

Front matter

lang: ru-RU title: Презентация по лабораторной работе №8 subtitle: Операционные системы author:

- Цоппа Е.Э institute:
- Российский университет дружбы народов, Москва, Россия

i18n babel

babel-lang: russian babel-otherlangs: english

Formatting pdf

toc: false toc-title: Содержание slide_level: 2 aspectratio: 169 section-titles: true theme: metropolis header-includes:

- \metroset{progressbar=frametitle,sectionpage=progressbar,numbering=fraction}
- \makeatletter'
- \beamer@ignorenonframefalse'
- \makeatother'

Fonts

mainfont: PT Serif romanfont: PT Serif sansfont: PT Sans monofont: PT Mono mainfontoptions: Ligatures=TeX romanfontoptions: Ligatures=TeX sansfontoptions: Ligatures=TeX,Scale=MatchLowercase monofontoptions: Scale=MatchLowercase,Scale=0.9

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. Осуществите вход в систему, используя соответствующее имя пользователя.
2. Запишите в файл file.txt названия файлов, содержащихся в каталоге /etc. Допи- шите 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 текстовой файл conf.txt.
4. Определите, какие файлы в вашем домашнем каталоге имеют имена, начинавшиеся с символа с? 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 директорий, имею- щихся в вашем домашнем каталоге.

Выполнение лабораторной работы

Вхожу в систему,используя соответствующее имя пользователя. С помощью перенапраления ">" добавляю в файл file.txt название файлов из /etc. С помощью команды head проверяю первые 10 строк файла

```
[evatsoppa@fedora ~]$ ls -LR /etc > file.txt
ls: невозможно получить доступ к '/etc/grub2-efi.cfg': Отказано в доступе
ls: невозможно получить доступ к '/etc/grub2.cfg': Отказано в доступе
ls: невозможно открыть каталог '/etc/audit': Отказано в доступе
ls: невозможно открыть каталог '/etc/credstore': Отказано в доступе
ls: невозможно открыть каталог '/etc/credstore.encrypted': Отказано в доступе
ls: невозможно открыть каталог '/etc/cups/ssl': Отказано в доступе
ls: невозможно открыть каталог '/etc/dhcp': Отказано в доступе
ls: невозможно открыть каталог '/etc/firewalld': Отказано в доступе
ls: невозможно открыть каталог '/etc/grub.d': Отказано в доступе
ls: невозможно открыть каталог '/etc/httpd/logs': Отказано в доступе
ls: невозможно открыть каталог '/etc/httpd/run': Отказано в доступе
ls: невозможно открыть каталог '/etc/httpd/statel': Отказано в доступе
```

Добавляю в файл имена из домашнего каталога

```
[evatsoppa@fedora ~]$ ls -LR ~/ >> file.txt
```

Проверяю файл file.txt

Вывожу на экран имена всех файлов,содержащих расщирение .conf, с помощью команды grep

```
[evatsoppa@fedora ~]$ grep .conf file.txt
anthy-unicode.conf
appstream.conf
asound.conf
brltty.conf
chrony.conf
dconf
dleyna-renderer-service.conf
dleyna-server-service.conf
dnsmasq.conf
dracut.conf
```

Записываю найденные файлы в новый файл

```
[evatsoppa@fedora ~]$ grep .conf file.txt > conf.txt
[evatsoppa@fedora ~]$ head conf.txt
anthy-unicode.conf
appstream.conf
asound.conf
brltty.conf
chrony.conf
dconf
dleyna-renderer-service.conf
dleyna-server-service.conf
dnsmasq.conf
dracut.conf
```

Нахожу в домашнем каталоге все файлы,которые начинаются на "с"

```
[evatsoppa@fedora ~]$ find ~ -name "c*" -print
/home/evatsoppa/.mozilla/firefox/f8liclpl.default-release/compatibility.ini
/home/evatsoppa/.mozilla/firefox/f8liclpl.default-release/cookies.sqlite
/home/evatsoppa/.mozilla/firefox/f8liclpl.default-release/cert9.db
/home/evatsoppa/.mozilla/firefox/f8liclpl.default-release/storage/permanent/chrome
/home/evatsoppa/.mozilla/firefox/f8liclpl.default-release/storage/default/https+
++web.telegram.org/cache
/home/evatsoppa/.mozilla/firefox/f8liclpl.default-release/storage/default/https+
++web.telegram.org/cache/caches.sqlite
/home/evatsoppa/.mozilla/firefox/f8liclpl.default-release/storage/default/https+
++www.youtube.com^partitionKey=%28https%2Cgoogle.com%29/cache
```

Еще один способ найти элементы с первым символом. Однако этот способ не работает для поиска файлов из подкаталогов

```
[evatsoppa@fedora ~]$ ls -LR | grep c*
conf.txt
```

С помощью опции find, вывожу на экран имена файлов из каталог /etc, начинающихся с символа h

```
[evatsoppa@fedora ~]$ find /etc -name "h*" -print
find: '/etc/audit': Отказано в доступе
/etc/avahi/hosts
/etc/brltty/Contraction/ha.ctb
/etc/brltty/Input/bm/horizontal.kti
/etc/brltty/Input/hd
/etc/brltty/Input/hm
/etc/brltty/Input/ht
/etc/brltty/Input/hw
/etc/brltty/Text/he.ttb
/etc/brltty/Text/hi.ttb
/etc/brltty/Text/hr.ttb
```

Запускаю в фоновом режиме процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log.

```
[evatsoppa@fedora ~]$ find ~ -name "log*" -print > logfile &
[1] 3217
```

Проверяю, что файл создан и удаляю его.

```
[evatsoppa@fedora ~]$ ls
abc1      file.txt  monthly  ski.plases  Документы  Музыка  Шаблоны
bin       logfile  my_os    work        Загрузки  Общедоступные
conf.txt  may      reports  Видео       Изображения  'Рабочий стол'
[1]+  Завершён      find ~ -name "log*" -print > logfile
[evatsoppa@fedora ~]$ rm logfile
[evatsoppa@fedora ~]$ ls
abc1      file.txt  my_os    work        Загрузки  Общедоступные
bin       may      reports  Видео       Изображения  'Рабочий стол'
conf.txt  monthly  ski.plases  Документы  Музыка  Шаблоны
```

Запускаю в фоновом режиме редактор mousepad, так как редактора gedit у меня нет

```
[evatsoppa@fedora ~]$ mousepad &
[1] 3247
```

Определяю идентификатор процесса mousepad, используя команду ps, конвейер и фильтр grep.

```
[evatsoppa@fedora ~]$ pgrep mousepad
[evatsoppa@fedora ~]$ ps aux | grep mousepad
evatsop+   3453   0.0   0.0 222568   2432 pts/2    S+   12:54   0:00 grep --color=auto mousepad
[evatsoppa@fedora ~]$ ps aux | grep mousepad | grep -v grep
```

Изучаю справку команды kill

```
KILL(1) User Commands

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 can 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, 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 and --queue options, and the possibility to specify processes by command name, are local extensions.

    If signal is 0, then no actual signal is sent, but error checking is still performed.

ARGUMENTS
    The list of processes to be signaled can be a mixture of names and PIDs.

    pid
        Each pid can be expressed in one of the following ways:
```

Удаляю процесс

```
$ kill 9361
```

Изучаю справку команды df и выполняю её,использовав утилиту -iv, которая позволяет увидеть информацию об инодах и сделать вывод читаемым.игнорирую сообщение системы о нем,так мы узнаем,сколько у нашей системы места

```
[evatsoppa@fedora ~]$ man df
[evatsoppa@fedora ~]$ df -vi
Файловая система  Инодов  ИИспользовано  ИСвободно  ИИспользовано%  Смонтировано
devtmpfs          463252          516        462736          1% /dev
tmpfs             468276           2        468274          1% /dev
tmpfs            819200          965        818235          1% /run
/dev/sda3           0              0           0            - /
tmpfs            1048576          48       1048528          1% /tmp
/dev/sda3           0              0           0            - /home
/dev/sda2          65536          394        65142          1% /boot
tmpfs             93655          147        93508          1% /run
```

Изучаю справку команды du и выполняю её,использовав утилиту -a,которая позволяет увидеть,сколько памяти занимают все файлы


```
$ du -a
```

Выполняю команду du

```
4      ./local/share/pnpm/global/5/.pnpm/rxjs@7.8.1/node_modules/rxjs/dist/cjs/internal/observable/fromEvent.js
4      ./local/share/pnpm/global/5/.pnpm/rxjs@7.8.1/node_modules/rxjs/dist/cjs/internal/observable/fromEventPattern.js
4      ./local/share/pnpm/global/5/.pnpm/rxjs@7.8.1/node_modules/rxjs/dist/cjs/internal/observable/fromSubscribable.js
4      ./local/share/pnpm/global/5/.pnpm/rxjs@7.8.1/node_modules/rxjs/dist/cjs/internal/observable/generate.js
4      ./local/share/pnpm/global/5/.pnpm/rxjs@7.8.1/node_modules/rxjs/dist/cjs/internal/observable/iif.js
12     ./local/share/pnpm/global/5/.pnpm/rxjs@7.8.1/node_modules/rxjs/dist/cjs/internal/observable/innerFrom.js
4      ./local/share/pnpm/global/5/.pnpm/rxjs@7.8.1/node_modules/rxjs/dist/cjs/internal/observable/interval.js
4      ./local/share/pnpm/global/5/.pnpm/rxjs@7.8.1/node_modules/rxjs/dist/cjs/internal/observable/merge.js
4      ./local/share/pnpm/global/5/.pnpm/rxjs@7.8.1/node_modules/rxjs/dist/cjs/internal/observable/never.js
4      ./local/share/pnpm/global/5/.pnpm/rxjs@7.8.1/node_modules/rxjs/dist/cjs/internal/observable/of.js
4      ./local/share/pnpm/global/5/.pnpm/rxjs@7.8.1/node_modules/rxjs/dist/cjs/internal/observable/onErrorResumeNext.js
4      ./local/share/pnpm/global/5/.pnpm/rxjs@7.8.1/node_modules/rxjs/dist/cjs/internal/observable/pairs.js
4      ./local/share/pnpm/global/5/.pnpm/rxjs@7.8.1/node_modules/rxjs/dist/cjs/internal/observable/partition.js
4      ./local/share/pnpm/global/5/.pnpm/rxjs@7.8.1/node_modules/rxjs/dist/cjs/internal/observable/race.js
4      ./local/share/pnpm/global/5/.pnpm/rxjs@7.8.1/node_modules/rxjs/dist/cjs/internal/observable/range.js
4      ./local/share/pnpm/global/5/.pnpm/rxjs@7.8.1/node_modules/rxjs/dist/cjs/internal/observable/throwError.js
4      ./local/share/pnpm/global/5/.pnpm/rxjs@7.8.1/node_modules/rxjs/dist/cjs/internal/observable/timer.js
4      ./local/share/pnpm/global/5/.pnpm/rxjs@7.8.1/node_modules/rxjs/dist/cjs/internal/observable/using.js
4      ./local/share/pnpm/global/5/.pnpm/rxjs@7.8.1/node_modules/rxjs/dist/cjs/internal/observable/zip.js
4      ./local/share/pnpm/global/5/.pnpm/rxjs@7.8.1/node_modules/rxjs/dist/cjs/internal/observable/bindCallback.js.map
4      ./local/share/pnpm/global/5/.pnpm/rxjs@7.8.1/node_modules/rxjs/dist/cjs/internal/observable/bindCallbackInternals.js.map
4      ./local/share/pnpm/global/5/.pnpm/rxjs@7.8.1/node_modules/rxjs/dist/cjs/internal/observable/bindNodeCallback.js.map
4      ./local/share/pnpm/global/5/.pnpm/rxjs@7.8.1/node_modules/rxjs/dist/cjs/internal/observable/combineLatest.js.map
4      ./local/share/pnpm/global/5/.pnpm/rxjs@7.8.1/node_modules/rxjs/dist/cjs/internal/observable/concat.js.map
4      ./local/share/pnpm/global/5/.pnpm/rxjs@7.8.1/node_modules/rxjs/dist/cjs/internal/observable/connectable.js.map
4      ./local/share/pnpm/global/5/.pnpm/rxjs@7.8.1/node_modules/rxjs/dist/cjs/internal/observable/ConnectableObservable.js.map
4      ./local/share/pnpm/global/5/.pnpm/rxjs@7.8.1/node_modules/rxjs/dist/cjs/internal/observable/defer.js.map
```

Вывела имена всех директорий, имеющихсх в моем домашнем каталоге, используя аргумент d у утилиты find опции -туре, то есть указываю тип файлов,который мне нужен и этот тип Директория. Утилита -а позволит увидеть размер всех файлов, а не только директорий

```
./local/share/pnpm/global/5/.pnpm/rxjs@7.8.1/node_modules/rxjs/dist/types/internal/scheduler
./local/share/pnpm/global/5/.pnpm/rxjs@7.8.1/node_modules/rxjs/dist/types/internal/operators
./local/share/pnpm/global/5/.pnpm/rxjs@7.8.1/node_modules/rxjs/dist/types/internal/scheduled
./local/share/pnpm/global/5/.pnpm/rxjs@7.8.1/node_modules/rxjs/dist/types/internal/observable
./local/share/pnpm/global/5/.pnpm/rxjs@7.8.1/node_modules/rxjs/dist/types/internal/observable/dom
./local/share/pnpm/global/5/.pnpm/rxjs@7.8.1/node_modules/rxjs/dist/types/operators
./local/share/pnpm/global/5/.pnpm/rxjs@7.8.1/node_modules/rxjs/dist/types/webSocket
./local/share/pnpm/global/5/.pnpm/rxjs@7.8.1/node_modules/rxjs/dist/bundles
./local/share/pnpm/global/5/.pnpm/rxjs@7.8.1/node_modules/rxjs/operators
./local/share/pnpm/global/5/.pnpm/rxjs@7.8.1/node_modules/rxjs/webSocket
./local/share/pnpm/global/5/.pnpm/is-windows@1.0.2
./local/share/pnpm/global/5/.pnpm/is-windows@1.0.2/node_modules
./local/share/pnpm/global/5/.pnpm/is-windows@1.0.2/node_modules/is-windows
./local/share/pnpm/global/5/.pnpm/tslib@2.6.2
./local/share/pnpm/global/5/.pnpm/tslib@2.6.2/node_modules
./local/share/pnpm/global/5/.pnpm/tslib@2.6.2/node_modules/tslib
./local/share/pnpm/global/5/.pnpm/tslib@2.6.2/node_modules/tslib/modules
./local/share/pnpm/global/5/.pnpm/merge@2.1.1
./local/share/pnpm/global/5/.pnpm/merge@2.1.1/node_modules
./local/share/pnpm/global/5/.pnpm/merge@2.1.1/node_modules/merge
./local/share/pnpm/global/5/.pnpm/merge@2.1.1/node_modules/merge/lib
./local/share/pnpm/global/5/.pnpm/merge@2.1.1/node_modules/merge/lib/src
```

Выводы

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

Ответы на 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.
- 2.Объясните разницу между операцией > и ». Этот знак > - перенаправление ввода/вывода, а » - перенаправление в режиме добавления.
- 3.Что такое конвейер? Конвейер (pipe) служит для объединения простых команд или утилит в цепочки, в которых результат работы предыдущей команды передаётся последующей.
- 4.Что такое процесс? Чем это понятие отличается от программы? Главное отличие между программой и процессом заключается в том, что программа - это набор инструкций, который позволяет ЦПУ выполнять определенную задачу, в то время как процесс - это исполняемая программа.
- 5.Что такое PID и GID? PPID - (parent process ID) идентификатор родительского процесса. Процесс может порождать и другие процессы. UID, GID - реальные идентификаторы пользователя и его группы, запустившего данный процесс.
- 6.Что такое задачи и какая команда позволяет ими управлять? 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? Команда htop похожа на команду top по выполняемой функции: они обе показывают информацию о процессах в реальном времени, выводят данные о потреблении системных ресурсов и позволяют искать, останавливать и управлять процессами.

У обеих команд есть свои преимущества. Например, в программе htop реализован очень удобный поиск по процессам, а также их фильтрация. В команде top это не так удобно — нужно знать кнопку для вывода функции поиска.

Зато в top можно разделять область окна и выводить информацию о процессах в соответствии с разными настройками. В целом top намного более гибкая в настройке отображения процессов.

8.Назовите и дайте характеристику команде поиска файлов. Приведите примеры использования этой команды.

Команда find - это одна из наиболее важных и часто используемых утилит системы Linux. 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.

Утилита find предустановлена по умолчанию во всех Linux дистрибутивах, поэтому вам не нужно будет устанавливать никаких дополнительных пакетов. Это очень важная находка для тех, кто хочет использовать командную строку наиболее эффективно.

Команда find имеет такой синтаксис: find [папка] [параметры] критерий шаблон [действие] Пример: find /etc -name "p*" -print

9.Можно ли по контексту (содержанию) найти файл? Если да, то как? find / -type f -exec grep -H 'текстДляПоиска' {} ;

10.Как определить объем свободной памяти на жёстком диске? С помощью команды df -h.

11.Как определить объем вашего домашнего каталога? С помощью команды du -s.

12.Как удалить зависший процесс? С помощью команды kill% номер задачи.

Список литературы{.unnumbered}

Лабораторная работа №8