

Лабораторная работа №6

Отчет по лабораторной работе

Хусяинова Адиля Фаритовна

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.

Задание

- 1) Сделать отчёт по лабораторной работе №6 в формате Markdown.
- 2) В качестве отчёта просьба предоставить отчёты в 3 форматах: pdf, docx и md (в архиве, поскольку он должен содержать скриншоты, Makefile и т.д.).

Выполнение лабораторной работы

1. Осуществляем вход в систему, используя соответствующее имя пользователя.
2. Запишем в файл file.txt названия файлов, содержащихся в каталоге /etc и допишем названия файлов, содержащихся в каталоге /etc, используем команду «ls-a/etc> file.txt». Далее с помощью команды «ls-a~ » file.txt» дописываем в этот же файл названия файлов, содержащихся в домашнем каталоге. Командой «catfile.txt» просматриваем файл(рис.1)

```
afkhusyainova@dk5n53 ~ $ ls -a /etc >file.txt
afkhusyainova@dk5n53 ~ $ ls -a ~ >> file.txt
afkhusyainova@dk5n53 ~ $ cat file.txt
.
..
a2ps
acpi
adjtime
afs.keytab
alsa
apache2
apparmor.d
appstream.conf
ati
audit
autofs
avahi
bash
bash_completion.d
bindresvport.blacklist
binfmt.d
blkid.tab.old
bluetooth
brltty
brltty.conf
ca-certificates
ca-certificates.conf
cachefilesd.conf
cfg-update.conf
cfg-update.hosts
cgroup
chromium
chrony
cifs-utils
common-lisp
conf.d
cpufreq-bench.conf
cron.d
cron.daily
cron.hourly
cron.monthly
crontab
cron.weekly
csh.cshrc
csh.env
csh.login
cups
cupshelpers
dbus-1
dconf
default
dhcp
```

Рис. 0.1.: Запись файлов

3. Выводим имена всех файлов из file.txt, имеющих расширение .conf, записываем их в новый текстовый файл conf.txt, используя команду «grep-e'.conf\$'file.txt> conf.txt». С помощью команды «catconf.txt», можем проверить правильность наших действий(рис.2)

```
afkhusyainova@dk5n53 ~ $ grep -e '\.conf$' file.txt > conf.txt
afkhusyainova@dk5n53 ~ $ cat conf.txt
appstream.conf
brltty.conf
ca-certificates.conf
cachefilesd.conf
cfg-update.conf
cpufreq-bench.conf
dhcpcd.conf
dispatch-conf.conf
dleyna-server-service.conf
dnsmasq.conf
dracut.conf
e2fsck.conf
e2scrub.conf
etc-update.conf
fluidsynth.conf
fuse.conf
gai.conf
genkernel.conf
gssapi_mech.conf
host.conf
idmapd.conf
idn2.conf
idnalias.conf
krb5.conf
ldap.conf
ld.so.conf
libaudit.conf
lightdm.conf
locale.conf
logrotate.conf
mailutils.conf
make.conf
man.conf
man_db.conf
mdadm.conf
metalog.conf
mke2fs.conf
mlocate-cron.conf
modules.conf
mplayer.conf
nscd.conf
nslcd.conf
nss-ldapd.conf
nsswitch.conf
nsswitch-sss.conf
ntp.conf
ntpd.conf
pear.conf
pe-format2.conf
```

Рис. 0.2.: Выводим имена файлов

4. Определить, какие файлы в домашнем каталоге имеют имена, начинающиеся с символа “с”, можно несколькими командами: - «find~ -maxdepth1 -name“с” -print», - «ls~/с»; - «ls-a~ | grepс*»

```
afkhusyainova@dk5n53 ~ $ find ~ -maxdepth 1 -name "c*" -print
/afs/.dk.sci.pfu.edu.ru/home/a/f/afkhusyainova/conf.txt
afkhusyainova@dk5n53 ~ $ ls ~/c*
/afs/.dk.sci.pfu.edu.ru/home/a/f/afkhusyainova/conf.txt
afkhusyainova@dk5n53 ~ $ ls -a ~ | grep c*
conf.txt
```

Рис. 0.3.: Создание файлов

5. Чтобы вывести на экран имена файлов из каталога /etc, начинающиеся с символа “h”, используем команду `find /etc -maxdepth 1 -name “h*” | less`

```
afkhusyainova@dk5n53 ~ $ find /etc -maxdepth 1 -name "h*" | less
```

Рис. 0.4.: Команда `find /etc -maxdepth 1 -name “h*” | less`


```
/etc/hsqldb  
/etc/host.conf  
/etc/harbour  
/etc/hosts  
/etc/httpd  
/etc/highlight  
/etc/hostname  
/etc/hotplug.d  
/etc/harbour.cfg  
/etc/htdig  
/etc/hosts.allow  
/etc/hotplug  
/etc/hal  
lines 1-13/13 (END)
```

Рис. 0.5.: Список файлов

6. Запускаем в фоновом режиме процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log, используя команду «find/-name“log*” > logfile&». Командой «catlogfile» проверяю выполненные действия (Рисунок 10). Далее удалила файл ~/logfile с помощью команды «rm logfile», данный файл уже был удалён, поэтому программа выводит сообщение об этом

```
afkhusyainova@dk5n53 ~ $ cat logfile
```

Рис. 0.6.: Команда cat logfile, удаление файла

```
afkhusyainova@dk5n53 ~ $ rm logfile  
rm: невозможно удалить 'logfile': Нет такого файла или каталога
```

Рис. 0.7.: Удаляем файл с помощью команды rm

8. Запускаем редактор gedit в фоновом режиме командой «gedit &» После этого на экране появляется окно редактора.

```
afkhusyainova@dk5n53 ~ $ gedit &  
[2] 16110  
afkhusyainova@dk5n53 ~ $ ps | grep -i "gedit"  
[2]+  Завершён      gedit
```

Рис. 0.8.: Редактор gedit

9. Чтобы определить идентификатор процесса gedit, используем команду «ps| grep -i“gedit”», чтобы узнать идентификатор процесса можно использовать команду «pgrep gedit»или «pidof gedit».

```
afkhusyainova@dk5n53 ~ $ gedit &  
[2] 16110  
afkhusyainova@dk5n53 ~ $ ps | grep -i "gedit"  
[2]+  Завершён      gedit
```

Рис. 0.9.: Команда «ps| grep -i“gedit”»

10. Прочитав информацию о команде kill с помощью команды «man kill», используем её для завершения процесса gedit

```
afkhusyainova@dk5n53 ~ $ man kill
```

Рис. 0.10.: Команда kill

```

NAME
    kill - send a signal to a process

SYNOPSIS
    kill [options] sigid [...]

DESCRIPTION
    The default signal for kill is TERM. Use 1 or % to list available signals. Particularly useful signals include HUP, INT, KILL, STOP, CONT, and 9. Alternate signals may be specified in three ways: -SIGNAME or -KILL. Negative PID values may be used to choose whose process group; see the PIDs column in an command output. A PID of 0 is special; it indicates all processes except the kill process itself and init.

OPTIONS
    -sigid [...]
        Send signal to every sigid listed.

    -signal
    -SIGNAME
    -signal signals
        Specify the signal to be sent. The signal can be specified by using name or number. The behavior of signals is outlined in signal(7) manual page.

    -s signame value
        Use signame(s) rather than kill(1) and the value argument is used to specify an integer to be sent with the signal. If the receiving process has installed a handler for this signal using the SA_SIGINFO flag to sigaction(2), then it can obtain this data via the si_value field of the siginfo_t structure.

    -l, --list [signal]
        List signal names. This option has optional argument, which will convert signal number to signal name, or other way round.

    -l, --table
        List signal names in a nice table.

NOTES
    Your shell (command line interpreter) may have a built-in kill command. You may need to run the command described here as /bin/kill to solve the conflict.

EXAMPLES
    kill -SIG ...
        Kill all processes you can kill.

    kill -1 ...
        Translate number 1 into a signal name.

    kill -1
        List the available signal choices in a nice table.

    kill 123 543 678 902
        Send the default signal, SIGTERM, to all those processes.

SEE ALSO
    kill(2), killall(1), nice(1), nohup(1), renice(1), signal(7), sigaction(2), wait(1)

```

Рис. 0.11.: Информация о команде kill

11. С помощью команд «man df» и «man du» узнаем информацию по необходимым командам и далее использую их.

```

afkhusyainova@dk5n53 ~ $ man df
afkhusyainova@dk5n53 ~ $ man du

```

Рис. 0.12.: Команды «man df» и «man du»

```

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 usage 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 mount node containing 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 systems, because on most kinds of systems doing so 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
        Include pseudo, duplicate, inaccessible file system

    -B, --block-size SIZE
        write sizes by SIZE before printing them, e.g., '-B4' prints sizes in units of 1,048,576 bytes; see SIZE format below

    -h, --human-readable
        print sizes in powers of 1024 (e.g., 1.02M)

    -H, --si
        print sizes in powers of 1000 (e.g., 1.1G)

    -i, --inodes
        list inode information instead of block usage

    -k, --block-size SIZE
        like --block-size=1

    -l, --local
        limit listing to local file systems

    --no-sync
        do not invoke sync before getting usage info (default)

    --output=FIELD...
        use the output format defined by FIELD.LIST, or print all fields if FIELD.LIST is omitted.

    -P, --portability
        use the POSIX output format

    --sync
        invoke sync before getting usage info

```

Рис. 0.13.: Информация о команде df

```
001)                                user commands                                001)

NAME    du - estimate file space usage

SYNOPSIS
  du [OPTION]... [FILE]...
  du [OPTION]... --[help|usage]

DESCRIPTION
  Summarize disk usage of the set of FILEs, recursively for directories.

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

  -D, --mail
      end each output line with MAIL, not 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=B
      scale sizes by B before printing them; e.g., "-BM" prints sizes in units of 1,048,576 bytes; see BULK format below

  -b, --bytes
      equivalent to "--apparent-size --block-size=1"

  -c, --total
      produce a grand total

  -D, --dereference-args
      dereference only symlinks that are listed on the command line

  -d, --max-depth=N
      print the total for a directory (or file, with --all) only if it is N or fewer levels below the command line argument; --max-depth=0 is the same as --summary

  --files-from=F
      summarize disk usage of the NUL-terminated file names specified in file F; if F is -, then read names from standard input

  -H, --human-readable
      equivalent to "--dereference-args -D"

  -h, --human-readable
      print sizes in human readable format (e.g., 1K 234M 2G)

  --inodes
      list inode usage information instead of block usage
```

Рис. 0.14.: Информация о команде du

```
afkhusyainova@dk5n53 ~ $ df
df: /run/user/4278/doc: Операция не позволена
Файловая система      1K-блоков  Использовано    Доступно  Использовано%  Смонтировано в
none                  3999736         20268        3979468           1% /run
udev                  10240           0         10240           0% /dev
tmpfs                  3999736         18288        3981448           1% /dev/shm
/dev/sda8              490892692      78546716      387340232          17% /
tmpfs                  3999736         191148        3808588           5% /tmp
/dev/sda6              91164400       675908        85814548           1% /var/cache/openafs
AFS                   2147483647      0      2147483647           0% /afs
mark.sci.pfu.edu.ru:/usr/portage 18350080      5853440      11013376          35% /usr/portage
tmpfs                  799944          204        799740           1% /run/user/4278
```

Рис. 0.15.: Команда df в консоли

```

afkhusyainova@dk5n53 ~ $ du
2      ./public/public_html
4      ./public
434    ./cache/samba
436    ./cache
19     ./config/dconf
44     ./config/gtk-3.0/assets
58     ./config/gtk-3.0
2      ./config/autostart-scripts
7      ./config/kdeconnect
2      ./config/akonadi
3      ./config/xsettingsd
3      ./config/goa-1.0
73     ./config/session
74     ./config/libaccounts-glib
3      ./config/ibus/bus
5      ./config/ibus
2      ./config/systemd
3      ./config/gtk-4.0
5      ./config/evolution/sources
42     ./config/evolution/mail
58     ./config/evolution
2      ./config/libreoffice/4/user/config/soffice.cfg/modules/swriter/menubar
2      ./config/libreoffice/4/user/config/soffice.cfg/modules/swriter/popupmenu
2      ./config/libreoffice/4/user/config/soffice.cfg/modules/swriter/toolbar
2      ./config/libreoffice/4/user/config/soffice.cfg/modules/swriter/statusbar
2      ./config/libreoffice/4/user/config/soffice.cfg/modules/swriter/images/Bitmaps
4      ./config/libreoffice/4/user/config/soffice.cfg/modules/swriter/images
14     ./config/libreoffice/4/user/config/soffice.cfg/modules/swriter
2      ./config/libreoffice/4/user/config/soffice.cfg/modules/scalc/menubar
2      ./config/libreoffice/4/user/config/soffice.cfg/modules/scalc/popupmenu
2      ./config/libreoffice/4/user/config/soffice.cfg/modules/scalc/toolbar
2      ./config/libreoffice/4/user/config/soffice.cfg/modules/scalc/statusbar
2      ./config/libreoffice/4/user/config/soffice.cfg/modules/scalc/images/Bitmaps
4      ./config/libreoffice/4/user/config/soffice.cfg/modules/scalc/images
14     ./config/libreoffice/4/user/config/soffice.cfg/modules/scalc
2      ./config/libreoffice/4/user/config/soffice.cfg/modules/simpres/menubar
2      ./config/libreoffice/4/user/config/soffice.cfg/modules/simpres/popupmenu
2      ./config/libreoffice/4/user/config/soffice.cfg/modules/simpres/toolbar
2      ./config/libreoffice/4/user/config/soffice.cfg/modules/simpres/statusbar

```

Рис. 0.16.: Команда du в консоли

df – утилита, показывающая список всех файловых систем по именам устройств, сообщает их размер, занятое и свободное пространство и точки монтирования. Синтаксис: df[опции] устройств. du – утилита, предназначенная для вывода информации об объеме дискового пространства, занятого файлами и директориями. Она принимает путь к элементу файловой системы и выводит информацию о количестве байт дискового пространства или блоков диска, задействованных для его хранения. Синтаксис: du [опции] каталог_или_файл

12. Выводим имена всех директорий, имеющих в моем домашнем каталоге с помощью команды «find~ -typed», до этого получаем информацию с помощью команды «man find»

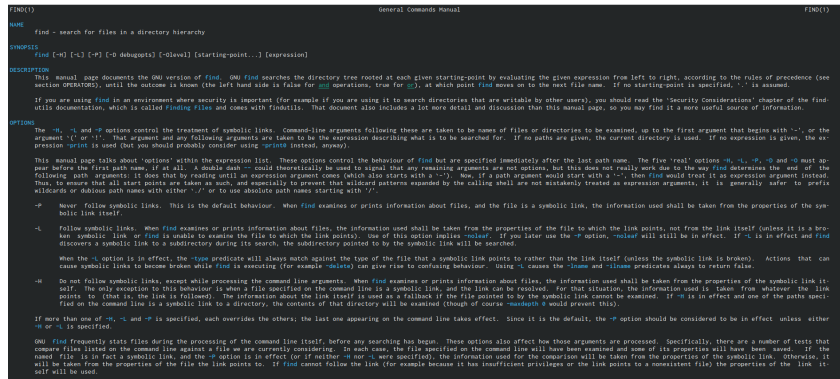


Рис. 0.17.: Информация команды find

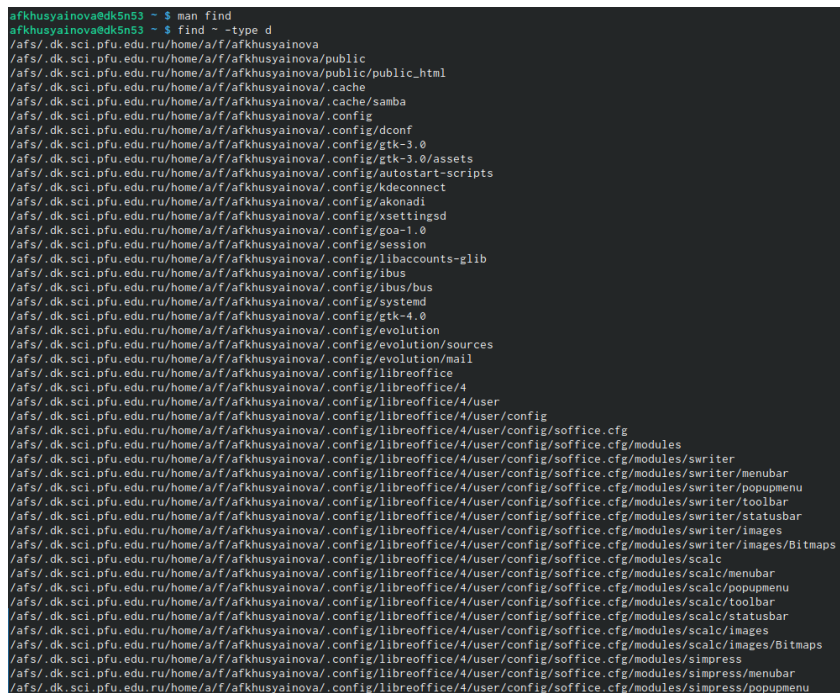


Рис. 0.18.: Имена всех директорий

#Выводы

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). 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`. 2). `>` Перенаправление вывода в файл `>>` Перенаправление вывода в файл и открытие файла в режиме добавления (данные добавляются в конец файла)/ 3). Конвейер (`pipe`) служит для объединения простых команд или утилит в цепочки, в которых результат работы предыдущей команды передаётся последующей. Синтаксис следующий: `команда1|команда2` (это означает, что вывод команды 1 передаётся на ввод команде 2) 4). 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). `pid`: идентификатор процесса (PID) процесса (`processID`), к которому вызывают метод `gid`: идентификатор группы

UNIX, в котором работает программа. 6). Любую выполняющуюся в консоли команду или внешнюю программу можно запустить в фоновом режиме. Для этого следует в конце имени команды указать знак амперсанда &. Запущенные фоновые программы называются задачами (jobs). Ими можно управлять с помощью команды jobs, которая выводит список запущенных в данный момент задач. 7). top – это консольная программа, которая показывает список работающих процессов в системе. Программа в реальном времени отсортирует запущенные процессы по их нагрузке на процессор. htop – это продвинутый консольный мониторинг процессов. Утилита выводит постоянно меняющийся список системных процессов, который сортируется в зависимости от нагрузки на ЦПУ. Если делать сравнение с top, то htop показывает абсолютно все процессы в системе, время их непрерывного использования, загрузку процессоров и расход оперативной памяти. 8). 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 -typef - искать только файлы -typed - 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). Файл по его содержимому можно найти с помощью команды grep: «grep -r” слово/выражение, которое нужно найти”». 10). Утилита df, позволяет проанализировать свободное пространство на всех подключенных к системе разделах. 11). 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). Основные сигналы (каждый сигнал имеет свой номер), которые используются для завершения процесса: SIGINT–самый безобидный сигнал завершения, означает Interrupt. Он отправляется процессу, запущенному из терминала с помощью сочетания клавиш Ctrl+C. Процесс правильно завершает все свои действия и возвращает управление; SIGQUIT–это еще один сигнал, который отправляется с помощью сочетания клавиш, программе, запущенной в терминале. Он сообщает ей что нужно завершиться и программа может выполнить корректное завершение или проигнорировать сигнал. В отличие от предыдущего, она генерирует дампы памяти. Сочетание клавиш Ctrl+/.; SIGHUP–сообщает процессу, что соединение с управляющим терминалом разорвано, отправляется, в основном, системой при разрыве соединения с интернетом; SIGTERM–немедленно завершает процесс, но обрабатывается программой, поэтому позволяет ей завершить дочерние процессы и освободить все ресурсы; SIGKILL–тоже немедленно завершает процесс, но, в отличие от предыдущего варианта, он не передается самому процессу, а обрабатывается ядром. Поэтому ресурсы и дочерние процессы остаются запущенными. Также для передачи сигналов процессам в Linux используется утилита kill, её синтаксис: kill [-сигнал] [pid_процесса] (PID – уни-

кальный идентификатор процесса). Сигнал представляет собой один из выше перечисленных сигналов для завершения процесса. Перед тем, как выполнить остановку процесса, нужно определить его PID. Для этого используют команды ps и grep. Команда ps предназначена для вывода списка активных процессов в системе и информации о них. Команда grep запускается одновременно с ps (в канале) и будет выполнять поиск по результатам команды ps. Утилита pkill – это оболочка для kill, она ведет себя точно так же, и имеет тот же синтаксис, только в качестве идентификатора процесса ей нужно передать его имя. killall работает аналогично двум предыдущим утилитам. Она тоже принимает имя процесса в качестве параметра и ищет его PID в директории /proc. Но эта утилита обнаружит все процессы с таким именем и завершит их.