

Отчёт по лабораторной работе №1

Информационная безопасность

Настройка рабочего пространства и конфигурация операционной системы на виртуальную машину. Система контроля версий Git. Язык разметки Markdown

Выполнила: Сингх Ааруши ,
НКАбд-02-23, 132215095

Содержание

Цель работы	1
Теоретическое введение	2
Выполнение лабораторной работы	3
Установка и конфигурация операционной системы на виртуальную машину.....	3
Virtual Box.....	3
Домашнее задание.....	6
Управление версиями	8
Настройка github	8
Установка программного обеспечения	8
Базовая настройка git.....	9
Создание ключа SSH	9
Создание ключа PGP	10
Добавление PGP ключа в GitHub	10
Вывод.....	11
Список литературы. Библиография	11

Цель работы

Целью данной работы является приобретение практических навыков установки операционной системы на виртуальную машину, настройки минимально необходимых для дальнейшей работы сервисов.

Теоретическое введение

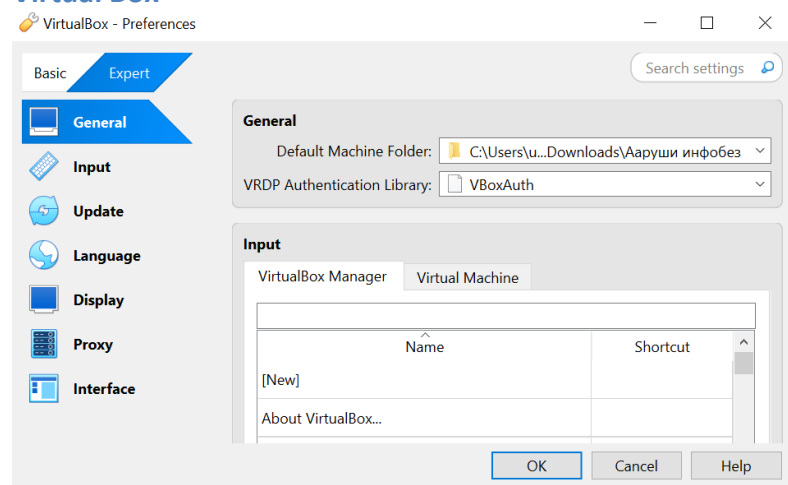
Oracle VM VirtualBox — это мощная и бесплатная виртуализационная платформа, разработанная корпорацией Oracle, которая позволяет пользователям создавать и управлять виртуальными машинами на своих компьютерах. [1] Системы контроля версий (Version Control System, VCS) применяются при работе нескольких человек над одним проектом. Обычно основное дерево проекта хранится в локальном или удалённом репозитории, к которому настроен доступ для участников проекта. При внесении изменений в содержание проекта система контроля версий позволяет их фиксировать, совмещать изменения, произведённые разными участниками проекта, производить откат к любой более ранней версии проекта, если это требуется. [2] В классических системах контроля версий используется централизованная модель, предполагающая наличие единого репозитория для хранения файлов. Выполнение большинства функций по управлению версиями осуществляется специальным сервером. Участник проекта (пользователь) перед началом работы посредством определённых команд получает нужную ему версию файлов. После внесения изменений пользователь размещает новую версию в хранилище. При этом предыдущие версии не удаляются из центрального хранилища и к ним можно вернуться в любой момент. Сервер может сохранять не полную версию изменённых файлов, а производить так называемую дельта-компрессию — сохранять только изменения между последовательными версиями, что позволяет уменьшить объём хранимых данных. Системы контроля версий поддерживают возможность отслеживания и разрешения конфликтов, которые могут возникнуть при работе нескольких человек над одним файлом. Можно объединить (слить) изменения, сделанные разными участниками (автоматически или вручную), вручную выбрать нужную версию, отменить изменения вовсе или заблокировать файлы для изменения. В зависимости от настроек блокировка не позволяет другим пользователям получить рабочую копию или препятствует изменению рабочей копии файла средствами файловой системы ОС, обеспечивая таким образом, привилегированный доступ только одному пользователю, работающему с файлом. Системы контроля версий также могут обеспечивать дополнительные, более гибкие функциональные возможности. Например, они могут поддерживать работу с несколькими версиями одного файла, сохраняя общую историю изменений до точки ветвления версий и собственные истории изменений каждой ветви. Кроме того, обычно доступна информация о том, кто из участников, когда и какие изменения вносил. Обычно такого рода информация хранится в журнале изменений, доступ к которому можно ограничить. В отличие от классических, в распределённых системах контроля версий центральный репозиторий не является обязательным. Среди классических VCS наиболее известны CVS, Subversion, а среди распределённых — Git, Bazaar, Mercurial. Принципы их работы схожи, отличаются они в основном синтаксисом используемых в работе команд. Примеры команд для Git: Название команды Описание команды `git clone` Клонирование репозитория на ПК `git commit -m "Initial Commit"` Оставление коммита `git push` Загрузка изменений на гит `make` Конвертация файла .md Markdown - это легковесный язык разметки, который широко используется для создания форматированного текста в веб-среде. Его простота и читаемость делают его популярным среди разработчиков, писателей и

блогеров. Синтаксис Markdown состоит из простых символов и правил форматирования, которые позволяют создавать заголовки, списки, ссылки, изображения и другие элементы веб-страниц без необходимости использовать сложные HTML-теги. Он также легко читается в исходном виде и может быть конвертирован в различные форматы, такие как HTML, PDF или документы Microsoft Word, делая Markdown удобным инструментом для создания содержательного и красочного контента в интернете. [3]

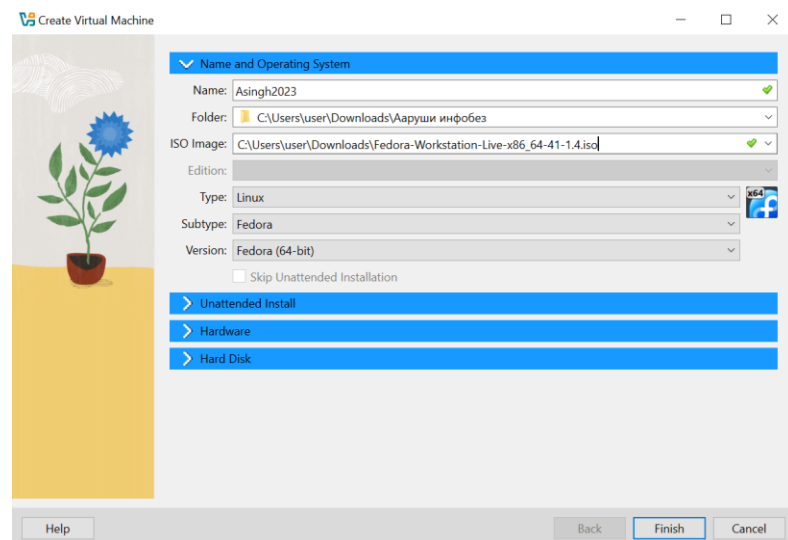
Выполнение лабораторной работы

Установка и конфигурация операционной системы на виртуальную машину

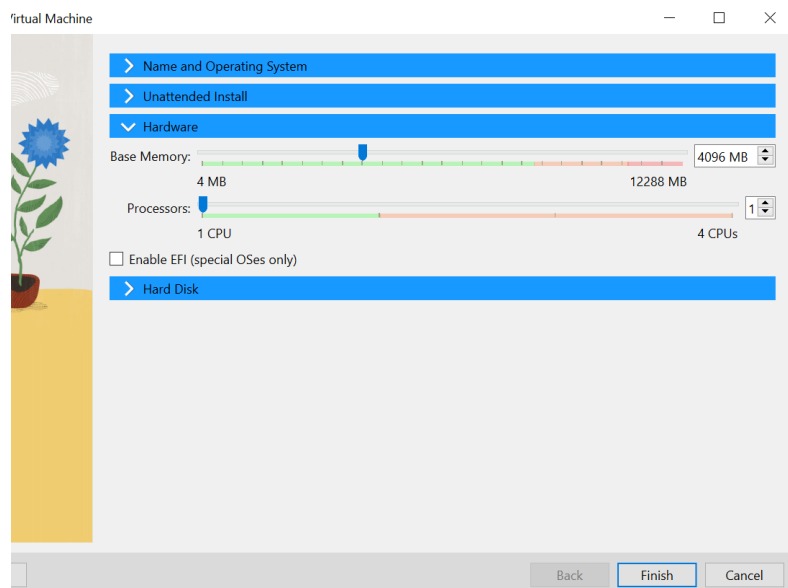
Virtual Box



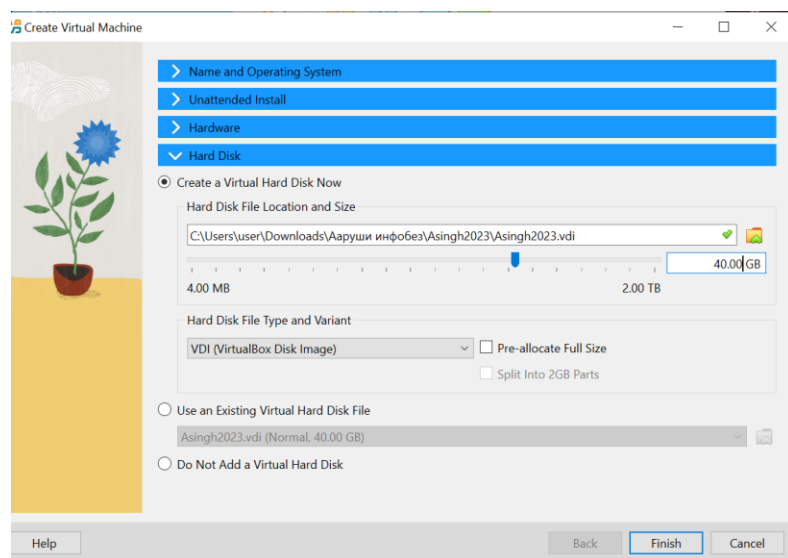
(рис. 1. Общие настройки)



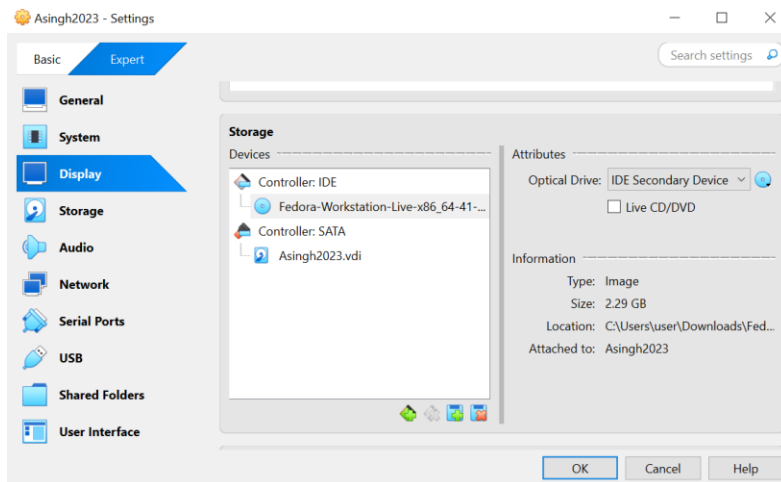
(рис. 2. Имя и путь ОС)



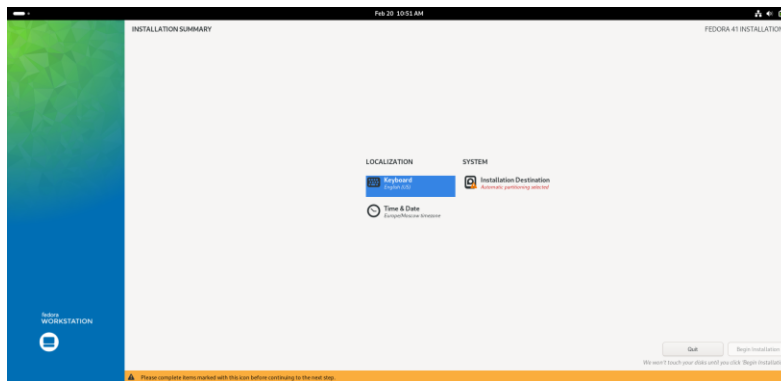
(рис. 3. Размер памяти и число процессоров)



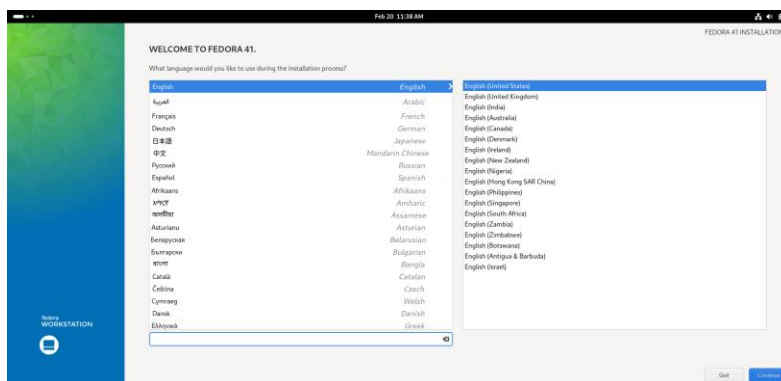
(рис. 4. Виртуальный жесткий диск)



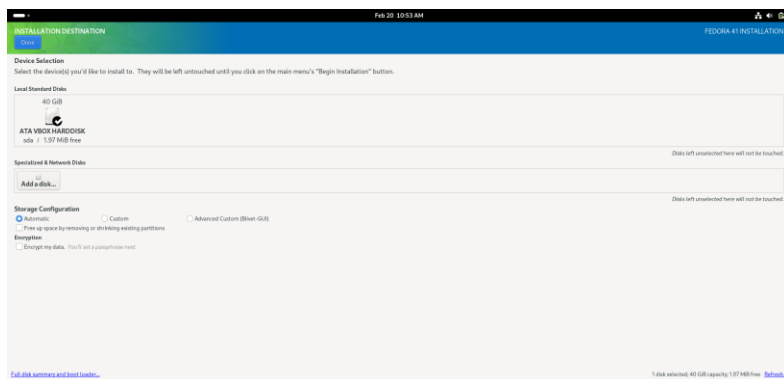
(рис. 5. Носители)



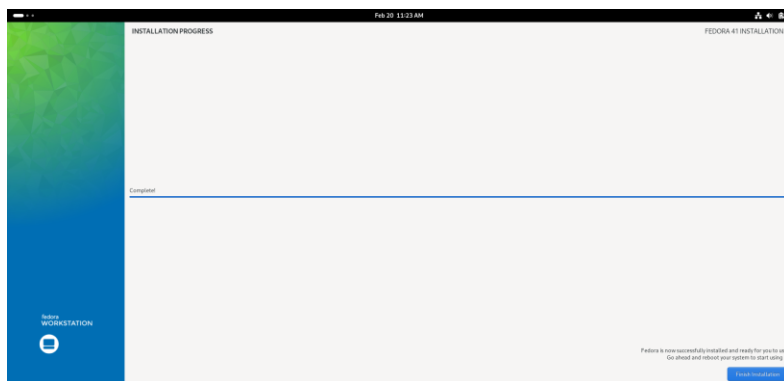
(рис. 6. Стартовое меню установки)



(рис. 7. Установка английского языка интерфейса ОС)



(рис. 8. Окно настройки установки: место установки)



(рис. 9. Успешное завершение установки и перезагрузка системы)

Домашнее задание

```
liveuser@localhost-live:~
liveuser@Asingh2003:~$ dmesg
dmesg: read kernel buffer failed: Operation not permitted
liveuser@Asingh2003:~$ % sudo sysctl kernel.dmesg_restrict=0
bash: fg: %: no such job
liveuser@Asingh2003:~$ % sudo sysctl kernel.dmesg_restrict=0
bash: fg: %: no such job
liveuser@Asingh2003:~$ sudo sysctl kernel.dmesg_restrict=0
kernel.dmesg_restrict = 0
liveuser@Asingh2003:~$ dmesg
[ 0.000000] Linux version 6.11.4-301.fc41.x86_64 (mockbuild@9b6b61418589428cb
880a7020233b56f) (gcc (GCC) 14.2.1 20240912 (Red Hat 14.2.1-3), GNU ld version 2
.43.1-2.fc41) #1 SMP PREEMPT_DYNAMIC Sun Oct 20 15:02:33 UTC 2024
[ 0.000000] Command line: BOOT_IMAGE=/images/pxeboot/vmlinuz root=live:CDLABE
L=Fedora-WS-Live-41-1-4 rd.live.image rd.live.check quiet
[ 0.000000] BIOS-provided physical RAM map:
[ 0.000000] BIOS-e820: [mem 0x0000000000000000-0x00000000000009fbff] usable
[ 0.000000] BIOS-e820: [mem 0x00000000000009fc00-0x00000000000009ffff] reserved
[ 0.000000] BIOS-e820: [mem 0x0000000000000f0000-0x0000000000000fffff] reserved
[ 0.000000] BIOS-e820: [mem 0x0000000001000000-0x000000000dffffffffff] usable
[ 0.000000] BIOS-e820: [mem 0x000000000dffff0000-0x000000000dffffffffff] ACPI data
[ 0.000000] BIOS-e820: [mem 0x00000000fec00000-0x00000000fec00fff] reserved
[ 0.000000] BIOS-e820: [mem 0x00000000fee00000-0x00000000fee00fff] reserved
[ 0.000000] BIOS-e820: [mem 0x00000000fffc0000-0x00000000ffffff] reserved
[ 0.000000] BIOS-e820: [mem 0x0000000100000000-0x000000011ffffffffff] usable
```

(рис. 10. dmesg)

```
liveuser@localhost-live:~  
[ 3556.236516] EXT4-fs (sda2): unmounting filesystem 5c4a5d5d-3440-49c8-a09c-f8d57e3442da.  
liveuser@Asingh2003:~$ dmesg | less  
liveuser@Asingh2003:~$ dmesg | grep -i "memory"  
[ 0.000000] DMI: Memory slots populated: 0/0  
[ 0.038647] ACPI: Reserving FACP table memory at [mem 0xdfff00f0-0xdfff01e3]  
[ 0.038651] ACPI: Reserving DSDT table memory at [mem 0xdfff0610-0xdfff2962]  
[ 0.038653] ACPI: Reserving FACS table memory at [mem 0xdfff0200-0xdfff023f]  
[ 0.038655] ACPI: Reserving FACS table memory at [mem 0xdfff0200-0xdfff023f]  
[ 0.038657] ACPI: Reserving APIC table memory at [mem 0xdfff0240-0xdfff0293]  
[ 0.038659] ACPI: Reserving SSDT table memory at [mem 0xdfff02a0-0xdfff060b]  
[ 0.039861] Early memory node ranges  
[ 0.229880] PM: hibernation: Registered nosave memory: [mem 0x00000000-0x0000ffff]  
[ 0.229885] PM: hibernation: Registered nosave memory: [mem 0x0009f000-0x0009ffff]  
[ 0.229887] PM: hibernation: Registered nosave memory: [mem 0x000a0000-0x000effff]  
[ 0.229889] PM: hibernation: Registered nosave memory: [mem 0x000f0000-0x000fffff]  
[ 0.229892] PM: hibernation: Registered nosave memory: [mem 0xdfff0000-0xdfffffff]  
[ 0.229894] PM: hibernation: Registered nosave memory: [mem 0xe0000000-0xfefbffff]  
[ 0.229896] PM: hibernation: Registered nosave memory: [mem 0xfec00000-0xfec0ffff]
```

(рис. 11. dmesg | less, Объем доступной оперативной памяти)

```
liveuser@Asingh2003:~$ dmesg | grep -i "linux version"  
[ 0.000000] Linux version 6.11.4-301.fc41.x86_64 (mockbuild@9b6b61418589428cb880a7020233b56f) (gcc (GCC) 14.2.1 20240912 (Red Hat 14.2.1-3), GNU ld version 2.43.1-2.fc41) #1 SMP PREEMPT_DYNAMIC Sun Oct 20 15:02:33 UTC 2024  
liveuser@Asingh2003:~$
```

(рис. 12. версия ядра линукс, частота процессора, модель процессора)

```
liveuser@Asingh2003:~$ df -T  
Filesystem Type 1K-blocks Used Available Use% Mounted on  
LiveOS_rootfs overlay 800416 168680 631736 22% /  
devtmpfs devtmpfs 4096 0 4096 0% /dev  
tmpfs tmpfs 2001036 104 2000932 1% /dev/shm  
tmpfs tmpfs 800416 168680 631736 22% /run  
/dev/sr0 iso9660 2400574 2400574 0 100% /run/initramfs/live  
/dev/loop0 squashfs 2206720 2206720 0 100% /run/rootfsbase  
tmpfs tmpfs 1024 0 1024 0% /run/credentials/systemd-journald.service  
tmpfs tmpfs 1024 0 1024 0% /run/credentials/systemd-network-generator.service  
tmpfs tmpfs 1024 0 1024 0% /run/credentials/systemd-udev-load-credentials.service  
tmpfs tmpfs 1024 0 1024 0% /run/credentials/systemd-tmpfiles-setup-dev-early.service  
tmpfs tmpfs 1024 0 1024 0% /run/credentials/systemd-sysctl.service  
tmpfs tmpfs 1024 0 1024 0% /run/credentials/systemd-tmpfiles-setup-dev.service  
vartmp tmpfs 2001036 0 2001036 0% /var/tmp  
tmpfs tmpfs 2001036 3540 1997496 1% /tmp  
tmpfs tmpfs 1024 0 1024 0% /run/credentials/systemd-tmpfiles-setup.service  
tmpfs tmpfs 1024 0 1024 0% /run/credentials/systemd
```

(рис. 13. Тип файловой системы корневого раздела)

```
liveuser@localhost-live:~  
liveuser@Asingh2003:~$ dmesg | grep -i "Mounted"  
[ 75.089967] systemd[1]: Mounted dev-hugepages.mount - Huge Pages File System.  
[ 75.111728] systemd[1]: Mounted dev-mqueue.mount - POSIX Message Queue File System.  
[ 75.112065] systemd[1]: Mounted sys-kernel-debug.mount - Kernel Debug File System.  
[ 75.112337] systemd[1]: Mounted sys-kernel-tracing.mount - Kernel Trace File System.  
[ 595.247248] EXT4-fs (sda2): mounted filesystem 5c4a5d5d-3440-49c8-a09c-f8d57e3442da r/w with ordered data mode. Quota mode: none.  
[ 3114.187198] EXT4-fs (sda2): mounted filesystem 5c4a5d5d-3440-49c8-a09c-f8d57e3442da ro with ordered data mode. Quota mode: none.  
[ 3236.212978] EXT4-fs (sda2): mounted filesystem 5c4a5d5d-3440-49c8-a09c-f8d57e3442da ro with ordered data mode. Quota mode: none.  
[ 3491.992099] EXT4-fs (sda2): mounted filesystem 5c4a5d5d-3440-49c8-a09c-f8d57e3442da ro with ordered data mode. Quota mode: none.  
[ 3518.188092] EXT4-fs (sda2): mounted filesystem 5c4a5d5d-3440-49c8-a09c-f8d57e3442da ro with ordered data mode. Quota mode: none.  
[ 3556.186284] EXT4-fs (sda2): mounted filesystem 5c4a5d5d-3440-49c8-a09c-f8d57e3442da ro with ordered data mode. Quota mode: none.  
liveuser@Asingh2003:~$
```

(рис. 14. Последовательность монтирования файловых систем)

Управление версиями

Настройка github

Уже имеется аккаунт: <https://github.com/Aarushi102003>

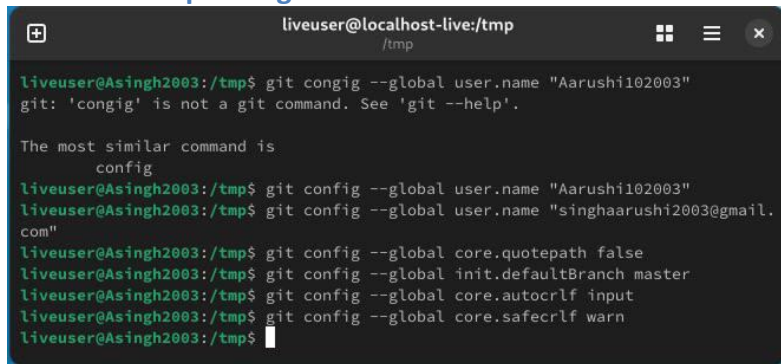
Установка программного обеспечения

Установка git-flow

```
liveuser@localhost-live:/tmp  
liveuser@Asingh2003:~$ cd /tmp  
liveuser@Asingh2003:/tmp$ wget --no-check-certificate -q https://raw.githubusercontent.com/petervanderdoes/gitflow/develop/contrib/gitflow-installer.sh  
liveuser@Asingh2003:/tmp$ chmod +x gitflow-installer.sh  
liveuser@Asingh2003:/tmp$ sudo ./gitflow-installer.sh install stable  
### git-flow no-make installer ###  
Installing git-flow to /usr/local/bin  
Cloning repo from GitHub to gitflow  
Cloning into 'gitflow'..  
remote: Enumerating objects: 4270, done.  
remote: Total 4270 (delta 0), reused 0 (delta 0), pack-reused 4270 (from 1)  
Receiving objects: 100% (4270/4270), 1.74 MiB | 2.98 MiB/s, done.  
Resolving deltas: 100% (2533/2533), done.  
Already up to date.  
branch 'master' set up to track 'origin/master'.  
Switched to a new branch 'master'  
install: creating directory '/usr/local/share/doc'  
install: creating directory '/usr/local/share/doc/gitflow'  
install: creating directory '/usr/local/share/doc/gitflow/hooks'  
'gitflow/git-flow' -> '/usr/local/bin/git-flow'  
'gitflow/git-flow-init' -> '/usr/local/bin/git-flow-init'
```

(рис. 15. Установка git-flow)

Базовая настройка git



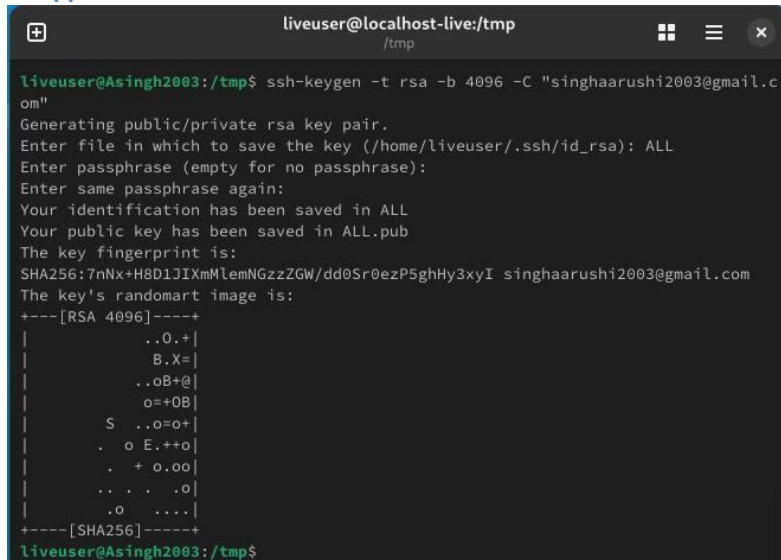
```
liveuser@localhost-live:/tmp
/tmp

liveuser@Asingh2003:/tmp$ git config --global user.name "Aarushi102003"
git: 'config' is not a git command. See 'git --help'.

The most similar command is
  config
liveuser@Asingh2003:/tmp$ git config --global user.name "singhaarushi2003@gmail.com"
liveuser@Asingh2003:/tmp$ git config --global core.quotepath false
liveuser@Asingh2003:/tmp$ git config --global init.defaultBranch master
liveuser@Asingh2003:/tmp$ git config --global core.autocrlf input
liveuser@Asingh2003:/tmp$ git config --global core.safecrlf warn
liveuser@Asingh2003:/tmp$
```

(рис. 16. Базовая настройка git)

Создание ключа SSH



```
liveuser@localhost-live:/tmp
/tmp

liveuser@Asingh2003:/tmp$ ssh-keygen -t rsa -b 4096 -C "singhaarushi2003@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/liveuser/.ssh/id_rsa): ALL
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in ALL
Your public key has been saved in ALL.pub
The key fingerprint is:
SHA256:7nNx+H8D1JIXmMlemNGzzZGW/dd0Sr0ezP5ghHy3xyI singhaarushi2003@gmail.com
The key's randomart image is:
+---[RSA 4096]-----+
|      .O.+      |
|      B.X=      |
|      ..oB+@    |
|      o+=oB     |
|      S ..o=o+  |
|      . o E.+o+  |
|      . + o.o    |
|      .. . . .o  |
|      .o  ....  |
+---[SHA256]-----+
liveuser@Asingh2003:/tmp$
```

(рис. 17. Создание ключа SSH)

Создание ключа PGP

```
liveuser@localhost-live:/tmp
/tmp
liveuser@Asingh2003:/tmp$ gpg --full-generate-key
gpg (GnuPG) 2.4.5; Copyright (C) 2024 g10 Code GmbH
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

gpg: directory '/home/liveuser/.gnupg' created
Please select what kind of key you want:
  (1) RSA and RSA
  (2) DSA and Elgamal
  (3) DSA (sign only)
  (4) RSA (sign only)
  (9) ECC (sign and encrypt) *default*
  (10) ECC (sign only)
  (14) Existing key from card
Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (3072) 4096
Requested keysize is 4096 bits
Please specify how long the key should be valid.
  0 = key does not expire
  <n> = key expires in n days
  <n>w = key expires in n weeks
  <n>m = key expires in n months
  <n>y = key expires in n years
Key is valid for? (0) 0
Key does not expire at all
```

(рис. 18. Создание ключа PGP(1))

Добавление PGP ключа в GitHub

```
liveuser@localhost-live:/tmp
/tmp
GnuPG needs to construct a user ID to identify your key.

Real name: Aarushi Singh
Email address: singhaarushi2003@gmail.com
Comment:
You selected this USER-ID:
  "Aarushi Singh <singhaarushi2003@gmail.com>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? o
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: /home/liveuser/.gnupg/trustdb.gpg: trustdb created
gpg: directory '/home/liveuser/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as '/home/liveuser/.gnupg/openpgp-revocs.d/7C
890892E76DA43899510134C827D8CC9579A3CC.rev'
public and secret key created and signed.

pub   rsa4096 2025-02-20 [SC]
      7C890892E76DA43899510134C827D8CC9579A3CC
uid           Aarushi Singh <singhaarushi2003@gmail.com>
sub   rsa4096 2025-02-20 [E]
```

(рис. 19. Создание ключа PGP(2))

Далее дополнительно настроили репозиторий курса согласно заданию, написали отчет на Markdown и конвертировали .md в .pdf и. docs при помощи команды make и файла Makefile.

Отправили все изменения и коммиты на GitHub.

Вывод

Были настроено рабочее пространство для лабораторных работ, приобретены практические навыки установки операционной системы на виртуальную машину и настройки минимально необходимых для дальнейшей работы сервисов. Были изучены идеология и применение средств контроля версий, освоены умения по работе с git. Были приобретены практические навыки оформления отчётов с помощью легковесного языка разметки Markdown.

Список литературы. Библиография

[1] Документация по Virtual Box: <https://www.virtualbox.org/wiki/Documentation>

[2] Документация по Git: <https://git-scm.com/book/ru/v2>

[3] Документация по Markdown: <https://learn.microsoft.com/ru-ru/contribute/markdown-reference>