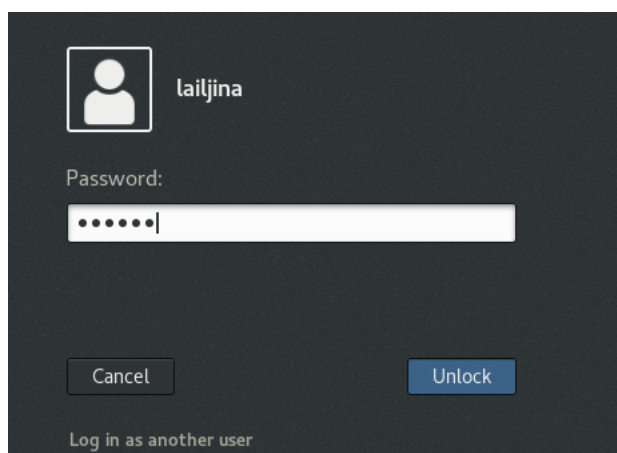


Рис. 3.1: Вход в систему

2. Командой `ls -a /etc >file.txt` запишем в файл `file.txt` названия файлов, содержащихся в каталоге `/etc`. Допишем в этот же файл названия файлов, содержащихся в вашем домашнем каталоге. `ls -a ~ >>file.txt`. Проверим результат (рис. 3.2)


```
[lailjina@lailjina report]$ ls -a /etc >file.txt
[lailjina@lailjina report]$ ls -a ~ - >>file.txt
ls: cannot access -: No such file or directory
[lailjina@lailjina report]$ ls -a ~ ->>file.txt
ls: cannot access -: No such file or directory
[lailjina@lailjina report]$ ls -a ~ >>file.txt
[lailjina@lailjina report]$ cat file.txt
.
..
abrt
adjtime
akonadi
aliases
aliases.db
alsa
alternatives
```

Рис. 3.2: Создание файла, запись и добавление в него названий файлов

3. Командой `grep '.conf' file.txt file.txt, .conf, conf.txt` `grep '.conf' file.txt > conf.txt` (рис. 3.3).

```
[lailjina@lailjina report]$ grep '.conf$' file.txt > conf.txt
[lailjina@lailjina report]$ grep '.conf$' file.txt
asound.conf
autofs.conf
autofs_ldap_auth.conf
brltty.conf
cgconfig.conf
cgrules.conf
cgsnapshot_blacklist.conf
chrony.conf
dconf
dleyna-server-service.conf
dnsmasq.conf
dracut.conf
e2fsck.conf
fprintd.conf
fuse.conf
gconf
GeoIP.conf
host.conf
idmapd.conf
ipsec.conf
```

Рис. 3.3: Поиск в файле с фильтром, вывод и запись результатов поиска в новый файл

4. Определим, какие файлы в вашем домашнем каталоге имеют имена, начинающиеся с символа `c`, несколькими командами (рис. 3.4 - 3.5) : `ls -d c* ls | grep '^c' find -name 'c'` (выводит также путь файла)

```

[lailjina@lailjina ~]$ ls -d c*
c-feathers
[lailjina@lailjina ~]$ ls | grep '^c*'
abcl
australia
c-feathers
Desktop
Documents
Downloads
env
.

```

Рис. 3.4: Вывод `ls -d c*` и `ls | grep '^c*'`

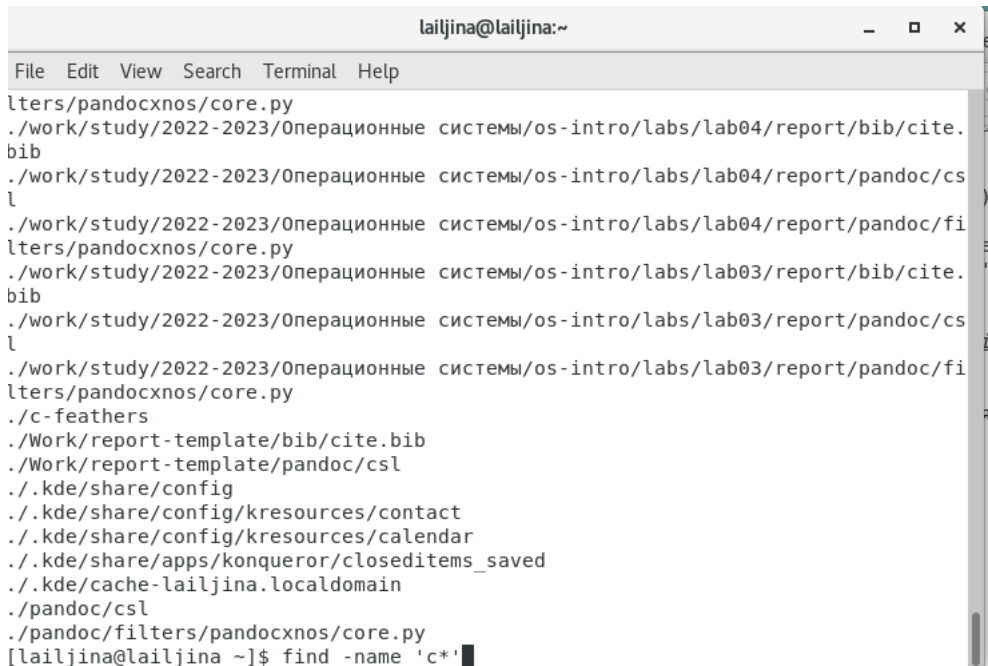


Рис. 3.5: Вывод `find -name 'c*' -print`

5. Командой `find /etc -name "h*" | less` выведем на экран (постранично) имена файлов из каталога `/etc`, начинающиеся с символа `h`. (рис. 3.6)

```

[lailjina@lailjina ~]$ find ~ -name "c*" | less

```

Рис. 3.6: Постраничный вывод имен файлов

6. Запустим в фоновом режиме, используя `find -name "log*" ~ >logfile.txt &`, процесс, который будет записывать в файл `~/logfile` файлы, имена которых начинаются с `log`. (рис. 3.7)

```

[lailjina@lailjina ~]$ find /var/log -name "log*" -print > logfile &
[1] 6777
find: '/var/log/samba': Permission denied
find: '/var/log/ppp': Permission denied
find: '/var/log/audit': Permission denied
find: '/var/log/speech-dispatcher': Permission denied
find: '/var/log/pluto/peer': Permission denied
find: '/var/log/libvirt': Permission denied
find: '/var/log/gdm': Permission denied
find: '/var/log/swtpm/libvirt/qemu': Permission denied
find: '/var/log/sss': Permission denied
find: '/var/log/mariadb': Permission denied
[1]+  Exit 1                  find /var/log -name "log*" -print > logfile
[lailjina@lailjina ~]$ cat logfile
/var/log

```

Рис. 3.7: Отбор файлов и запись их имен в файл в фоновом режиме

7. Удалите файл ~/logfile. (рис. 3.8)

```

[lailjina@lailjina ~]$ rm logfile
[lailjina@lailjina ~]$ find logfile
find: 'logfile': No such file or directory

```

Рис. 3.8: Удаление файла

8. Запустите из консоли в фоновом режиме редактор gedit. (рис. 3.9)

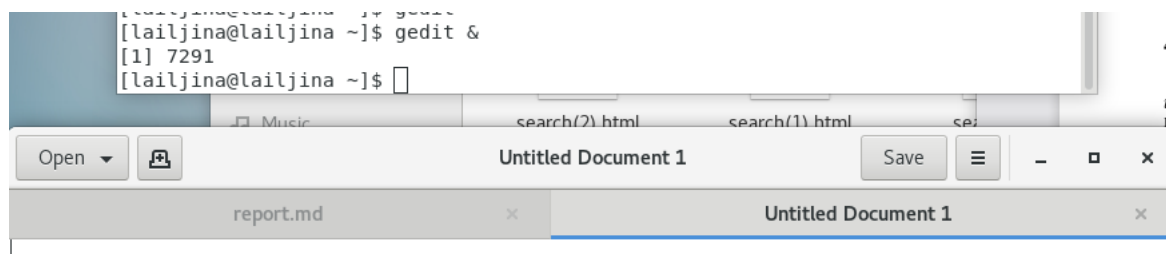


Рис. 3.9: Запуск gedit

9. Определите идентификатор процесса gedit, используя команду ps, конвейер и фильтр grep. Определим этот идентификатор более простым способом, используя: pgrep gedit или pidof gedit (рис. 3.10)

```
[lailjina@lailjina ~]$ ps aux | grep 'gedit'
lailjina  4026  0.2  0.4 828392 38872 ?        Sl   11:22   0:27 gedit /home/lai
lailjina/work/study/2022-2023/Операционные системы/os-intro/labs/lab07/report/repor
t.md
lailjina  7425  0.0  0.0 112808   980 pts/0    S+   14:03   0:00 grep --color=au
to gedit
[1]+  Done                  gedit
[lailjina@lailjina ~]$ pgrep gedit
4026
[lailjina@lailjina ~]$ pidof gedit
4026
```

Рис. 3.10: Запуск gedit

10. Прочтите справку (man) команды kill, после чего используйте её для завершения процесса gedit. (рис. 3.11 - 3.12). Команда kill для завершения процесса

```
lailjina@lailjina:~  
File Edit View Search Terminal Help  
KILL(1) User Commands KILL(1)  
  
NAME  
    kill - terminate a process  
  
SYNOPSIS  
    kill [-s signal|-p] [-q sigval] [-a] [--] pid...  
    kill -l [signal]  
  
DESCRIPTION  
    The command kill sends the specified signal to the specified process or process group. If no signal is specified, the TERM signal is sent. The TERM signal will kill processes which do not catch this signal. For other processes, it may be necessary to use the KILL (9) signal, since this signal cannot be caught.  
  
    Most modern shells have a builtin kill function, with a usage rather similar to that of the command described here. The '-a' and '-p' options, and the possibility to specify processes by command name are a local extension.  
  
    If sig is 0, then no signal is sent, but error checking is still performed.  
  
OPTIONS  
    pid... Specify the list of processes that kill should signal. Each pid can be one of five things:  
  
        n      where n is larger than 0. The process with pid n will be signaled.  
  
        0      All processes in the current process group are signaled.  
  
        -1     All processes with pid larger than 1 will be signaled.  
  
        -n    where n is larger than 1. All processes in process group n are signaled. When an argument of the form '-n' is given, and it is meant to denote a process group, either the signal must be specified first, or the argument must be preceded by a '--' option, otherwise it will be taken as the signal to send.
```

Рис. 3.11: Просмотр справки о команде kill

```
[lailjina@lailjina ~]$ man kill  
[lailjina@lailjina ~]$ kill 4026
```

Рис. 3.12: Завершение gedit

11. Выполните команды `df` и `du`, предварительно получив более подробную информацию об этих командах, с помощью команды `man`. (рис. 3.13 - 3.15)

```
lailjina@lailjina:~  
File Edit View Search Terminal Help  
DF(1) User Commands DF(1)  
  
NAME  
df - report file system disk space usage  
  
SYNOPSIS  
df [OPTION]... [FILE]...  
  
DESCRIPTION  
This manual page documents the GNU version of df. df displays the  
amount of disk space available on the file system containing each file  
name argument. If no file name is given, the space available on all  
currently mounted file systems is shown. Disk 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 disk device node contain-  
ing a mounted file system, df shows the space available on that file  
system rather than on the file system containing the device node. This  
version of df cannot show the space available on unmounted file sys-  
tems, because on most kinds of systems doing so requires very non-  
portable 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  
include pseudo. duplicate. inaccessible file svstems
```

Рис. 3.13: Просмотр справки о команде df

```

lailjina@lailjina:~
File Edit View Search Terminal Help
DU(1) User Commands DU(1)

NAME
    du - estimate file space usage

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

DESCRIPTION
    Summarize disk usage of each FILE, recursively for directories.

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

    -0, --null
        end each output line with 0 byte rather than newline

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

    --apparent-size
        print apparent sizes, rather than disk usage; although the
        apparent size is usually smaller, it may be larger due to holes
        in ('sparse') files, internal fragmentation, indirect blocks,
        and the like

    -B, --block-size=SIZE
        scale sizes by SIZE before printing them; e.g., '-BM' prints
        sizes in units of 1,048,576 bytes; see SIZE format below

```

Рис. 3.14: Просмотр справки о команде du

```

[lailjina@lailjina ~]$ df -t tmpfs
Filesystem      1K-blocks  Used Available Use% Mounted on
tmpfs            4086912    0   4086912    0% /dev/shm
tmpfs            4086912  9684   4077228    1% /run
tmpfs            4086912    0   4086912    0% /sys/fs/cgroup
tmpfs            817384    44    817340    1% /run/user/1000

```

Рис. 3.15: Выполнение команды df

```

[lailjina@lailjina ~]$ du ski.places
0      ski.places/equipment
0      ski.places/plans
0      ski.places

```

Рис. 3.16: Выполнение команды du

12. Воспользовавшись справкой команды find, выведите имена всех директо-
рий, имеющихсся в вашем домашнем каталоге find -type d -maxdepth 1 (рис.
3.17 - 3.18)

```
lailjina@lailjina:~  
File Edit View Search Terminal Help  
FIND(1) General Commands Manual FIND(1)  
  
NAME  
    find - search for files in a directory hierarchy  
  
SYNOPSIS  
    find [-H] [-L] [-P] [-D debugopts] [-Olevel] [path...] [expression]  
  
DESCRIPTION  
    This manual page documents the GNU version of find. GNU find searches  
    the directory tree rooted at each given file name by evaluating the  
    given expression from left to right, according to the rules of prece-  
    dence (see section OPERATORS), until the outcome is known (the left  
    hand side is false for and operations, true for or), at which point  
    find moves on to the next file name.  
  
    If you are using find in an environment where security is important  
    (for example if you are using it to search directories that are  
    writable by other users), you should read the "Security Considerations"  
    chapter of the findutils documentation, which is called Finding Files  
    and comes with findutils. That document also includes a lot more  
    detail and discussion than this manual page, so you may find it a more  
    useful source of information.  
  
OPTIONS  
    The -H, -L and -P options control the treatment of symbolic links.  
    Command-line arguments following these are taken to be names of files  
    or directories to be examined, up to the first argument that begins  
    with -, or the argument ' or !. That argument and any following  
    arguments are taken to be the expression describing what is to be  
    searched for. If no paths are given, the current directory is used.  
    If no expression is given, the expression -print is used (but you  
    should probably consider using -print0 instead, anyway).  
  
    This manual page talks about 'options' within the expression list.  
    These options control the behaviour of find but are specified immedi-  
    ately after the last path name. The five 'real' options -H, -L, -P, -D  
    and -O must appear before the first path name, if at all. A double  
    dash -- can also be used to signal that any remaining arguments are not  
    options (though ensuring that all start points begin with either ./  
    or / is generally safer if you use wildcards in the list of start  
    points).
```

Рис. 3.17: Просмотр справки о команде find


```
[lailjina@lailjina ~]$ find -type d -maxdepth 1
find: warning: you have specified the -maxdepth option after a non-option argume
nt -type, but options are not positional (-maxdepth affects tests specified befo
re it as well as those specified after it). Please specify options before other
arguments.

./mozilla
./cache
./dbus
./config
./local
./Desktop
./Downloads
./Templates
./Public
./Documents
./Music
./Pictures
./Videos
./ssh
./git
./gnupg
./work
./pki
./australia
./play
./usr
./Work
./kde
./gstreamer-0.10
./pandoc
./ski.plases
```

Рис. 3.18: Вывод всех директорий с помощью find

Контрольные вопросы:

1. Какие потоки ввода, вывода вы знаете? 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. Объясните разницу между операцией > и » >' Перенаправление вывода в файл '»' Перенаправление вывода в файл и открытие файла в режиме добавления (данные добавляются в конец файла)
3. Что такое конвейер? Конвейер (pipe) служит для объединения простых команд или утилит в цепочки, в которых результат работы предыдущей коман-

ды передаётся последующей. Синтаксис следующий: команда1|команда2 (это означает, что вывод команды 1 передастся на ввод команде 2)

4. Что такое процесс? Чем это понятие отличается от программы? Процесс рассматривается операционной системой как заявка на потребление всех видов ресурсов, кроме одного – процессорного времени. Этот последний важнейший ресурс распределяется операционной системой между другими единицами работы – потоками, которые и получили свое название благодаря тому, что они представляют собой последовательности (потоки выполнения) команд. Процесс – это выполнение программы. Он считается активной сущностью и реализует действия, указанные в программе. Программа представляет собой статический набор команд, а процесс это набор ресурсов и данных, использующихся при выполнении программы.
5. Что такое PID и GID? pid: идентификатор процесса (PID) процесса (processID), к которому вызывают метод gid: идентификатор группы UNIX, в котором работает программа.
6. Что такое задачи и какая команда позволяет ими управлять? 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.
7. Найдите информацию об утилитах top и htop. Каковы их функции? 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.

8. Назовите и дайте характеристику команде поиска файлов. Приведите примеры использования этой команды. `find` – это команда для поиска файлов и каталогов на основе специальных условий. Ее можно использовать в различных обстоятельствах, например, для поиска файлов по разрешениям, владельцам, группам, типу, размеру и другим подобным критериям. Команда `find` имеет такой синтаксис: `find[папка][параметры] критерий шаблон [действие]` Папка – каталог в котором будем искать Параметры – дополнительные параметры, например, глубина поиска, и т.д. Критерий – по какому критерию будем искать: имя, дата создания, права, владелец и т.д. Шаблон – непосредственно значение по которому будем отбирать файлы. Основные параметры: `-P` никогда не открывать символические ссылки `-L` – получает информацию о файлах по символическим ссылкам. Важно для дальнейшей обработки, чтобы обрабатывалась не ссылка, а сам файл. `-maxdepth` – максимальная глубина поиска по подкаталогам, для поиска только в текущем каталоге установите 1. `-depth` – искать сначала в текущем каталоге, а потом в подкаталогах `-mount` искать файлы только в этой файловой системе. `-version` – показать версию утилиты `find` `-print` – выводить полные имена файлов `-type f` – искать только файлы `-type d` – поиск папки в Linux Основные критерии: `-name` – поиск файлов по имени `-perm` – поиск файлов в Linux по режиму доступа `-user` – поиск файлов по владельцу `-group` – поиск по группе `-mtime` – поиск по времени модификации файла `-atime` – поиск файлов по дате последнего чтения `-nogroup` – поиск файлов, не принадлежащих ни одной группе `-nouser` – поиск файлов без владельцев `-newer` – найти файлы новее чем указанный `-size` – поиск файлов в Linux по их размеру Примеры: `find ~ -type d` поиск директорий в домашнем каталоге `find ~ -type f -name “.*”` поиск скрытых файлов в домашнем каталоге
9. Можно ли по контексту (содержанию) найти файл? Если да, то как? Файл по его содержимому можно найти с помощью команды `grep`: «`grep -r`» слово/выражение, которое нужно найти».

10. Как определить объем свободной памяти на жёстком диске? Утилита `df`, позволяет проанализировать свободное пространство на всех подключенных к системе разделах.
11. Как определить объем вашего домашнего каталога? При выполнении команды `du` (без указания папки и опции) можно получить все файлы и папки текущей директории с их размерами. Для домашнего каталога: `du ~/`
12. Как удалить зависший процесс? С помощью команды `kill`

4 Выводы

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