

# Домашнее задание #7

---

## Критерии оценки

- 0 баллов - выполнено менее 60% пунктов задания.
- 2 балла - выполнено от 60% до 69% пунктов задания, отчет соответствует требованиям.
- 4 балла - выполнено от 70% до 89% пунктов задания, отчет соответствует требованиям.
- 6 баллов - выполнено более 90% пунктов задания.

## Оформление отчета

Отчет должен включать в себя файл `typescript.txt` полученный в результате выполнения команды:

```
$ script typescript.txt
```

Команду выше необходимо запустить перед выполнением домашнего задания. Для завершения записи достаточно выполнить команду `exit`. При выполнении команды `script` история команд Bash не сохраняется.

*Примечание.* Вместо `???` необходимо подставить нужную команду, чтобы решить задачу.

## Работа с настроенными deb-репозиториями

1. Выведите список доступных репозиториев. Определите их количество.

```
$ sudo grep ^ /etc/apt/sources.list /etc/apt/sources.list.d/*
```

2. Выведите список установленных пакетов и подсчитайте их количество.

```
$ sudo apt list --installed | ???
```

3. Обновите базу данных пакетов `apt`.

```
$ sudo apt update
```

4. Определите, установлен ли в системе пакет сканирования сети `nmap`.

```
$ sudo apt list --installed | ???
```

5. Найдите пакет `nmap` в доступных репозиториях.

```
$ sudo apt search nmap
```

6. Выведите информацию о пакете `nmap`.

```
$ sudo apt info nmap
```

7. Установите пакет `nmap`.

```
$ sudo apt install nmap
```

8. Выведите список установленных файлов из пакета `nmap`.

```
$ sudo dpkg-query -L nmap
```

9. Определите зависимости для пакета `nmap`.

```
$ sudo apt depends nmap
```

10. Попробуйте удалить пакет `nmap` с помощью утилиты `dpkg`.

```
$ sudo dpkg --remove nmap
```

11. Удалите пакет `nmap` и его зависимости. Сравните работу утилит `dpkg` и `apt`.

```
$ sudo apt autoremove nmap
```

## Подключение сторонних репозиториев

1. Обновите базу данных пакетов `apt`.

```
$ sudo apt update
```

2. Установите дополнительные утилиты для удобной работы с репозиториями.

```
$ sudo apt-get install ca-certificates curl
```

3. Создайте папку доверенных ключей.

```
$ sudo install -m 0755 -d /etc/apt/keyrings
```

4. Добавьте официальный GPG ключ от Docker.

```
$ 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
```

5. Добавьте репозиторий.

```
$ echo "deb [arch=amd64 signed-by=/etc/apt/keyrings/docker.asc]  
https://download.docker.com/linux/ubuntu jammy stable" | sudo tee  
/etc/apt/sources.list.d/docker.list > /dev/null
```

6. Обновите базу данных пакетов `apt`.

```
$ sudo apt update
```

7. Произведите установку утилит Docker.

```
$ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin  
docker-compose-plugin
```

8. Для проверки запустите контейнер `hello-world`. В случае успешной установки команда напечатает подтверждающее сообщение.

```
$ sudo docker run hello-world
```

## Смена зеркал репозитория по умолчанию

1. Создайте бэкап файла `/etc/apt/sources.list`.

```
$ sudo cp /etc/apt/sources.list /etc/apt/sources.list.backup.
```

2. Ознакомьтесь со списком доступных репозиториев <https://launchpad.net/ubuntu/+archivemirrors>.
3. В файле `/etc/apt/sources.list` осуществите замену всех текущих ссылок на ссылки зеркала.  
Например:

```
deb http://ru.archive.ubuntu.com/ubuntu
```

на

```
deb http://mirror.yandex.ru/ubuntu
```

4. Обновите базу данных пакетов `apt`.

```
$ sudo apt update
```