

Задание 1 – установить MongoDB.

*создать таблицу data; создать пользователя manager, у которого будет доступ только на чтение этой таблицы.

Добавляем репу

```
anex13@anextmshwsrv:~$ echo "deb [ arch=amd64,arm64 signed-by=/usr/share/keyrings/mongodb-server-7.0.gpg ] https://repo.mongodb.org/apt/ubuntu jammy/mongodb-org/7.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-7.0.list
```

Добавляем ключ

```
anex13@anextmshwsrv:~$ curl -fsSL https://www.mongodb.org/static/pgp/server-7.0.asc | \
  sudo gpg -o /usr/share/keyrings/mongodb-server-7.0.gpg \
  --dearmor
anex13@anextmshwsrv:~$
```

Обновляем кэш репозитория командой `sudo apt update`, видим что новый работает.

```
anex13@anextmshwsrv:~$ sudo apt update
Hit:1 http://by.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://by.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://by.archive.ubuntu.com/ubuntu jammy-backports InRelease
Ign:4 https://repo.mongodb.org/apt/ubuntu jammy/mongodb-org/7.0 InRelease
Hit:5 http://security.ubuntu.com/ubuntu jammy-security InRelease
Get:6 https://repo.mongodb.org/apt/ubuntu jammy/mongodb-org/7.0 Release [2,090 B]
Get:7 https://repo.mongodb.org/apt/ubuntu jammy/mongodb-org/7.0 Release.gpg [866 B]
Get:8 https://repo.mongodb.org/apt/ubuntu jammy/mongodb-org/7.0/multiverse amd64 Packages [35.0 kB]
Get:9 https://repo.mongodb.org/apt/ubuntu jammy/mongodb-org/7.0/multiverse arm64 Packages [33.9 kB]
Fetched 71.8 kB in 2s (46.0 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
3 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

Устанавливаем

```
anex13@anextmshwsrv:~$ sudo apt install -y mongodb-org
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
```

Запускаем проверяем

```
anex13@anextmshwsrv:~$ sudo service mongod start
anex13@anextmshwsrv:~$ sudo service mongod status
● mongod.service - MongoDB Database Server
   Loaded: loaded (/lib/systemd/system/mongod.service; disabled; vendor preset: enabled)
   Active: active (running) since Tue 2024-04-09 09:57:44 UTC; 9s ago
     Docs: https://docs.mongodb.org/manual
   Main PID: 8440 (mongod)
    Memory: 72.8M
      CPU: 814ms
    CGroup: /system.slice/mongod.service
            └─8440 /usr/bin/mongod --config /etc/mongod.conf
```

Заходим в оболочку монго

```
anex13@anextmshwsrv:~$ mongosh
Current MongoDB Log ID: 661511797b8f6b37abef634a
Connecting to: mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh
Using MongoDB: 7.0.8
Using Mongosh: 2.2.3

For mongosh info see: https://docs.mongodb.com/mongodb-shell/
```

Создаем и переходим в таблицу data

```
test> use data
switched to db data
data>
```

Создадим юзера manager и паролем password с правом чтения данной базы

```
data> db.createUser({  user: "manager",pwd: "password",roles:[{role: "read" , db:"data"}]})
{ ok: 1 }
data>
```

Задание 2 – ознакомиться с нижеуказанной статьей по теме «Bash»

<https://habr.com/ru/post/52871/>

Задание 3 – написать Bash-скрипт в соответствии с требованиями:

Содержание скрипта: замена существующего расширения в имени файла на заданное. Исходное имя файла и новое расширение передаются скрипту в качестве параметров. Основное средство: нестандартное раскрытие переменных. Усложнение: предусмотреть штатную реакцию на отсутствие расширения в исходном имени файла.

Листинг

```
GNU nano 6.2                               hw1.sh
#!/bin/bash

input=$1
output=$2

if [[ -z "$input" ]]; then echo 'исходный файл не указан'
elif [[ "$input" == "$output" ]]; then echo 'файл выхода не может быть равен исходному'
else
    if [[ $input != *.* ]]; then echo 'не указано расширение исходного файла но линуксу всеравно'
    fi
    echo "файл $input будет переименован в $output" #тут мы должны скопировать cp $input $output
fi
read -p 'Удалить исходный файл [y/n]' ans
if [[ $ans == 'y' || $ans == 'Y' ]]; then echo "файл $input удален" # удаляем файл rm $input -f
fi
exit 0
```

Вариант выполнения с удалением исходного и отсутствием расширения.сдуфк

```
anex13@anextmshwsrv:~$ ./hw1.sh test1 test1.txt
не указано расширение исходного файла но линуксу всеравно
файл test1 будет переименован в test1.txt
Удалить исходный файл [y/n]y
файл test1 удален
anex13@anextmshwsrv:~$
```

```
anex13@anextmshwsrv:~$ ./hw1.sh test1.log test1.txt
файл test1.log будет переименован в test1.txt
Удалить исходный файл [y/n]n
anex13@anextmshwsrv:~$
```

Задание 4 – написать Bash-скрипт в соответствии с требованиями:

Содержание скрипта: выделение из исходной строки подстроки с границами, заданными порядковыми номерами символов в исходной строке. Усложнение:

предусмотреть возможность не выделения, а удаления подстроки. Основные средства: команда cut, переменные оболочки.

Просто вариант извлечения строки

```
GNU nano 6.2 hw2.sh
#!/bin/bash

string=$1
start=$2
end=$3
echo ${string:$2-1:$3-1}
exit 0
```

Результат выполнения

```
anex13@anextmshwsrv:~$ ./hw2.sh sdgfhbsfgnj 2 5
dgfh
anex13@anextmshwsrv:~$
```

Вариант с вырезанием , cut и переменными оболочки

```
GNU nano 6.2 hw3.sh
#!/bin/bash

export STRING=$1
start=$2
end=$3
echo $STRING | cut -c -$start,$end-
unset $STRING
exit 0
```

Выполнение

```
anex13@anextmshwsrv:~$ ./hw3.sh sdgfhbsfgnj 2 5
sdhbsfgnj
anex13@anextmshwsrv:~$
```