

Отчёт по лабораторной работе №6

Дисциплина: Операционные системы

Горпинич Елена Михайловна

Содержание

Цель работы.....	1
Теоретическое введение.....	1
Перенаправление ввода-вывода	1
Управление задачами	2
Управление процессами	2
Выполнение лабораторной работы.....	2
Вывод	6
Контрольные вопросы.....	6

Цель работы

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

Теоретическое введение

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`. Потоки вывода и ввода можно перенаправлять на другие файлы или устройства. Проще всего это делается с помощью символов `>`, `>>`, `<`, `<<` ## Конвейер Конвейер (`pipe`) служит для объединения простых

команд или утилит в цепочки, в которых результат работы предыдущей команды передаётся последующей. Конвейеры можно группировать в цепочки и выводить с помощью перенаправления в файл. Чаще всего скрипты на Bash используются в качестве автоматизации каких-то рутинных операций в консоли, отсюда иногда возникает необходимость в обработке stdout одной команды и передача на stdin другой команде, при этом результат выполнения команды должен обработан. ## Поиск файла Команда find используется для поиска и отображения на экран имён файлов, соответствующих заданной строке символов. ## Фильтрация текста Найти в текстовом файле указанную строку символов позволяет команда grep. ## Проверка использования диска Команда df показывает размер каждого смонтированного раздела диска.

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

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 &. 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

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. ## Получение информации о процессах

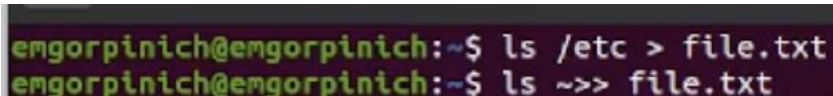
Команда ps используется для получения информации о процессах.

Для получения информации о процессах, управляемых вами и запущенных (работающих или остановленных) на вашем терминале, используйте опцию aux.

Для запуска команды в фоновом режиме необходимо в конце командной строки указать знак & (амперсанд).

Выполнение лабораторной работы

- 1) Запишем в файл file.txt названия файлов, содержащихся в каталоге /etc. Добавим в этот же файл названия файлов, содержащихся в вашем домашнем каталоге. (рис.[1])



```
emgorpinich@emgorpinich:~$ ls /etc > file.txt
emgorpinich@emgorpinich:~$ ls ~ >> file.txt
```

- 2) Выведем имена всех файлов из file.txt, имеющих расширение .conf, после чего запишем их в новый текстовый файл conf.txt. (рис.[2-3])

```
emgorpinich@emgorpinich:~$ cat file.txt | grep .conf
adduser.conf
apg.conf
appstream.conf
brltty.conf
ca-certificates.conf
ca-certificates.conf.dpkg-old
dconf
debconf.conf
deluser.conf
e2scrub.conf
sensors3.conf
sysctl.conf
ucf.conf
usb_modeswitch.conf
xattr.conf
emgorpinich@emgorpinich:~$ cat file.txt | grep .conf >> conf.txt
```

- 3) Посмотрим, какие файлы в нашем домашнем каталоге имеют имена, начинавшиеся с символа c (рис.[4])

```
emgorpinich@emgorpinich:~$ find ~ -name "c*" -print
/home/emgorpinich/.cache/mesa_shader_cache/b1/c4c79819a6ca7e658b557f351c65d4dfc3b5a9
/home/emgorpinich/.cache/mesa_shader_cache/c2
/home/emgorpinich/.cache/mesa_shader_cache/41/cf5d02f4d8c50e5610c7fcb03040de57af0862
/home/emgorpinich/.cache/mesa_shader_cache/ca
```

- 4) Выведем на экран имена файлов из каталога /etc, начинающиеся с символа h(рис.[5])

```
emgorpinich@emgorpinich:~$ find /etc -name "h*" -print
find: '/etc/multipath': Permission denied
/etc/init.d/hwclock.sh
/etc/udev/hwdb.d
/etc/cloud/templates/hosts.redhat.tpl
/etc/cloud/templates/hosts.freebsd.tpl
/etc/cloud/templates/hosts.alpine.tpl
/etc/init.d/hotplug
```

- 5) Запустим в фоновом режиме процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log. Удалим файл logfile

```
emgorpinich@emgorpinich:~$ find ~ -name "log*" -print >> logfile &
[1] 3572
emgorpinich@emgorpinich:~$ ls
conf.txt      study_2021-2022_os-intro  Загрузки      Общедоступные
file.txt      Видео                     Изображения   'Рабочий стол'
logfile       Документы                Музыка        Шаблоны
[1]+  Done                  find ~ -name "log*" -print >> logfile
emgorpinich@emgorpinich:~$ rm logfile
emgorpinich@emgorpinich:~$ ls
conf.txt      Видео                     Изображения   'Рабочий стол'
file.txt      Документы                Музыка        Шаблоны
study_2021-2022_os-intro  Загрузки      Общедоступные
```

- 6) Запустим из консоли в фоновом режиме редактор gedit.(рис.[7])

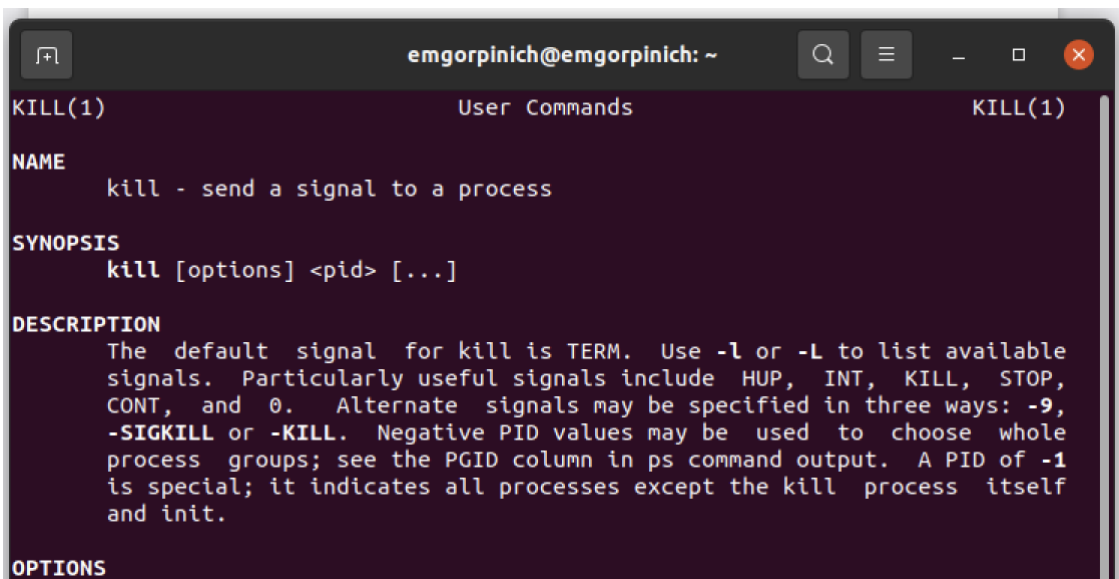
```
emgorpinich@emgorpinich:~$ gedit &
[1] 3618
emgorpinich@emgorpinich:~$ jobs
[1]+  Running                  gedit &
```

- 7) Определим идентификатор процесса gedit,используя две различные команды (рис.[7])

```
emgorpinich@emgorpinich:~$ ps aux| grep -i gedit
emgorpi+  3618  0.1  1.4 816344 57040 pts/0    Sl   11:22   0:00 gedit
emgorpi+  3698  0.0  0.0  5964   672 pts/0    S+   11:28   0:00 grep --color=
auto -i gedit
emgorpinich@emgorpinich:~$ pidof gedit
3618
```

- 8) Прочитаем справку (man) команды kill, после чего используем её для завершения процесса gedit.(рис.[8-9])

```
emgorpinich@emgorpinich:~$ man kill
emgorpinich@emgorpinich:~$ kill 3618
emgorpinich@emgorpinich:~$ jobs
[1]+  Terminated              gedit
```



- 9) Выполним команды df и du, предварительно получив более подробную информацию об этих командах, с помощью команды man.(рис.[9-12])

```
emgorpinich@emgorpinich:~$ man df
emgorpinich@emgorpinich:~$ man du
emgorpinich@emgorpinich:~$ df
Filesystem            1K-blocks    Used Available Use% Mounted on
udev                  1941364         0   1941364  0% /dev
tmpfs                  401064      1700    399364  1% /run
/dev/mapper/ubuntu--vg-ubuntu--lv 10255636 7905292   1809672 82% /
tmpfs                  2005320         0   2005320  0% /dev/shm
tmpfs                   5120         4     5116  1% /run/lock
```

(Скриншот процесса выполнения du на Linux не сохранился, поэтому я вставила фото выполнения этой команды на своём личном компьютере)


```
elenagorpinich@MacBook-Air-Elena ~ % du
1400    ./Music/Audio Music Apps/Databases/Tags
1400    ./Music/Audio Music Apps/Databases
0       ./Music/Audio Music Apps/Patches/Instrument
0       ./Music/Audio Music Apps/Patches
0       ./Music/Audio Music Apps/Custom Icons
0       ./Music/Audio Music Apps/Sampler Instruments
0       ./Music/Audio Music Apps/Plug-In Settings/Stereo Delay
0       ./Music/Audio Music Apps/Plug-In Settings/Klopfgeist
0       ./Music/Audio Music Apps/Plug-In Settings/Channel EQ
0       ./Music/Audio Music Apps/Plug-In Settings/Vintage Electric Piano
0       ./Music/Audio Music Apps/Plug-In Settings/Echo
0       ./Music/Audio Music Apps/Plug-In Settings/PlatinumVerb

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

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 the set of FILEs, recursively for directories.

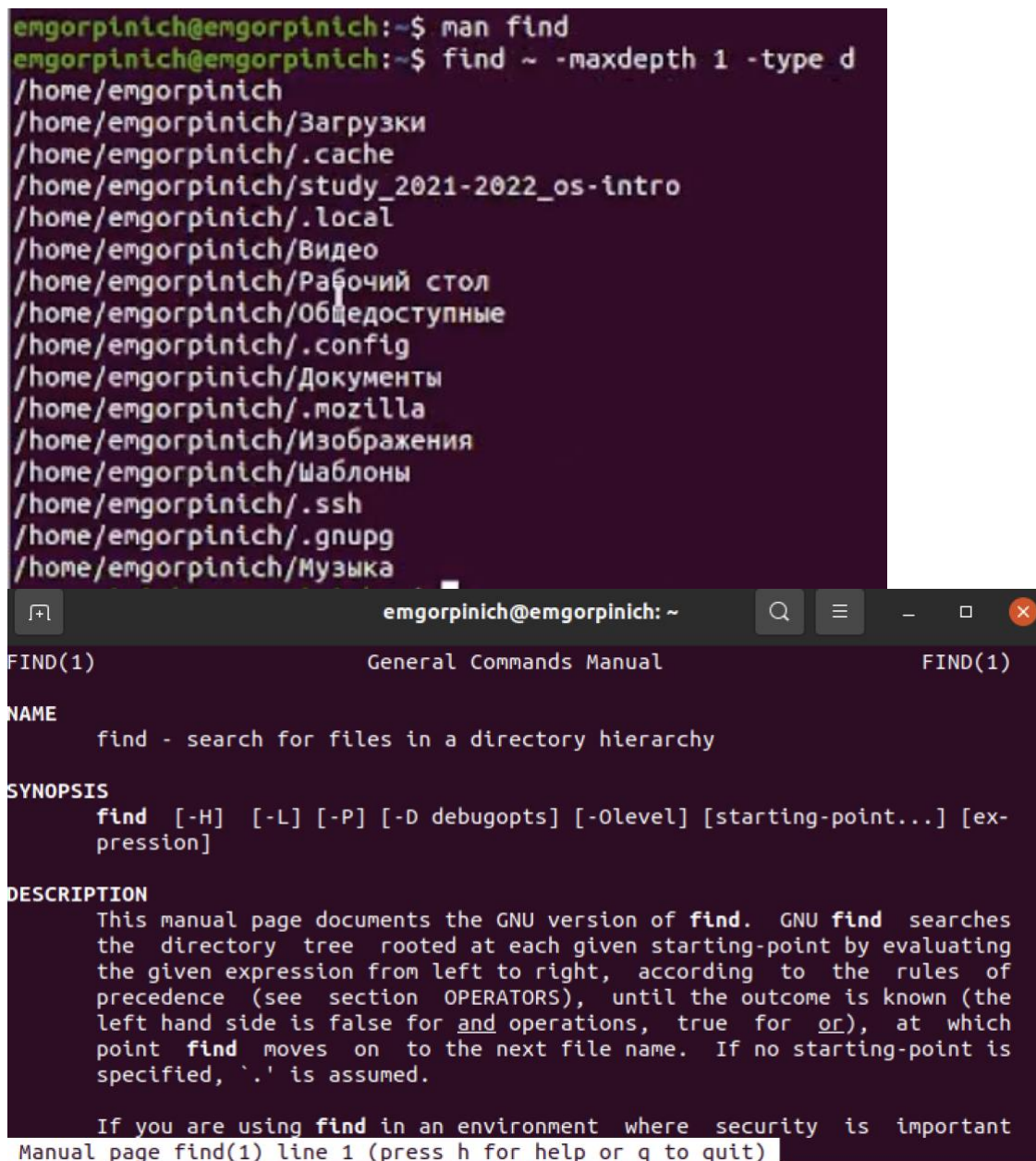
    Mandatory arguments to long options are mandatory for short options
    too.

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

    -a, --all
    Manual page du(1) line 1 (press h for help or q to quit)
```

- 10) Воспользовавшись справкой команды `find`, просмотрим имена всех директорий, имеющих в нашем домашнем каталоге. (рис.[13-13.1])

```
emgorpinich@emgorpinich:~$ man find
emgorpinich@emgorpinich:~$ find ~ -maxdepth 1 -type d
/home/emgorpinich
/home/emgorpinich/Загрузки
/home/emgorpinich/.cache
/home/emgorpinich/study_2021-2022_os-intro
/home/emgorpinich/.local
/home/emgorpinich/Видео
/home/emgorpinich/Рабочий стол
/home/emgorpinich/Общедоступные
/home/emgorpinich/.config
/home/emgorpinich/Документы
/home/emgorpinich/.mozilla
/home/emgorpinich/Изображения
/home/emgorpinich/Шаблоны
/home/emgorpinich/.ssh
/home/emgorpinich/.gnupg
/home/emgorpinich/Музыка
```



The screenshot shows a terminal window with a dark background. The top part displays the execution of the 'find' command to list directories in the home directory. The bottom part shows the manual page for 'find', which includes sections for NAME, SYNOPSIS, and DESCRIPTION. The terminal window has a title bar with the text 'emgorpinich@emgorpinich: ~' and standard window controls.

```
emgorpinich@emgorpinich: ~
FIND(1)                                General Commands Manual                                FIND(1)

NAME
    find - search for files in a directory hierarchy

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

DESCRIPTION
    This manual page documents the GNU version of find. GNU find searches
    the directory tree rooted at each given starting-point by evaluating
    the given expression from left to right, according to the rules of
    precedence (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 no starting-point is
    specified, find assumes the current directory.

    If you are using find in an environment where security is important
    Manual page find(1) line 1 (press h for help or q to quit)
```

Вывод

В ходе лабораторной работы я 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.

Контрольные вопросы

- 1) Потоки ввода/вывода используются для передачи данных в файловые потоки, на консоль или на сетевые соединения. Существует два вида потоков ввода/вывода: байтовые и символьные.

- 2) “>” : Перезаписывает существующий файл или создает файл, если файл с указанным именем отсутствует в каталоге. “>>” : добавляет существующий файл или создает файл, если файл с указанным именем отсутствует в каталоге.
- 3) Конвейер — некоторое множество процессов, для которых выполнено следующее перенаправление ввода-вывода: то, что выводит на поток стандартного вывода предыдущий процесс, попадает в поток стандартного ввода следующего процесса.
- 4) Процесс относится к выполнению программы - он представляет собой запущенный экземпляр программы, составленный из инструкций, данных, считанных из файлов, других программ, или полученных от пользователя. Основное различие между программой и процессом заключается в том, что программа представляет собой группу инструкций для выполнения определенной задачи, тогда как процесс представляет собой программу в процессе выполнения. Хотя процесс является активной сущностью, программа считается пассивной.
- 5) *PID* - идентификационный номер уникального процесса *GID* - идентификатор группы
- 6) Задача - это базовая единица программирования, которой управляет операционная система. Существует множество команд для управления процессами в командной строки. Например: `ps` — список запущенных процессов на компьютере; `kill` — отправка сигнала одному или нескольким процессам (обычно «убийство» процесса); `jobs` — просмотр списка собственных задач (процессов); `bg` — выполнение процесса в фоновом режиме; `fg` — возврат процесса из фонового режима
- 7) **top** — команда, которая выводит список работающих в системе процессов и информацию о них. По умолчанию она в реальном времени сортирует их по нагрузке на процессор. Команда **htop** — это команда, которая позволяет пользователю интерактивно отслеживать необходимые процессы системы в режиме реального времени.
- 8) Команда `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. Найти файлы с определенной маской прав, например, 0664: `find . -type f -perm 0664`
- 9) Да, с помощью команды `grep`. Синтаксис команды выглядит следующим образом: `grep [опции] шаблон [имя файла...]` Или: `команда | grep [опции] шаблон`
- 10) Используя команду `df`
- 11) С помощью команды `du`
- 12) Командой `kill`. Синтаксис команды: `$ kill -сигнал pid_процесса`