

Отчёт по лабораторной работе №2

Специальность: архитектура компьютеров

Ерфан Хосейнабади

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретические сведения	7
4	Задание для самостоятельной работы	9
4.1	1. Создание базовой конфигурации для работы с Git	9
4.2	2. Создание ключа SSH	10
4.3	3. Создание ключа PGP	12
4.4	4. Настройка подписей Git	14
4.5	5. Регистрация на GitHub	16
4.6	6. Создание локального каталога для выполнения заданий	16
4.7	Использование репозитория	16
	4.7.1 Удаление лишних файлов	16
	4.7.2 Создание необходимых каталогов	16
4.8	Выводы	17
5	Ответы на контрольные вопросы.	18

Список иллюстраций

Список таблиц

1 Цель работы

- Изучить идеологию и применение средств контроля версий.
- Освоить умения по работе с git.

2 Задание

1. Создать базовую конфигурацию для работы с git.
2. Создать ключ SSH.
3. Создать ключ PGP.
4. Настроить подписи git.
5. Зарегистрироваться на Github.
6. Создать локальный каталог для выполнения заданий по предмету.

3 Теоретические сведения

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. 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 зави-

симости 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

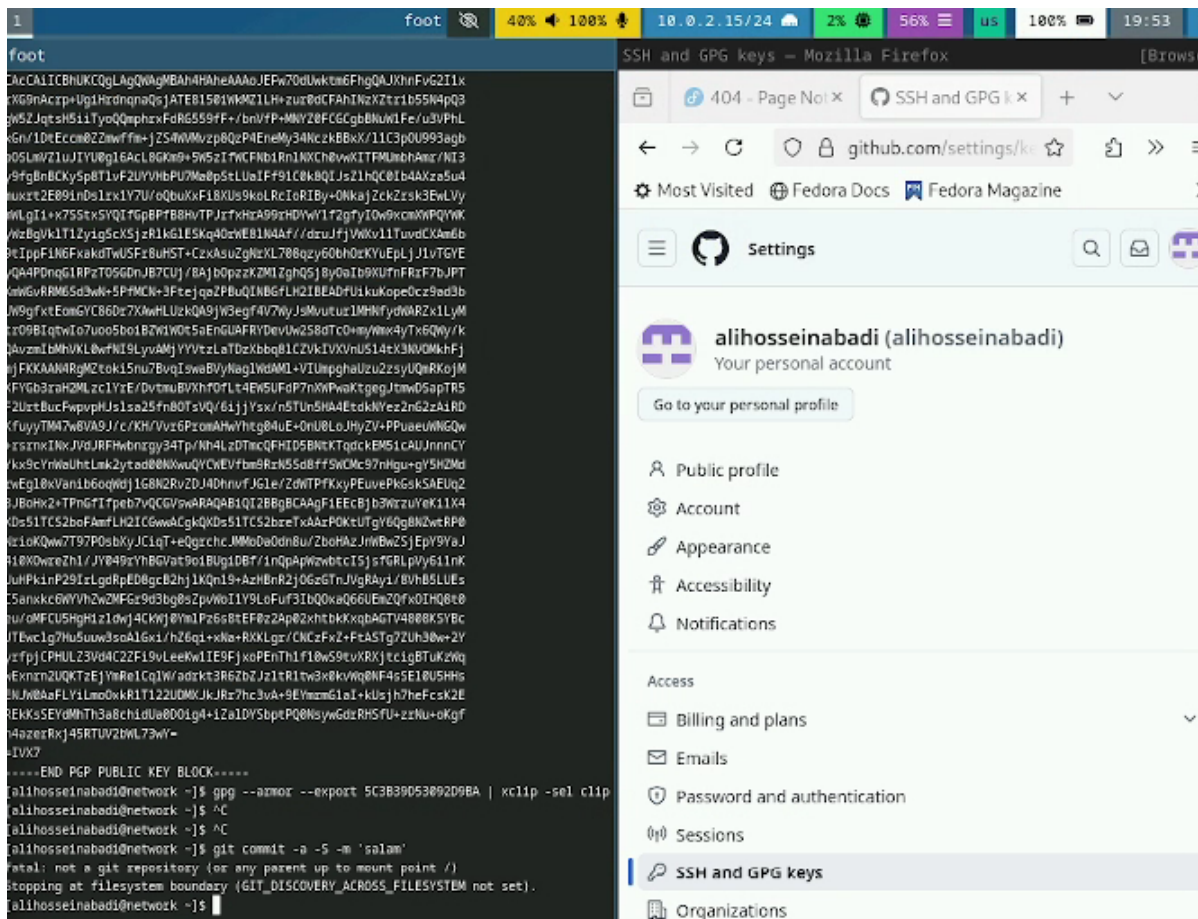
4 Задание для самостоятельной работы

4.1 1. Создание базовой конфигурации для работы с Git

Для начала необходимо выполнить базовую настройку **Git**, задав имя и email владельца репозитория. Используйте следующие команды:

```
git config --global user.name "Ваше Имя"
```

```
git config --global user.email "ваш_email@example.com"
```



{#fig:001

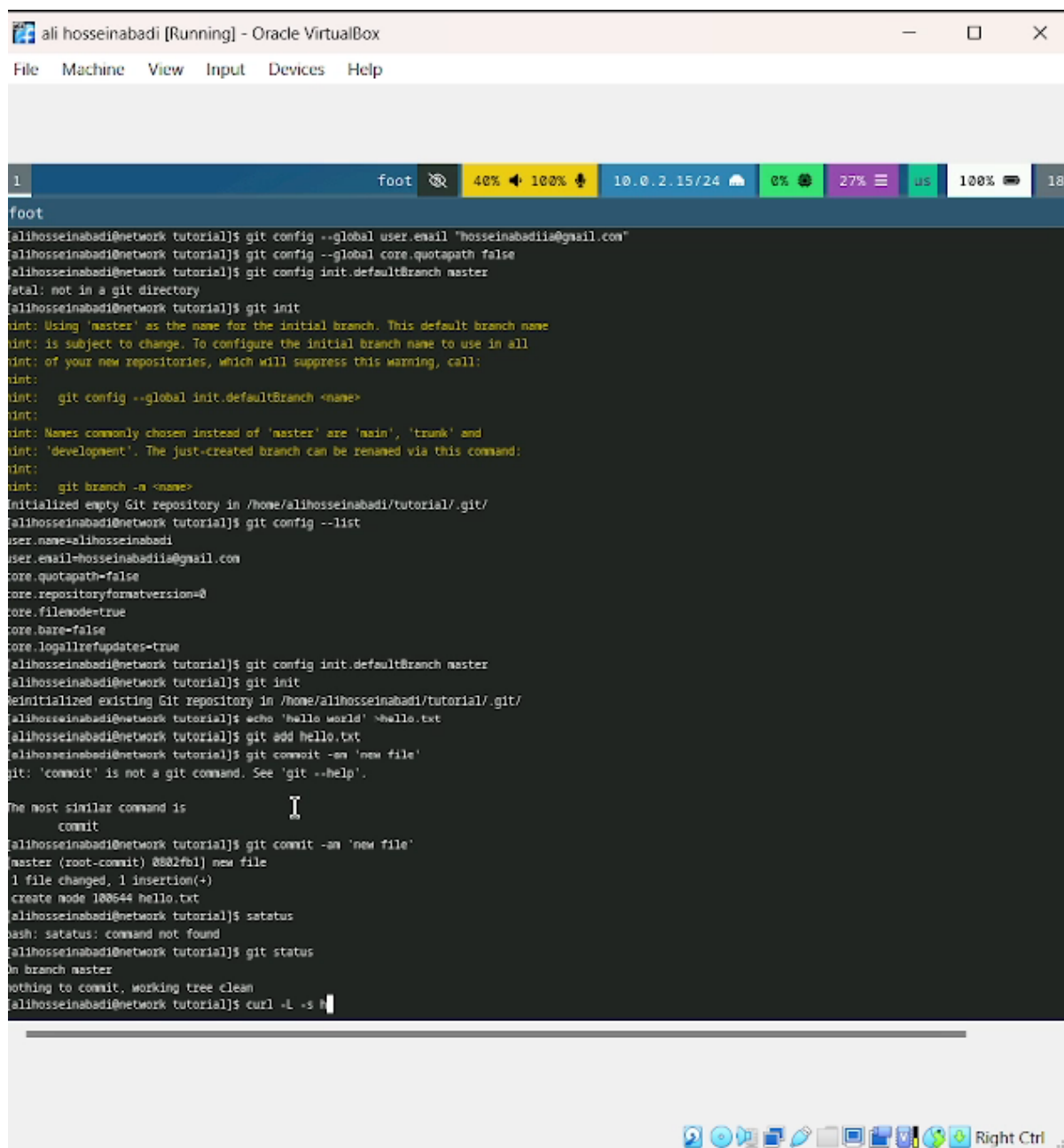
width:70%}

4.2 2. Создание ключа SSH

Следующим шагом является создание ключа **SSH** по алгоритму RSA с размером ключа **4096** бит. Введите команду:

```
ssh-keygen -t rsa -b 4096 -C "ваш_email@example.com"
```

При выполнении этой команды будет предложено указать место для сохранения ключа. Нажмите Enter, чтобы использовать значение по умолчанию.



```
alihosseinalbadi@network tutorial]$ git config --global user.email 'hosseinabadi@gmail.com'
alihosseinalbadi@network tutorial]$ git config --global core.quotapath false
alihosseinalbadi@network tutorial]$ git config init.defaultBranch master
fatal: not in a git directory
alihosseinalbadi@network tutorial]$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -n <name>
Initialized empty Git repository in /home/alihosseinalbadi/tutorial/.git/
alihosseinalbadi@network tutorial]$ git config --list
user.name=alihosseinalbadi
user.email=hosseinabadi@gmail.com
core.quotapath=false
core.repositoryformatversion=0
core.filemode=true
core.bare=false
core.logallrefupdates=true
alihosseinalbadi@network tutorial]$ git config init.defaultBranch master
alihosseinalbadi@network tutorial]$ git init
Reinitialized existing Git repository in /home/alihosseinalbadi/tutorial/.git/
alihosseinalbadi@network tutorial]$ echo 'hello world' >hello.txt
alihosseinalbadi@network tutorial]$ git add hello.txt
alihosseinalbadi@network tutorial]$ git commit -m 'new file'
git: 'commit' is not a git command. See 'git --help'.

The most similar command is
    commit
alihosseinalbadi@network tutorial]$ git commit -m 'new file'
[master (root-commit) 882fb1] new file
1 file changed, 1 insertion(+)
create mode 100644 hello.txt
alihosseinalbadi@network tutorial]$ git status
bash: git status: command not found
alihosseinalbadi@network tutorial]$ git status
On branch master
nothing to commit, working tree clean
alihosseinalbadi@network tutorial]$ curl -L -s H
```

{#fig:002

width:70%}

После этого вы можете скопировать сгенерированный SSH ключ из файла и вставить его в **GitHub** в разделе настроек SSH.

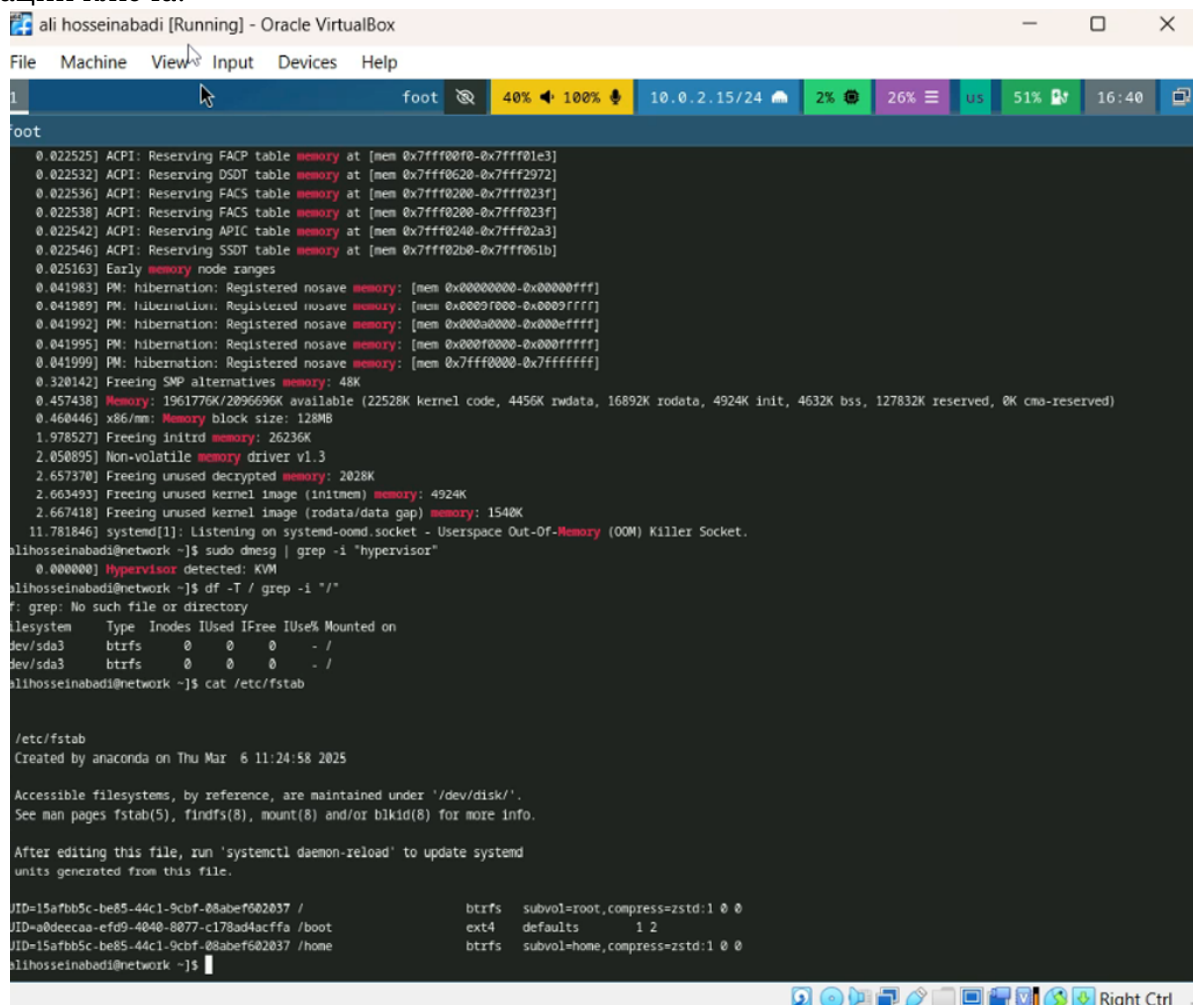
image{#fig:003 width:70%}

4.3 3. Создание ключа PGP

Для создания ключа PGP выполните команду:

```
gpg --full-generate-key
```

Выберите параметры, указанные на изображениях, и завершите процесс генерации ключа.



```
ali hosseinabadi [Running] - Oracle VirtualBox
File Machine View Input Devices Help
1
root
0.022525] ACPI: Reserving FACP table memory at [mem 0x7fff00f0-0x7fff01e3]
0.022532] ACPI: Reserving DSDT table memory at [mem 0x7fff0620-0x7fff2972]
0.022536] ACPI: Reserving FACS table memory at [mem 0x7fff0200-0x7fff023f]
0.022538] ACPI: Reserving FACS table memory at [mem 0x7fff0200-0x7fff023f]
0.022542] ACPI: Reserving APIC table memory at [mem 0x7fff0240-0x7fff02a3]
0.022546] ACPI: Reserving SSDT table memory at [mem 0x7fff02b0-0x7fff061b]
0.025163] Early memory node ranges
0.041983] PM: hibernation: Registered nosave memory: [mem 0x00000000-0x00000fff]
0.041989] PM: hibernation: Registered nosave memory: [mem 0x0000f000-0x0000ffff]
0.041992] PM: hibernation: Registered nosave memory: [mem 0x000a0000-0x000effff]
0.041995] PM: hibernation: Registered nosave memory: [mem 0x000f0000-0x000fffff]
0.041999] PM: hibernation: Registered nosave memory: [mem 0x7fff0000-0x7fffffff]
0.320142] Freeing SMP alternatives memory: 48K
0.457438] Memory: 1961776K/2096696K available (22528K kernel code, 4456K rdata, 16892K rodata, 4924K init, 4632K bss, 127832K reserved, 0K cma-reserved)
0.460446] x86/mm: Memory block size: 128MB
1.978527] Freeing initrd memory: 26236K
2.050895] Non-volatile memory driver v1.3
2.657370] Freeing unused decrypted memory: 2028K
2.663493] Freeing unused kernel image (initmem) memory: 4924K
2.667418] Freeing unused kernel image (rodata/data gap) memory: 1540K
11.781846] systemd[1]: Listening on systemd-oomd.socket - Userspace Out-Of-Memory (OOM) Killer Socket.
alihosseinabadi@network ~]$ sudo dmesg | grep -i "hypervisor"
0.000000] Hypervisor detected: KVM
alihosseinabadi@network ~]$ df -T / grep -i "/"
Filesystem Type Inodes IUsed IFree IUse% Mounted on
dev/sda3 btrfs 0 0 0 - /
dev/sda3 btrfs 0 0 0 - /
alihosseinabadi@network ~]$ cat /etc/fstab

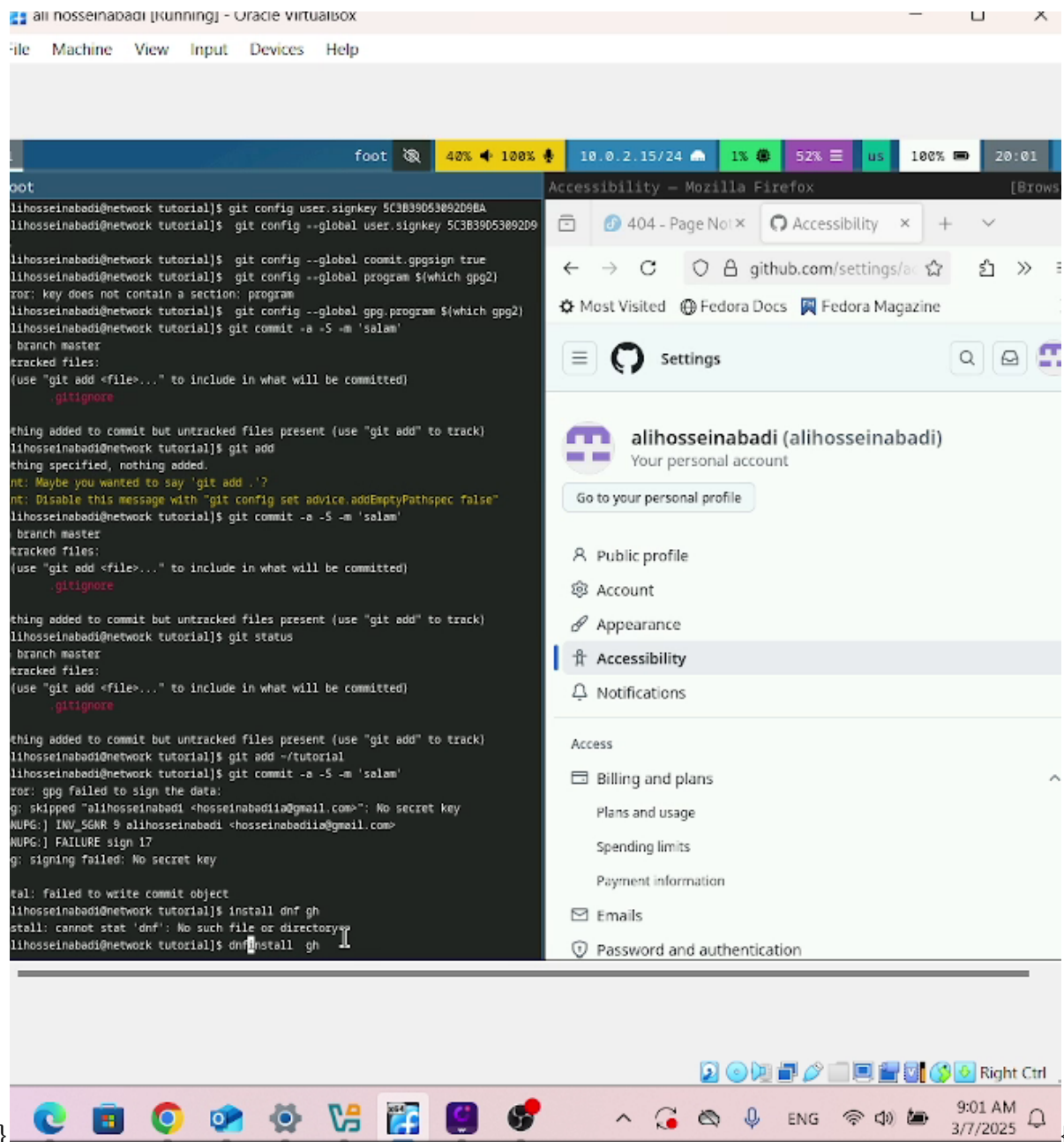
/etc/fstab
Created by anaconda on Thu Mar 6 11:24:58 2025

Accessible filesystems, by reference, are maintained under '/dev/disk/'.
See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.

After editing this file, run 'systemctl daemon-reload' to update systemd
units generated from this file.

UUID=15afb55c-be85-44c1-9cbf-08abef602037 / btrfs subvol=root,compress=zstd:1 0 0
UUID=a0deecaa-cfd9-4040-8077-c178ad4acffa /boot ext4 defaults 1 2
UUID=15afb55c-be85-44c1-9cbf-08abef602037 /home btrfs subvol=home,compress=zstd:1 0 0
alihosseinabadi@network ~]$
```

#fig:004



width:70%}

width:70%}

Чтобы скопировать ваш сгенерированный PGP ключ в буфер обмена, используйте:

`gpg --armor --export ваш_email@example.com`

```
ali hosseinabadi [Running] - Oracle VirtualBox
File Machine View Input Devices Help

foot 40% 100% 10.0.2.15/24 0% 27% us 100% 18:

root
git: is subject to change. To configure the initial branch name to use in all
git: of your new repositories, which will suppress this warning, call:
git:
git: git config --global init.defaultBranch <name>
git:
git: Names commonly chosen instead of 'master' are 'main', 'trunk' and
git: 'development'. The just-created branch can be renamed via this command:
git:
git: git branch -m <name>
Initialized empty Git repository in /home/alihosseinabadi/tutorial/.git/
alihosseinabadi@network tutorial]$ git config --list
user.name=alihosseinabadi
user.email=hosseinabadiia@gmail.com
core.quotepath=false
core.repositoryformatversion=0
core.filemode=true
core.bare=false
core.logallrefupdates=true
alihosseinabadi@network tutorial]$ git config init.defaultBranch master
alihosseinabadi@network tutorial]$ git init
Initialized existing Git repository in /home/alihosseinabadi/tutorial/.git/
alihosseinabadi@network tutorial]$ echo 'hello world' >hello.txt
alihosseinabadi@network tutorial]$ git add hello.txt
alihosseinabadi@network tutorial]$ git commit -m 'new file'
git: 'commit' is not a git command. See 'git --help'.

The most similar command is
commit
alihosseinabadi@network tutorial]$ git commit -m 'new file'
master (root-commit) 8802f01 new file
1 file changed, 1 insertion(+)
create mode 100644 hello.txt
alihosseinabadi@network tutorial]$ git status
bash: git status: command not found
alihosseinabadi@network tutorial]$ git status
On branch master
nothing to commit, working tree clean
alihosseinabadi@network tutorial]$ curl -L -s https://www.gitignore.io/api/c >> .gitignore
alihosseinabadi@network tutorial]$ curl -L -s https://www.gitignore.io/api/c++ >> .gitignore
alihosseinabadi@network tutorial]$ ssh-keygen -C "alihosseinabadi <work@nail>"
bash: ssh-keygen: command not found
alihosseinabadi@network tutorial]$ ssh-keygen -C "alihosseinabadi <hosseinabadiia@gmail.com>"
bash: ssh-keygen: command not found
alihosseinabadi@network tutorial]$ ssh-keygen
```

{#fig:006

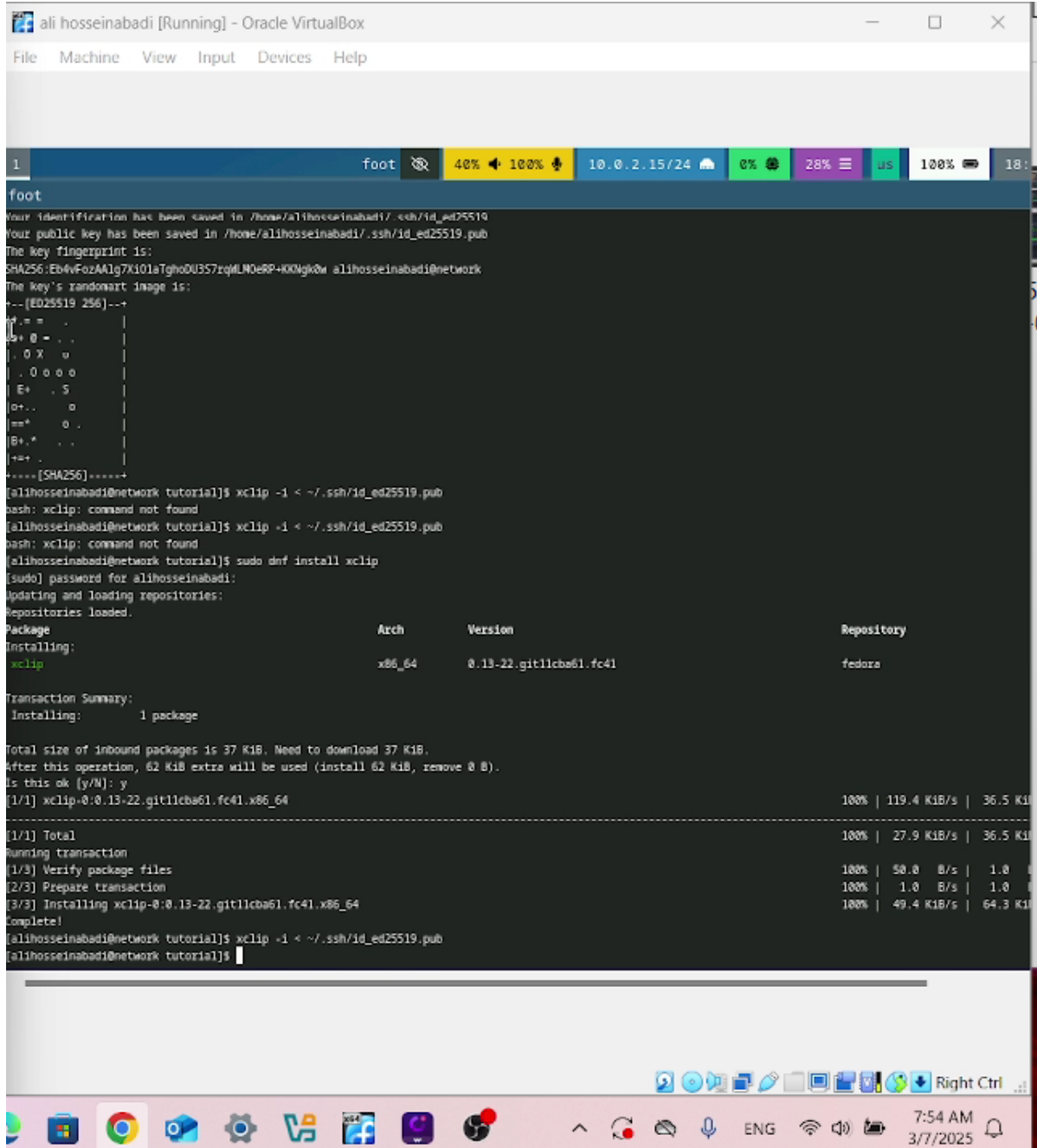
width:70%}

4.4 4. Настройка подписей Git

Для настройки автоматической подписи коммитов используйте следующие команды:


```
git config --global commit.gpgSign true
git config --global user.signingkey ваш_PGP_ключ
```

Эти команды позволят **Git** использовать указанный PGP ключ для подписи КОММИТОВ.



{#fig:007

width:70%}

4.5 5. Регистрация на GitHub

Необходимо авторизоваться на **GitHub**. Перейдите на сайт и выполните вход в свой аккаунт.

4.6 6. Создание локального каталога для выполнения заданий

Создайте локальный каталог для хранения ваших заданий по предмету:

```
mkdir название_каталога
```

```
cd название_каталога
```

Мы создали каталог для нашего курса.

4.7 Использование репозитория

После создания каталога вы можете использовать репозиторий **yamadharm**:

4.7.1 Удаление лишних файлов

Удалите ненужные файлы с помощью команды:

```
rm ненужный_файл
```

4.7.2 Создание необходимых каталогов

Создайте все необходимые подкаталоги для организации вашего проекта:

```
mkdir подкаталог1 подкаталог2
```


4.8 Выводы

В результате выполнения задания вы:

- Настроили **Git** с вашим именем и email.
- Сгенерировали и добавили SSH ключ на **GitHub**.
- Создали PGP ключ и настроили автоматическую подпись коммитов.
- Зарегистрировались на **GitHub** и создали локальный каталог для работы.

Эти шаги являются основными для успешной работы с **Git** и **GitHub**.

В этой лаборатории с помощью команд мы создаем новый репозиторий. Мы также смогли создать новый каталог с информацией о новом курсе. Как связать нашу учетную запись Github, чтобы мы могли работать, а также как сохранять, а затем загружать документы.

5 Ответы на контрольные вопросы.

1. Что такое системы контроля версий (VCS) и для решения каких задач они предназначаются? Системы контроля версий (VCS) — это инструменты, которые позволяют управлять изменениями в коде, документах или других файлах в проекте. Они решают следующие задачи:
2. Отслеживание изменений файлов.
3. Хранение истории изменений.
4. Совместная работа над проектом.
5. Возможность отката к предыдущим версиям.
6. Разрешение конфликтов при одновременном редактировании.
7. Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия.
8. Что представляют собой и чем отличаются централизованные и децентрализованные VCS? Приведите примеры VCS каждого вида.
9. Опишите действия с VCS при единоличной работе с хранилищем.
10. Опишите порядок работы с общим хранилищем VCS.
11. Каковы основные задачи, решаемые инструментальным средством git?
12. Назовите и дайте краткую характеристику командам git.
13. Приведите примеры использования при работе с локальным и удалённым репозиториями.
14. Что такое и зачем могут быть нужны ветви (branches)?
15. Как и зачем можно игнорировать некоторые файлы при commit?