



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных
технологий

Отчет по практическим работам №5-9

по дисциплине «Системное программное обеспечение»

Выполнил:

Студент группы ИВБО-11-23

Туктаров Т.А.

Проверил:

Ст. преп. Овчинникова М.А.

МОСКВА 2025 г.

1 ПРАКТИЧЕСКАЯ РАБОТА 5

Задание

Используя утилиту sed выполнить следующие задачи:

- Вывести только строки, содержащие слово «SUCCESS» Удалить все пустые строки.
- Выделить строки, содержащие IP-адреса, добавив в начало символ «!».
- Вставить строку «-- End of section --» после строки, содержащей «Section End».
- Вставить строку «Insert Here» после пятой строки.
- Добавить номер текущей строки в квадратных скобках для каждой строки.

При помощи утилит sed реализовать поиск информации, хранящейся в файле MOCK_DATA.csv. Результаты должны выводиться в файл, в вывод необходимо записать найденные строки с их номерами. Для данного варианта надо найти IP со значением 141 во втором октете.

При помощи утилиты sed с флагом -n и команды p с указанием искомой последовательности ищем нужные строки.

```
[tim@ladego ~]$ sed -n '/SUCCESS/p' SED_FILE.txt  
SUCCESS: Backup completed at 2024-02-24 23:59:59
```

Для добавления ! в начала строки воспользуемся регулярным выражением, а также якорем ^.

```
sed '/[0-9]\+\.[0-9]\+\.[0-9]\+\.[0-9]\+\/s/^/!/' SED_FILE.txt
```

```
# IP-адреса  
!127.0.0.1 - Localhost  
!192.168.1.1 - Router  
!255.255.255.255 - Broadcast
```

Для добавления строки воспользуемся командой a.

```
sed '/Section End/a -- End of section --' SED_FILE.txt
```

Чтобы вставить строчку, воспользуемся командой a и укажем номер, куда надо вставить значение

```
[tim@ladego ~]$ sed '5a Insert Here' SED_FILE.txt
# Лог-файл сервера с разными сообщениями
INFO: Server started successfully
ERROR: Connection timeout while connecting to 192.168.1.100
WARN: Low disk space on /dev/sda1
SUCCESS: Backup completed at 2024-02-24 23:59:59
Insert Here
INFO: New user registered - user@example.com

# JSON-данные с ошибками формата
{ key1:value1, key2: value2, key3: "value3" }
{ "correct_key": "value", "another_key":12345 }
```

Для вставки номеров воспользуемся туннелированием значений из вывода утилиты sed с командой = и вставим значения в скобки, затем объединим.

```
[tim@ladego ~]$ sed '=' SED_FILE.txt | sed 'N;s/\n/] /; s/^/[/'
[1] # Лог-файл сервера с разными сообщениями
[2] INFO: Server started successfully
[3] ERROR: Connection timeout while connecting to 192.168.1.100
[4] WARN: Low disk space on /dev/sda1
[5] SUCCESS: Backup completed at 2024-02-24 23:59:59
[6] INFO: New user registered - user@example.com
[7]
[8] # JSON-данные с ошибками формата
[9] { key1:value1, key2: value2, key3: "value3" }
[10] { "correct_key": "value", "another_key":12345 }
[11]
[12] # CSV-данные
[13] 1,John Doe,john.doe@example.com,35,New York
[14] 2,Jane Smith,jane.smith@domain.com,28,Los Angeles
[15] 3,Bob Brown,bob.brown@somewhere.org,45,Chicago
[16]
```

Для получения значений из файла используется аналогичный подход с регулярными выражениями, а затем добавление номера страниц

```
sed -n '/\<[0-9]\+\.\.141\.\.[0-9]\+\.\.[0-9]\+\>/' MOCK_DATA.csv > output.txt
```

Вывод в output.txt:

```
491
593
864
```

2 ПРАКТИЧЕСКАЯ РАБОТА 6

Задание

Написать сценарии командной строки:

1. Сценарий должен анализировать журнал systemd (journalctl) и выводит список ошибок (ERROR) и предупреждений (WARN). Подсказка: вывод journalctl можно анализировать через пайп "|" и команды анализа текста.
2. Сценарий должен выводить самый большой файл в указанной директории и полный путь до него. При поиске должны учитываться и поддиректории.

При помощи только циклов и функций реализовать поиск информации, хранящейся в файле MOCK_DATA.csv. Результаты должны выводиться в файл, в вывод необходимо записать номер строки, а также найденное значение. Найти строки, где IP со значением 141 во втором октете.

Задание 1, первый сценарий (bash-файл и его вывод):

```
#!/bin/bash

# 1. Анализ journalctl на наличие ERROR и WARN
echo "Ошибки (ERROR) и предупреждения (WARN) из journalctl"
journalctl --no-pager -p 0..4 -xb 2>/dev/null | grep -E -i 'error|warn' | sort | uniq -c | sort -nr

[tim@ladego ~]$ ./bash_file1.sh
Ошибки (ERROR) и предупреждения (WARN) из journalctl
1 apr 13 13:58:32 ladego ModemManager[2489]: <warn> [plugin/generic] could not grab port ttyS0: Cannot add port 'tty/ttyS0', unhandled port type
1 apr 13 13:58:32 ladego ModemManager[2489]: <warn> [base-manager] couldn't create modem for device '/sys/devices/pci0000:00/0000:00:16.3': Failed to find primary AT port
```

Задание 1, второй сценарий (bash-файл и его вывод):

```
#!/bin/bash

# 2. Поиск самого большого файла в указанной директории
read -p "Введите путь к директории: " dir

if [ -d "$dir" ]; then
    largest_file=$(find "$dir" -type f -exec du -h {} + 2>/dev/null | sort -rh | head -n 1 | cut -f2)

    if [ -n "$largest_file" ]; then
        echo "Самый большой файл: $largest_file"
    else
        echo "В указанной директории нет файлов."
    fi
else
    echo "Ошибка: директория '$dir' не существует или недоступна."
fi

[tim@ladego ~]$ ./bash_file2.sh
Введите путь к директории: ./Загрузки/
Самый большой файл: ./Загрузки/virtualbox-7.1_7.1.6-167084_Ubuntu_oracular_amd64.deb
```

Задание 2 (bash-файл и вывод):

```
#!/bin/bash

function search_ips {
    local input_file="$1"
    local output_file="$2"
    local line_number=0

    > "$output_file"

    while IFS= read -r line; do
        ((line_number++))
        ip=$(echo "$line" | cut -d',' -f1)
        if [[ "$ip" =~ ^[0-9]+\.[0-9]+\.[0-9]+\.[0-9]* ]]; then # [0-9]+\.[0-9]+\.[0-9]+\.[0-9]*
            echo "Строка $line_number: $line" >> "$output_file"
            echo "Строка $line_number: $line"
        fi
    done < "$input_file"
}

input_file="MOCK_DATA.csv"
output_file="output.txt"

if [ ! -f "$input_file" ]; then
    echo "Ошибка: файл $input_file не найден!"
    exit 1
fi

search_ips "$input_file" "$output_file"

echo "Поиск завершен. Результаты сохранены в $output_file"
```

```
[tim@ladego ~]$ cat output.txt
Строка 492: 47.141.75.208,VitaeConsectetuerEget.xls,https://gov.uk/quis/odio/consequat.json,eremnantdm@gov.uk
Строка 593: 122.141.155.143,Augue.xls,https://cnet.com/sociis/natoque/penatibus/et/magnis.html,jillistongf@histats.com
Строка 864: 45.141.30.20,DapibusNulla.mp3,https://rakuten.co.jp/pulvinar/sed/nisl/nunc/rhonus.jsp,bnibloeny@wiley.com
```

3 ПРАКТИЧЕСКАЯ РАБОТА 7

Задание

Согласно варианту создайте сценарии, а затем поместите их в crontab с указанным расписанием и сохранением вывода в указанный файл.

1. Сценарий при помощи утилиты ps выводит в терминал процессы, принадлежащие текущему пользователю. Вывод должен начинаться с текущего времени, а также должен содержать название столбцов ps. Работает раз в 30 минут с субботы по воскресенье.
2. Сценарий логирования загруженности процессора. Вывод должен содержать время логирования, а также информацию об общей загруженности процессора. Информацию о загруженности процессора можно найти в файле /proc/stat или в любой утилите мониторинга системы. Работает раз в 2 минуты с 1 по 6 месяц и раз в 1 минуту с 7 по 12 месяц.

Задание 1. Здесь в bash-файле надо просто вывести текущее время, а затем вызвать команду ps

```
#!/bin/bash

CURRENT_USER=$(whoami)

echo "Текущее время: $(date '+%Y-%m-%d %H:%M:%S')"
ps -u "$CURRENT_USER" -f 2>/dev/null
```

А crontab будет выглядеть вот так

```
*/30 * * * 6-7 user_processes.sh
```

Вывод скрипта:

```

Текущее время: 2025-04-15 02:35:32
UID      PID    PPID   C  STIME TTY      TIME CMD
tim      3155    1      0  anp14 ?    00:00:00 /lib/systemd/systemd --user
tim      3157    3155   0  anp14 ?    00:00:00 (sd-pam)
tim      3169    1      0  anp14 ?    00:00:00 /usr/lib/kf5/bin/kwalletd5 --pam-login 8 9
tim      3170    3147   0  anp14 ?    00:00:00 /usr/lib/kf5/bin/startplasma-x11
tim      3213    1      0  anp14 ?    00:00:00 ssh-agent -u
tim      3228    1      0  anp14 ?    00:00:00 /usr/bin/xdmcp-xauthority-hostname-updater -d
tim      3273    3155   0  anp14 ?    00:00:00 /bin/dbus-daemon --session --address=systemd: --nofork --nopidfile --systemd-activation --syslog-only
tim      3300    3155   0  anp14 ?    00:00:00 /usr/libexec/bluetooth/obexd
tim      3318    3155   0  anp14 ?    00:00:01 /usr/lib/kf5/bin/ksmserver
tim      3320    3155   0  anp14 ?    00:01:52 /usr/lib/kf5/bin/kded5
tim      3321    3155   2  anp14 ?    00:08:36 /usr/lib/kf5/bin/kwin_x11 --replace
tim      3367    3155   0  anp14 ?    00:00:01 /usr/lib/kf5/bin/kglobalaccel5
tim      3375    3155   0  anp14 ?    00:01:31 /usr/lib/kf5/bin/plasmashell --no-respond
tim      3383    3155   0  anp14 ?    00:00:00 /usr/libexec/dconf-service
tim      3403    3155   0  anp14 ?    00:00:00 /usr/libexec/kf5/kactivitymanagerd
tim      3411    3155   0  anp14 ?    00:00:00 /usr/lib/kf5/bin/gmenudbusmenuproxy
tim      3412    3155   0  anp14 ?    00:00:00 /usr/libexec/kf5/polkit-kde-authentication-agent-1
tim      3413    3155   0  anp14 ?    00:00:05 /usr/libexec/kf5/org_kde_powerdevil
tim      3414    3155   0  anp14 ?    00:00:00 /usr/lib/kf5/bin/xembedsniproxy
tim      3415    3155   0  anp14 ?    00:00:00 /usr/bin/ssh-agent -D -a /run/user/500/ssh-agent.socket
tim      3489    3155   0  anp14 ?    00:00:02 /usr/libexec/kf5/kdeconnectd
tim      3496    3155   0  anp14 ?    00:00:01 /usr/lib/kf5/bin/kaccess
tim      3501    3155   0  anp14 ?    00:00:02 /usr/libexec/DiscoverNotifier
tim      3502    3155   0  anp14 ?    00:00:01 /usr/lib/kf5/bin/kalendarac
tim      3507    3155   0  anp14 ?    00:00:00 /usr/libexec/kf5/kscreen_backend_launcher
tim      3527    3155   0  anp14 ?    00:00:00 /usr/bin/xsettingsd
tim      3768    3155   0  anp14 ?    00:00:00 /usr/bin/pulseaudio --start --log-target=syslog
tim      3911    3375   1  anp14 ?    00:06:20 /usr/lib64/chromium-gost/chromium-gost --use-gl=any --disable-features=AutofillUseParameterizedSectioning

```

Задание 2. В Bash-файле с помощью `grep` будет считываться кол-во всех процессов, а затем при помощи простой формулы высчитываться степень нагрузки, и выводиться пользователю.

```

#!/bin/bash

get_cpu_usage() {
    read -r cpu user nice system idle iowait irq softirq steal guest guest_nice <<(grep '^cpu ' /proc/stat)

    total=$((user + nice + system + idle + iowait + irq + softirq + steal))
    idle_total=$((idle + iowait))
    usage=$((100 - (idle_total * 100) / total))

    echo "$usage%"
}

log_entry="[$(date '+%Y-%m-%d %H:%M:%S')] CPU Usage: $(get_cpu_usage)"

echo "$log_entry"

```

crontab для данного bash-файла будет выглядеть так:

```

*/2 * * 1-6 * cpu_logger.sh
* * * 7-12 * cpu_logger.sh

```

Пример работы:

```
[2025-04-14 22:35:42] CPU Usage: 8%
```


4 ПРАКТИЧЕСКАЯ РАБОТА 8

Задание

1. Установите пакеты **git** и **btcp** (запустите их после установки для проверки её корректности);
2. Удалите пакет **git** без удаления конфигурационных файлов;
3. Удалить пакет **btcp** с удалением конфигурационных файлов
4. Найдите пакет **tomcat** в дереве репозитория, после чего выведите информацию о пакете;
5. Установите **Docker**, следуя официальной документации <https://docs.docker.com/engine/install/ubuntu/>.

Для того, чтобы установить пакеты **git** и **btcp** нужно воспользоваться командой `apt install` с правами корневого пользователя.

```
[tim@ladego ~]$ sudo apt-get install git
Чтение списков пакетов... Завершено
Построение дерева зависимостей... Завершено
Следующие дополнительные пакеты будут установлены:
  diffstat git-core perl-Error perl-Git perl-TermReadKey perl-libintl
Следующие НОВЫЕ пакеты будут установлены:
  diffstat git git-core perl-Error perl-Git perl-TermReadKey perl-libintl
0 будет обновлено, 7 новых установлено, 0 пакетов будет удалено и 358 не будет обновлено.
Необходимо получить 4408kB архивов.
После распаковки потребуется дополнительно 26,3МБ дискового пространства.
Продолжить? [Y/n] y
Получено: 1 http://ftp.altlinux.org p10/branch/x86_64/classic diffstat 1.64-alt1:sisyphus+278100.700.1.1@1626056683 [35,6kB]
Получено: 2 http://ftp.altlinux.org p10/branch/noarch/classic perl-Error 0.17029-alt1:sisyphus+245975.100.1.1@1581530233 [18,5kB]
Получено: 3 http://ftp.altlinux.org p10/branch/x86_64/classic perl-libintl 1.32-alt1:sisyphus+279723.5100.1.1@1626650309 [155kB]
Получено: 4 http://ftp.altlinux.org p10/branch/x86_64/classic perl-TermReadKey 2.38-alt1.1:sisyphus+279723.1000.1.1@1626649040 [22,7kB]
Получено: 5 http://ftp.altlinux.org p10/branch/x86_64/classic git-core 2.42.2-alt1:p10+350723.100.3.1@1718390536 [4084kB]
Получено: 6 http://ftp.altlinux.org p10/branch/noarch/classic perl-Git 2.42.2-alt1:p10+350723.100.3.1@1718390536 [62,0kB]
Получено: 7 http://ftp.altlinux.org p10/branch/x86_64/classic git 2.42.2-alt1:p10+350723.100.3.1@1718390536 [29,6kB]
Получено 4408kB за 0s (9086kB/s).
Совершаем изменения...
Подготовка... [100%]
Обновление / установка...
1: perl-TermReadKey-2.38-alt1.1 [14%]
2: perl-libintl-1.32-alt1 [29%]
3: perl-Error-0.17029-alt1 [43%]
4: diffstat-1.64-alt1 [57%]
5: git-core-2.42.2-alt1 [71%]
6: perl-Git-2.42.2-alt1 [86%]
7: git-2.42.2-alt1 [100%]
Завершено.
[tim@ladego ~]$
```

Аналогично устанавливается и пакет **btcp**. Этот пакет потребовал небольшого разрешения на занятие небольшого пространства памяти.

```
[tim@ladego ~]$ sudo apt-get install btcp
Чтение списков пакетов... Завершено
Построение дерева зависимостей... Завершено
Следующие НОВЫЕ пакеты будут установлены:
  btcp
0 будет обновлено, 1 новых установлено, 0 пакетов будет удалено и 358 не будет обновлено.
Необходимо получить 450kB архивов.
После распаковки потребуется дополнительно 1378kB дискового пространства.
Получено: 1 http://ftp.altlinux.org p10/branch/x86_64/classic btcp 1.2.12-alt1:p10+313112.100.1.1@1673349873 [450kB]
Получено 450kB за 0s (5083kB/s).
Совершаем изменения...
Подготовка... [100%]
Обновление / установка...
1: btcp-1.2.12-alt1 [100%]
Завершено.
```

Оба пакета установились успешно

Далее нужно удалить пакет **git** без удаления конфигурационных файлов. Делается это с помощью команды `remove`


```

[tim@ladego ~]$ sudo apt-get remove git
Чтение списков пакетов... Завершено
Построение дерева зависимостей... Завершено
Следующие пакеты будут УДАЛЕНЫ:
  git
0 будет обновлено, 0 новых установлено, 1 пакетов будет удалено и 358 не будет обновлено.
Необходимо получить 0В архивов.
После распаковки потребуется дополнительно 0В дискового пространства.
Продолжить? [Y/n] y
Совершаем изменения...
Подготовка... ##### [100%]
Очистка / удаление...
1: git-2.42.2-alt1 ##### [100%]
Завершено.

```

Далее нужно удалить пакет btop с конфигурационными файлами. Делается это с помощью команды remove и флага `--purge`.

```

[tim@ladego ~]$ sudo apt-get remove --purge btop
Чтение списков пакетов... Завершено
Построение дерева зависимостей... Завершено
Следующие пакеты будут УДАЛЕНЫ:
  btop*
0 будет обновлено, 0 новых установлено, 1 пакетов будет удалено и 358 не будет обновлено.
Необходимо получить 0В архивов.
После распаковки будет освобождено 1378kB дискового пространства.
Продолжить? [Y/n] y
Совершаем изменения...
Подготовка... ##### [100%]
Очистка / удаление...
1: btop-1.2.12-alt1 ##### [100%]
Завершено.

```

Чтобы найти пакет tomcat третьей версии воспользуемся командой `apt search`. Среди выведенных файлов есть и искомый пакет.

```

[tim@ladego ~]$ sudo apt-cache search tomcat
netdata - Real-time performance monitoring, done right!
pki-tps - Dogtag PKI TPS Package
tomcat-native - Tomcat native library
i586-tomcat-native - Tomcat native library
guacamole-client - Guacamole web application
resource-agents-tomcat - resource agent manages Tomcat
python3-module-salt - Management component for salt, a parallel remote execution system
sos - A set of tools to gather troubleshooting information from a system
tomcat - Apache Servlet/JSP Engine, RI for Servlet 4.0/JSP 2.3 API
tomcat-admin-webapps - The host-manager and manager web applications for Apache Tomcat
tomcat-docs-webapp - The docs web application for Apache Tomcat
tomcat-el-3.0-api - Apache Tomcat Expression Language v3.0 API Implementation Classes
tomcat-jsp-2.3-api - Apache Tomcat JavaServer Pages v2.3 API Implementation Classes
tomcat-lib - Libraries needed to run the Tomcat Web container
tomcat-servlet-4.0-api - Apache Tomcat Java Servlet v4.0 API Implementation Classes
tomcat-webapps - The ROOT web application for Apache Tomcat
tomcat-taglibs-parent - Apache Taglibs Parent
tomcat-taglibs-standard - Apache Standard Taglib
tomcat-taglibs-standard-javadoc - Javadoc for tomcat-taglibs-standard
tomcat10 - Apache Servlet/JSP Engine, RI for Servlet 6.0/JSP 3.1 API
tomcat10-admin-webapps - The host-manager and manager web applications for Apache Tomcat
tomcat10-docs-webapp - The docs web application for Apache Tomcat
tomcat10-el-5.0-api - Apache Tomcat Expression Language v5.0 API Implementation Classes
tomcat10-jsp-3.1-api - Apache Tomcat JavaServer Pages v3.1 API Implementation Classes
tomcat10-lib - Libraries needed to run the Tomcat Web container
tomcat10-servlet-6.0-api - Apache Tomcat Java Servlet v6.0 API Implementation Classes
tomcat10-webapps - The ROOT web application for Apache Tomcat
tomcatjss - JSSE module for Apache Tomcat that uses JSS

```

Информацию о пакете можно вывести при помощи `apt show`.

```

tim@ladego:~$ sudo apt-cache show tomcat
Package: tomcat
Section: System/Servers
Installed Size: 342839
Maintainer: Sergey Gvozdetskiy <serjigva@altlinux.org>
Version: 1:9.0.98-alt0_1jpp17.p10.1:p10+371974.2460.17.1@1742466531
Pre-Depends: shadow-change, shadow-check, shadow-convert, shadow-edit, shadow-groups, shadow-submap, shadow-utils, /bin/sh, /usr/sbin/groupadd, /usr/sbin/useradd, glibc-utils, /usr/sbin/post_service, /bin/rm, /usr/sbin/preun_service, rpmlib(PayloadIsLzma)
Depends: javapackages-tools, tomcat-native (>= 1.2.21), /bin/bash, /bin/sh, /etc/logrotate.d, /etc/rc.d/init.d, /lib/lsb/init-functions, /usr/share/java-utils, /java-functions, coreutils, sed, tomcat-lib (= 1:9.0.98-alt0_1jpp17.p10.1:p10+371974.2460.17.1)
Conflicts: tomcat10
Provides: tomcat-log4j (= 1:9.0.98-alt0_1jpp17.p10.1), tomcat (= 1:9.0.98-alt0_1jpp17.p10.1:p10+371974.2460.17.1)
Architecture: noarch
Size: 88461
MD5Sum: fcfce150b934da47c6361f1ae1f18fd2
Filename: tomcat-9.0.98-alt0_1jpp17.p10.1.noarch.rpm
Description: Apache Servlet/JSP Engine, RI for Servlet 4.0/JSP 2.3 API
 Tomcat is the servlet container that is used in the official Reference
 Implementation for the Java Servlet and JavaServer Pages technologies.
 The Java Servlet and JavaServer Pages specifications are developed by
 Sun under the Java Community Process.

 Tomcat is developed in an open and participatory environment and
 released under the Apache Software License version 2.0. Tomcat is intended
 to be a collaboration of the best-of-breed developers from around the world.
Changelog:
 * 4т map 20 2025 Sergey Gvozdetskiy <serjigva@altlinux.org> 1:9.0.98-alt0_1jpp17.p10.1
 - NMU: new version
 - suppress unmet lsb-init symlink
 - security fixes:
   + CVE-2024-23672: Denial of Service
   + CVE-2024-24549: Denial of Service

```

Для установки Docker будем следовать инструкциям. Изначально нужно установить репозиторий Docker.

```

timur@DESKTOP-CSJARVU:~$ # Add Docker's official GPG key:
sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

# Add the repository to Apt sources:
echo \
 "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu \
 $(. /etc/os-release && echo "${UBUNTU_CODENAME:-$VERSION_CODENAME}") stable" | \
 sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
[sudo] password for timur:
Sorry, try again.
[sudo] password for timur:
Hit:1 http://archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:4 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:5 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [992 kB]
Get:6 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [748 kB]
Get:7 http://archive.ubuntu.com/ubuntu noble-updates/main Translation-en [219 kB]
Get:8 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [151 kB]
Get:9 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [13.5 kB]
Get:10 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1054 kB]
Get:11 http://archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [266 kB]
Get:12 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [367 kB]
Get:13 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [26.0 kB]

```

Далее можно приступить к установке.

```
timur@DESKTOP-C5JARVU:~$ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  libllvm17t64
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  docker-ce-rootless-extras iptables libip4tc2 libip6tc2 libltdl7 libnetfilter-contrack3 libnfnetlink0 libnftables1
  libnftnl11 libslirp0 nftables pigz slirp4netns
Suggested packages:
  cgroupfs-mount | cgroup-lite firewallld
The following NEW packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-cli docker-ce-rootless-extras docker-compose-plugin iptables
  libip4tc2 libip6tc2 libltdl7 libnetfilter-contrack3 libnfnetlink0 libnftables1 libnftnl11 libslirp0 nftables pigz
  slirp4netns
0 upgraded, 18 newly installed, 0 to remove and 63 not upgraded.
Need to get 122 MB of archives.
After this operation, 446 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.ubuntu.com/ubuntu noble/universe amd64 pigz amd64 2.8-1 [65.6 kB]
Get:2 https://download.docker.com/linux/ubuntu noble/stable amd64 containerd.io amd64 1.7.27-1 [30.5 MB]
```

Убедимся в правильности установки.

```
timur@DESKTOP-C5JARVU:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
e6590344b1a5: Pull complete
Digest: sha256:424f1f86cdf501deb591ace8d14d2f40272617b51b374915a87a2886b2025ece
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```


5 ПРАКТИЧЕСКАЯ РАБОТА 9

Задание

Выполните сборку и установку iproute2 из исходных кодов

Сперва необходимо установить исходные файлы:

```
timur@DESKTOP-C5JARVU:~$ git clone https://github.com/iproute2/iproute2.git
Cloning into 'iproute2'...
remote: Enumerating objects: 37620, done.
remote: Counting objects: 100% (4077/4077), done.
remote: Compressing objects: 100% (1292/1292), done.
remote: Total 37620 (delta 2996), reused 2850 (delta 2781), pack-reused 33543 (from 3)
Receiving objects: 100% (37620/37620), 13.70 MiB | 2.43 MiB/s, done.
Resolving deltas: 100% (27324/27324), done.
```

Репозиторий уже хранил файл configure. Переходим в директорию и запускаем его

```
timur@DESKTOP-C5JARVU:~/iproute2$ ./configure
TC schedulers
./configure: 31: pkg-config: not found

lib directory: /usr/lib
libc has setns: yes
libc has name_to_handle_at: yes
SELinux support: ./configure: 345: pkg-config: not found
no
libtirpc support: ./configure: 358: pkg-config: not found
no
libbpf support: ./configure: 287: pkg-config: not found
no
ELF support: ./configure: 220: pkg-config: not found
no
libmnl support: ./configure: 371: pkg-config: not found
no
Berkeley DB: no
need for strlcpy: no
libcap support: ./configure: 430: pkg-config: not found
no
color output: never
```

После успешной установки зависимостей начнем сборку пакета.

```
timur@DESKTOP-C5JARVU:~/iproute2$ make
lib
  CC      libgenl.o
  CC      libnetlink.o
libnetlink.c:152:2: warning: #warning "libmnl required for error support" [-Wcpp]
 152 | #warning "libmnl required for error support"
      | ^~~~~~
  AR      libnetlink.a
  CC      utils.o
  CC      utils_math.o
  CC      rt_names.o
  CC      ll_map.o
  CC      ll_types.o
  CC      ll_proto.o
  CC      ll_addr.o
  CC      inet_proto.o
  CC      namespace.o
  CC      json_writer.o
  CC      json_print.o
  CC      json_print_math.o
  CC      names.o
  CC      color.o
  CC      bpf_legacy.o
  CC      bpf_glue.o
  CC      exec.o
  CC      fs.o
  CC      cg_map.o
  CC      ppp_proto.o
```

Установим собранный пакет

```
timur@DESKTOP-C5JARVU:~/iproute2$ sudo make install
[sudo] password for timur:

lib
make[1]: Nothing to be done for 'all'.

ip
make[1]: Nothing to be done for 'all'.

tc
  YACC      emp_ematch.tab.c
/bin/sh: 1: bison: not found
make[1]: *** [Makefile:161: emp_ematch.tab.c] Error 127
make: *** [Makefile:81: all] Error 2
```

Проверим корректность работы установленного приложения

```
timur@DESKTOP-C5JARVU:~/iproute2$ ip route
default via 172.17.112.1 dev eth0 proto kernel
172.17.112.0/20 dev eth0 proto kernel scope link src 172.17.120.122
172.18.0.0/16 dev docker0 proto kernel scope link src 172.18.0.1 linkdown
```