

Homework_Lesson4_OS_3

Домашнее задание из личного кабинета. Цель: получить практический опыт установки пакетов с помощью сторонних репозиториев и пакетного менеджера. Научиться выполнять задачи автоматизации с помощью Bash.

Задание 1 – установить MongoDB. *создать таблицу data; создать пользователя manager, у которого будет доступ только на чтение этой таблицы.

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.

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

Выполнение 1 задания:

1.1 Установка MongoDB 7.

Сначала была попытка установки MongoDB 7 на VirtualBox, но завершилась она безуспешно.

Перед самой установкой MongoDB 7 на VirtualBox необходимо было установить зависимости:

```
$ apt update
```

```
$ apt install dirmngr gnupg apt-transport-https ca-certificates software-properties-common
```

1.2 Для установки актуального пакета MongoDB, необходимо добавить его в список репозиториев. Но предварительно следует импортировать открытый ключ для MongoDB. После добавить репозиторий MongoDB в директорию /etc/apt/sources.list.d/mongodb-org-7.0.list:

Импорт ключа:

```
$ curl -fsSL https://pgp.mongodb.com/server-7.0.asc | sudo gpg -o /usr/share/keyrings/mongodb-server-7.0.gpg --dearmor
```

Добавление репозитория в ~/apt/source.list.d/~:

```
$ 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" | tee  
/etc/apt/sources.list.d/mongodb-org-7.0.list
```

После обновляем локальный список пакетов, для добавления MongoDB 7.0:

```
$ apt update
```

```
$ apt install mongodb-org
```

После установки MongoDB 7.0 проверяем успешно ли установилась:

\$mongod--version

```
root@makarov-VirtualBox:~# mongod --version
Illegal instruction (core dumped)
root@makarov-VirtualBox:~# service mongod status
* mongod.service - MongoDB Database Server
   Loaded: loaded (/lib/systemd/system/mongod.service; disabled; vendor preset: enabled)
   Active: failed (Result: core-dump) since Thu 2024-10-24 23:48:21 +03; 10min ago
     Docs: https://docs.mongodb.org/manual
   Process: 8342 ExecStart=/usr/bin/mongod --config /etc/mongod.conf (code=dumped, signal=ILL)
   Main PID: 8342 (code=dumped, signal=ILL)
      CPU: 9ms

кac 24 23:48:21 makarov-VirtualBox systemd[1]: Started MongoDB Database Server.
кac 24 23:48:21 makarov-VirtualBox systemd[1]: mongod.service: Main process exited, code=dumped, status=4/ILL
кac 24 23:48:21 makarov-VirtualBox systemd[1]: mongod.service: Failed with result 'core-dump'.
root@makarov-VirtualBox:~#
```

Выскакивает ошибка Illegal instruction (core dumped). При проверки service также ошибка работоспособности MongoDB 7. В интернете у юзеров проблема решалась повторной переустановкой, возможно с ключами проблемы и т.д. и т.п. В моей ситуации, данные манипуляции не помогли. После долгих поисков проблемы, выяснилось, что на VirtualBox машинах успешно устанавливается MongoDB версии 4 не выше т.к. в сборках до 4 версии отсутствует поддержка AVX. Более поздние версии MongoDB, начиная с 5 и более новые, требуют поддержки в процессоре флага AVX, а VM VirtualBox почему-то скрывает её.

```
* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

New release '24.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Thu Oct 24 19:54:42 +03 2024 on tty3
makarov@makarov-VirtualBox:~$ sudo -i
[sudo] password for makarov:
root@makarov-VirtualBox:~# cat /proc/cpuinfo
processor       : 0
vendor_id      : GenuineIntel
cpu family     : 6
model          : 140
model name     : 11th Gen Intel(R) Core(TM) i5-11320H @ 3.20GHz
stepping       : 2
microcode      : 0xffffffff
cpu MHz        : 3187.198
cache size     : 8192 KB
physical id    : 0
siblings       : 1
core id        : 0
cpu cores      : 1
apicid         : 0
initial apicid : 0
fpu            : yes
fpu_exception  : yes
cpuid level    : 22
wp             : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ht syscall nx rdtscp lm constant_tsc rep_g
d nopl xtopology nonstop_tsc cpuid tsc_known_freq pni pclmulqdq ssse3 cx16 pcid sse4_1 sse4_2 movbe popcnt aes rdrand hypervisorlahf_lm abm 3dnowprefetch ibrs
enhanced fsgsbase bmi1 bmi2 invpcid rdseed clflushopt arat md_clear flush_l1d arch_capabilities
bugs           : spectre_v1 spectre_v2 spec_store_bypass swapgs retbleed eibrs_pbrsb bhi
bogomips       : 6374.39
clflush size   : 64
cache_alignm   : 64
address sizes  : 39 bits physical, 48 bits virtual
power managem  :

root@makarov-VirtualBox:~#
```

1.3 Пришлось развернуть все повторно на wsl Ubuntu 22.04.

Выполняем все те же действия, что и на VirtualBox, только проверив процессор на наличие флага AVX:

```

bugs          : spectre_v1 spectre_v2 spec_store_bypass swapgs retbleed eibrs_pbrsb gds
bogomips      : 6374.39
clflush size  : 64
cache_alignment : 64
address sizes  : 39 bits physical, 48 bits virtual
power management:

processor      : 7
vendor_id      : GenuineIntel
cpu family     : 6
model          : 140
model name     : 11th Gen Intel(R) Core(TM) i5-11320H @ 3.20GHz
stepping       : 2
microcode      : 0xffffffff
cpu MHz        : 3187.199
cache size     : 8192 KB
physical id    : 0
siblings       : 8
core id        : 3
cpu cores      : 4
apicid         : 7
initial apicid : 7
fpu            : yes
fpu_exception  : yes
cpuid level    : 21
wp             : yes
flags          : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ss h
t syscall nx pdpe1gb rdtscp lm constant_tsc arch_perfmon rep_good noopl xtopology tsc_reliable nonstop_tsc cpuid pni pclm
ulqddq sse3 fma cx16 pcdm pcid sse4_1 sse4_2 movbe popcnt aes xsave avx f16c rdrand hypervisor lahf_lm abm 3dnowprefetch
invpcid_single ssbd ibrs ibpb stibp ibrs_enhanced fsgsbase bmi1 avx2 smep bmi2 erms invpcid avx512f avx512dq rdseed adx
smmap avx512ifma cflushopt clwb avx512cd sha_ni avx512bw avx512vl xsaveopt xsavec xgetbv1 xsaves avx512vbmi umip avx512
_vbmi2 gfni vaes vpclmulqdq avx512_vnni avx512_bitalg avx512_vpopcntdq rdpid fsrm avx512_vp2intersect md_clear flush_l1d
arch_capabilities

bugs          : spectre_v1 spectre_v2 spec_store_bypass swapgs retbleed eibrs_pbrsb gds
bogomips      : 6374.39
clflush size  : 64
cache_alignment : 64
address sizes  : 39 bits physical, 48 bits virtual
power management:

root@DESKTOP-UG6J7T7:~#

```

Проблем с виртуализацией на WSL нету. Также выяснилось, что таких же проблем на VMware Workstation.

Приступаем к повторной установке, только уже на WSL Ubuntu 22.04:

```
$ apt update
```

```
$ apt install dirmngr gnupg apt-transport-https ca-certificates software-properties-common
```

```
$ curl -fsSL https://pgp.mongodb.com/server-7.0.asc | sudo gpg -o /usr/share/keyrings/mongodb-server-7.0.gpg --dearmor
```

```
$ 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" | tee
/etc/apt/sources.list.d/mongodb-org-7.0.list
```

```
$ apt update
```

```
$ apt install mongodb-org
```

Выполняем проверку, успешно ли установилась MongoDB 7.0 на WSL Ubuntu 22.04:

```
$ mongod --version
```

```

Adding group `mongodb' (GID 118) ...
Done.
Adding user `mongodb' to group `mongodb' ...
Adding user mongodb to group mongodb
Done.
Setting up mongodb-org-shell (7.0.14) ...
Setting up mongodb-database-tools (100.10.0) ...
Setting up mongodb-org-mongos (7.0.14) ...
Setting up mongodb-org-database-tools-extra (7.0.14) ...
Setting up mongodb-org-database (7.0.14) ...
Setting up mongodb-org-tools (7.0.14) ...
Setting up mongodb-org (7.0.14) ...
Processing triggers for man-db (2.10.2-1) ...
root@DESKTOP-UG6J7T7:~# mongod --version
db version v7.0.14
Build Info: {
  "version": "7.0.14",
  "gitVersion": "ce59cfc6a3c5e5c067dca0d30697edd68d4f5188",
  "opensslVersion": "OpenSSL 3.0.2 15 Mar 2022",
  "modules": [],
  "allocator": "tcmalloc",
  "environment": {
    "distmod": "ubuntu2204",
    "distarch": "x86_64",
    "target_arch": "x86_64"
  }
}
root@DESKTOP-UG6J7T7:~#

```

Также проверяем запущен ли сервис MongoDB:

\$ systemctl status mongod

```

root@DESKTOP-UG6J7T7:~# systemctl start mongod
root@DESKTOP-UG6J7T7:~# systemctl status mongod
● mongod.service - MongoDB Database Server
   Loaded: loaded (/lib/systemd/system/mongod.service; disabled; vendor preset: enabled)
   Active: active (running) since Fri 2024-10-25 00:24:39 +03; 9s ago
     Docs: https://docs.mongodb.org/manual
    Main PID: 1446 (mongod)
      Memory: 85.5M
    CGroup: /system.slice/mongod.service
            └─1446 /usr/bin/mongod --config /etc/mongod.conf

Oct 25 00:24:39 DESKTOP-UG6J7T7 systemd[1]: Started MongoDB Database Server.
Oct 25 00:24:39 DESKTOP-UG6J7T7 mongod[1446]: {"t":{"$date":"2024-10-24T21:24:39.868Z"},"s":"I", "c":"CONTROL", "id":>
lines 1-11/11 (END)

```

Установка MongoDB завершена.

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

Запускаем оболочку Mongosh:

\$ mongosh

Проверяем наличие баз в MongoDB и создаем свою БД вводя команды:

> show databases //показывает список баз

> use testdb //создание БД testdb

```

makarov@DESKTOP-UG6J7T7:~$ mongosh
Current Mongosh Log ID: 671abed720fda3fb2dfe6910
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.3.2
Using MongoDB:      7.0.14
Using Mongosh:       2.3.2

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

To help improve our products, anonymous usage data is collected and sent to MongoDB periodically (https://www.mongodb.com/legal/privacy-policy).
You can opt-out by running the disableTelemetry() command.

-----
  The server generated these startup warnings when booting
  2024-10-25T00:24:39.936+03:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
  2024-10-25T00:24:40.095+03:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
  2024-10-25T00:24:40.095+03:00: /sys/kernel/mm/transparent_hugepage/enabled is 'always'. We suggest setting it to 'never' in this binary version
  2024-10-25T00:24:40.095+03:00: vm.max_map_count is too low
-----

test> show databases;
admin    40.00 KiB
config   12.00 KiB
local    40.00 KiB
test> use testdb
switched to db testdb
testdb> show databases;
admin    40.00 KiB
config   12.00 KiB
local    40.00 KiB
testdb>

```

Создаем таблицу data в БД testdb:

```
> db.data.insertOne({ newField: "newValue" })           //создание таблицы data
```

```
> db.data.find ()                                     // смотрим на созданную таблицу data
```

```

testdb> db.data.find()
[ { _id: ObjectId('671b6a6ea13c830c26fe6911'), newField: 'newValue' } ]
testdb>

```

Создаем пользователя manager с правами только на чтение:

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

После проверяем его в наличии БД testdb:

```
> db.getUsers()
```

```
test> use testdb
switched to db testdb
testdb> db.createUser({
...   user: "manager",
...   pwd: "manager",
...   roles: [
...     { role: "read", db: "testdb" }
...   ]
... })
{ ok: 1 }
testdb> db.getUsers()
{
  users: [
    {
      _id: 'testdb.manager',
      userId: UUID('82d6be2f-86db-4493-86cb-9040802daaaa'),
      user: 'manager',
      db: 'testdb',
      roles: [ { role: 'read', db: 'testdb' } ],
      mechanisms: [ 'SCRAM-SHA-1', 'SCRAM-SHA-256' ]
    }
  ],
  ok: 1
}
testdb>
```

Правим конфигурационный файл /etc/mongod.conf, строку security убираем комментарий и добавляем authorization: enabled:

```
# how the process runs
processManagement:
  timeZoneInfo: /usr/share/zoneinfo

security:
  authorization: enabled

#operationProfiling:
```

После делаем рестарт службы mongod и пробуем залогиниться под новым созданным юзером manager:

```
makarov@DESKTOP-UG6J7T7:~$ mongosh -u manager -p manager testdb
Current Mongosh Log ID: 671b6cf6c66641781afe6910
Connecting to:      mongodb://<credentials>@127.0.0.1:27017/testdb?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.3.2
Using MongoDB:      7.0.14
Using Mongosh:      2.3.2

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

testdb> db.Car.insertOne({ newField:"newValue" })
MongoServerError[Unauthorized]: not authorized on testdb to execute command { insert: "Car", documents: [ { newField: "newValue", _id: ObjectId('671b6cf6c66641781afe6911') } ], ordered: true, lsid: { id: UUID("46d976cb-9201-4b91-97e7-1ea4f6128a1e") }, $db: "testdb" }
testdb>
```

Пробуем добавить документ в таблицу data из под пользователя manager:

```
testdb> db.data.insertOne({ new2Field:"new2Value" })
MongoServerError[Unauthorized]: not authorized on testdb to execute command { insert: "data", documents: [ { new2Field: "new2Value", _id: ObjectId('671b8320c66641781afe6914') } ], ordered: true, lsid: { id: UUID("46d976cb-9201-4b91-97e7-1ea4f6128a1e") }, $db: "testdb" }
testdb>
```

Пользователь manager не может добавить документ, т.к. у него роль создана только на чтение.

Выполнение 3 задания:

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.

```
$ replac_ext.sh X Homework_Lesson4_OS_3.pdf name.exe
Homework_Lesson4_OS_3 > script_bash > $ replac_ext.sh
1  #!/bin/bash
2
3  # Получение параметров
4
5  read -p "Введите файл, который нужно изменить:" file_name
6
7  echo $file_name
8
9  read -p "Введите нужное расширение:" new_ext
10
11 echo $new_ext
12
13 # Извлечение имени файла без расширения и новое имя
14 new_file_name="${file_name%.*}.${new_ext}"
15
16 mv "$file_name" "$new_file_name"
17 echo "Файл переименован в: $new_file_name"
18 |
```

Вывод выполнения скрипта:

```
root@DESKTOP-UG6J7T7:/u01# ./replac_ext.sh
Введите файл, который нужно изменить:test.exe
test.exe
Введите нужное расширение:txt
txt
Файл переименован в: test.txt
root@DESKTOP-UG6J7T7:/u01# ls -l
total 8
-rwxrwxrwx 1 root root 1051 Oct 26 01:52 1_bash.sh
-rwxr-xr-x 1 root root 598 Oct 28 22:23 replac_ext.sh
-rwxrwxrwx 1 root root 0 Oct 28 21:26 test.txt
root@DESKTOP-UG6J7T7:/u01#
```

Выполнение 4 задания:

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.