

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«КРЫМСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ им. В. И. ВЕРНАДСКОГО»
ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ
Кафедра компьютерной инженерии и моделирования

ОТЧЕТ ПО ПРАКТИЧЕСКОМУ ЗАДАНИЮ №8
«Инсталляция и конфигурация ОС Arch GNU/Linux»

Практическая работа
по дисциплине «Современные технологии программирования»
студента 1 курса группы ПИ-б-о-231(2)
Аметов Кемран Ленверович
направления подготовки 09.03.04 «Программная инженерия»

Симферополь, 2024

Цель:

Познакомиться с основами установки ОС Arch GNU/Linux, а также подготовить базу для выполнения следующих лабораторных и практических занятий.

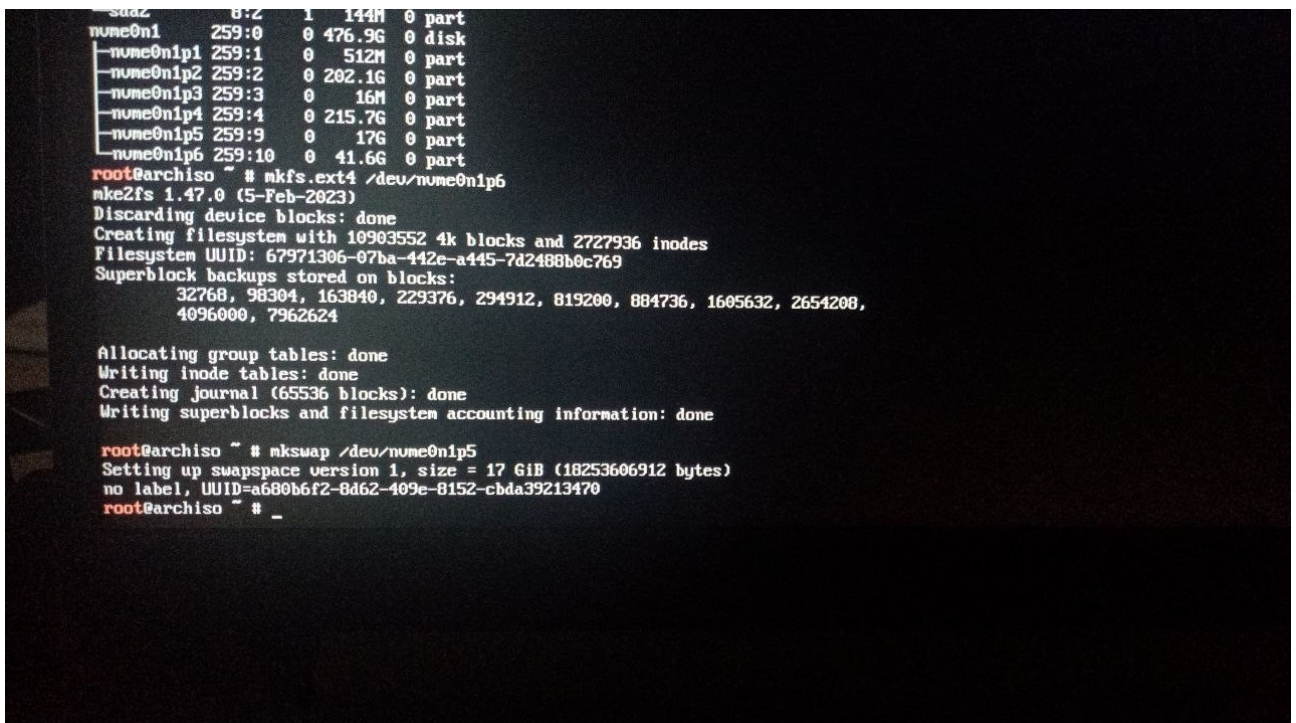
Ход выполнения задания.

программ.

1. Произведите установку операционной системы Arch GNU/Linux на реальный компьютер:
 - i. Загрузите и запишите [установочный образ Arch GNU/Linux](#) на съёмный носитель.
 - ii. Загрузитесь со съёмного носителя.
 - iii. Далее следуйте [официальным инструкциям установки](#).
 - а. Разметку дисков выполнил вручную с помощью cfdisk:
 - Корневая директория: 41 ГБ /dev/nvme0n1p6
 - Раздел подкачки: выделил ей размер ОЗУ + 1ГБ, в моём случае 17 гб /dev/nvme0n1p5



После создания разделов отформатировал основную директорию в ext4 с помощью mkfs.ext4. А своп раздел с помощью mkswap



После чего смонтировал корневой каталог в /mnt, а своп раздел активировал.

```

root@archiso ~ # mount --mkdir /dev/nvme0n1p6 /mnt
root@archiso ~ # swapon /dev/nvme0n1p5
root@archiso ~ #

```

Так же смонтирую каталог efi, у меня он уже был так что мне не нужно было его создавать.

Установим систему с помощью pacstrap

```

root@archiso ~ # pacstrap -K /mnt base linux firmware go python3 wget mesa pantheon-session lightdm-pantheon-greeter2-agent pantheon-polkit-agent pantheon-print pantheon-settings-daemon lightdm-pantheon-greeter pantheon-
can base-devel curl git intel-ucode sudo network-manager-applet nano openssh networkmanager ufw-supplciant dialog xorg dialog bluez f2fs-tools s2fsprogs
=> Creating install root at /mnt
gpg: /mnt/etc/pacman.d/gnupg/trustdb.gpg: trustdb created
gpg: no ultimately trusted keys found
gpg: starting migration from earlier GnuPG versions
gpg: porting secret keys from '/mnt/etc/pacman.d/gnupg/secring.gpg' to gpg-agent
gpg: migration succeeded
gpg: Generating pacman master key. This may take some time.
gpg: Generating pacman keyring master key...
gpg: directory '/mnt/etc/pacman.d/gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as '/mnt/etc/pacman.d/gnupg/openpgp-revocs.d/658F7E3282BA28784913513F996A2776225987.rev'
gpg: done
=> Updating trust database...
gpg: keyring needed: 3 completes needed: 1 trust model: gpg
gpg: depth: 0 valid: 1 ulgood: 0 trust: 0-, 0g, 0n, 0f, 1u
=> Installing packages to /mnt
:: Synchronizing package databases...
core
extra
error: target not found: lightdm-pantheon-greeter2-agent
error: target not found: switchboard
error: target not found: pantheon-screencast
error: target not found: pantheon-shortcut-overlay
error: target not found: ufw-supplciant
=> ERROR: failed to install packages to new root
1 root@archiso ~ # pacstrap -K /mnt base linux firmware go python3 wget mesa pantheon-session pantheon-greeter2-agent pantheon-polkit-agent pantheon-print pantheon-settings-daemon lightdm-pantheon-greeter pantheon-
tchboard cannet-assist epiphany pantheon-calculator pantheon-code pantheon-camera pantheon-files pantheon-music pantheon-photos pantheon-screenshot pantheon-shortcut-overlay pantheon-terminal pantheon-videos single-ssn base
intel-ucode sudo network-manager-applet nano openssh networkmanager ufw-supplciant dialog xorg dialog bluez f2fs-tools s2fsprogs

```

Далее с помощью genfstab создадим fstab. А заодно выставим свой регион(часовой пояс)


```

root@archiso ~ # genfstab -U /mnt >> /mnt/etc/fstab
root@archiso ~ # arch-chroot /mnt
[root@archiso /]# ln -sf /usr/share/
Display all 171 possibilities? (y or n)
[root@archiso /]# ln -sf /usr/share/zoneinfo/
Africa/          Australia/      Cuba/          Etc/            GMT+0          Iceland
America/         Brazil/        EET            Europe/         GMT-0          Indian/
Antarctica/      CET            EST            Factory         GMT0           Iran
Arctic/          CST6CDT        EST5EDT        GB              Greenwich      Israel
Asia/            Canada/        Egypt          GB-Eire         HST            Jamaica
Atlantic/        Chile/         Eire           GMT             Hongkong       Japan
[root@archiso /]# ln -sf /usr/share/zoneinfo/Europe/
Amsterdam  Belfast      Brussels      Chisinau      Guernsey      Jersey        Kyiv          Luxembourg    Minsk
Andorra    Belgrade     Bucharest     Copenhagen    Helsinki      Kaliningrad   Lisbon        Madrid        Monaco
Astrakhan  Berlin       Budapest     Dublin        Isle_of_Man   Kiev          Ljubljana    Malta         Moscow
Athens     Bratislava   Busingen     Gibraltar     Istanbul      Kirou         London       Mariehamn     Nicosia
[root@archiso /]# ln -sf /usr/share/zoneinfo/Europe/Moscow /etc/localtime
[root@archiso /]# hwclock --systohc
[root@archiso /]#

```

Выберу необходимые мне локали(в /etc/locale.gen) и сгенерирую их с помощью locale-gen.

```

#pt_PT.UTF-8 UTF-8
#pt_PT ISO-8859-1
#pt_PT@euro ISO-8859-15
#quz_PE UTF-8
#raj_IN UTF-8
#rif_MA UTF-8
#ro_RO.UTF-8 UTF-8
#ro_RO ISO-8859-2
#ru_RU.KOI8-R KOI8-R
ru_RU.UTF-8 UTF-8
#ru_RU ISO-8859-5
#ru_UA.UTF-8 UTF-8
#ru_UA KOI8-U
#rw_RW UTF-8
#sa_IN UTF-8
#sah_RU UTF-8
#sat_IN UTF-8
#sc_IT UTF-8
#sd_IN UTF-8
#sd_IN@devanagari UTF-8

```

```

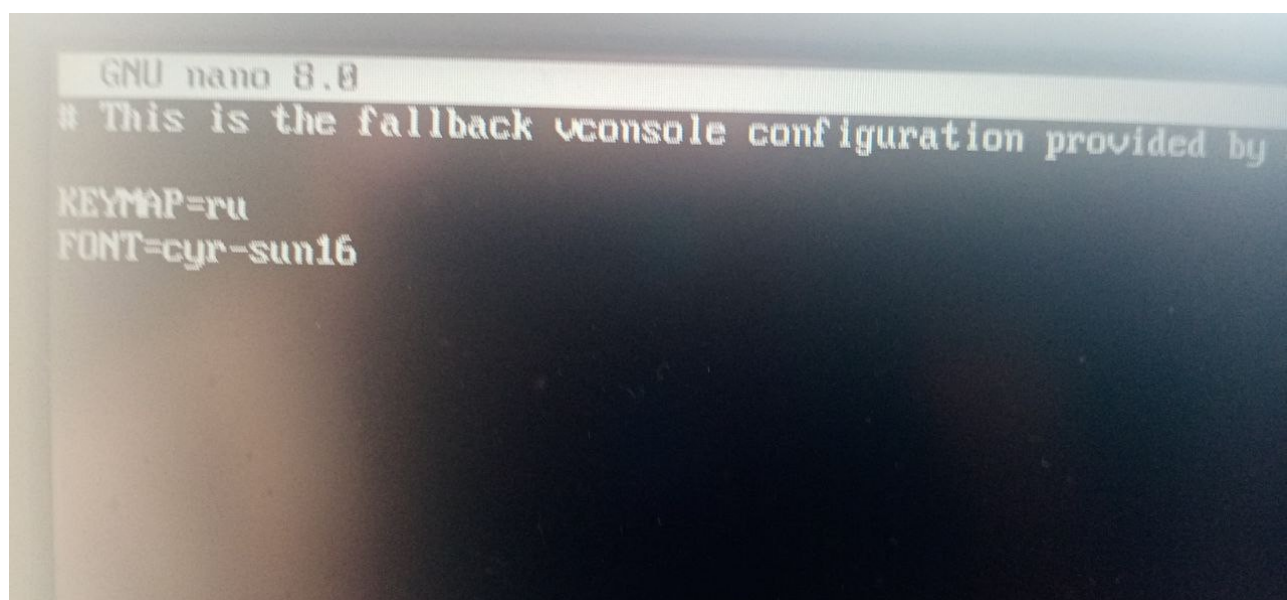
#en_US ISO-8859-1
en_US.UTF-8 UTF-8
#en_US ISO-8859-1
#en_ZA.UTF-8 UTF-8
#en_ZA ISO-8859-1
#en_ZM UTF-8

```

```
[root@archiso /]# locale-gen
Generating locales...
  en_US.UTF-8... done
  ru_RU.UTF-8... done
Generation complete.
[root@archiso /]# _
```

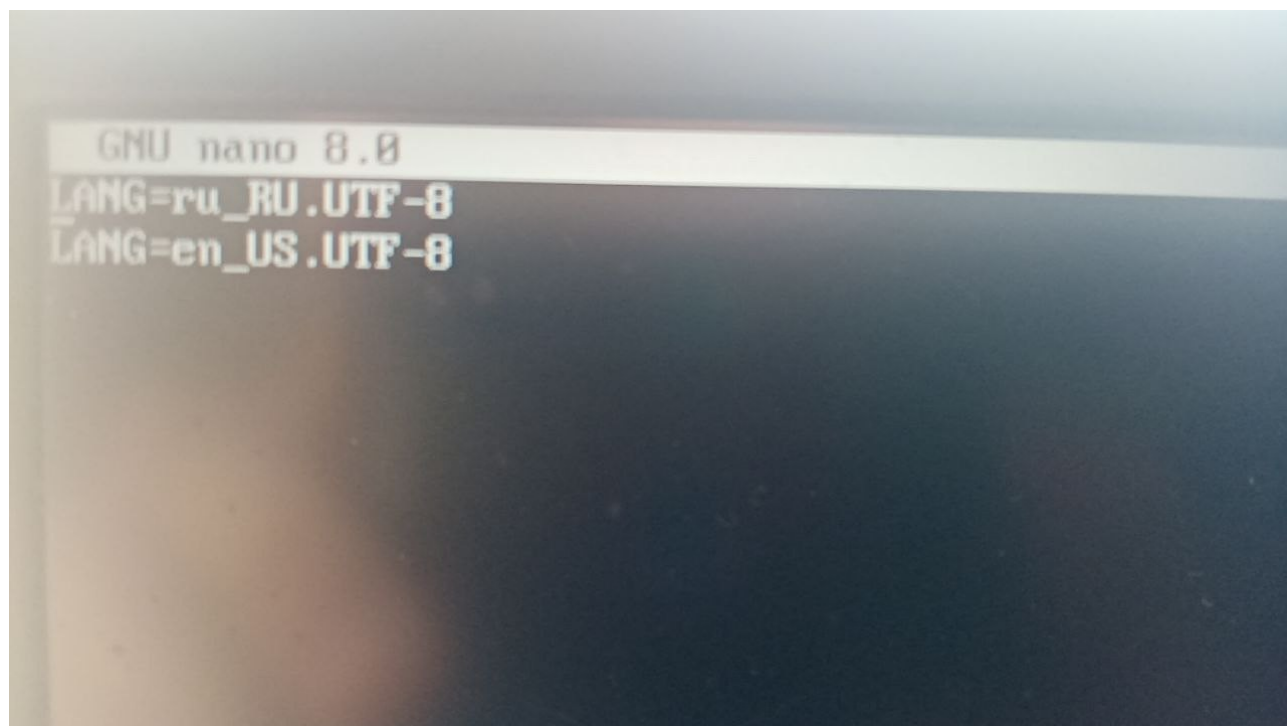
Так же нельзя забыть прописать свой язык в /etc/vconsole.conf и /etc/locale.conf

/etc/vconsole.conf

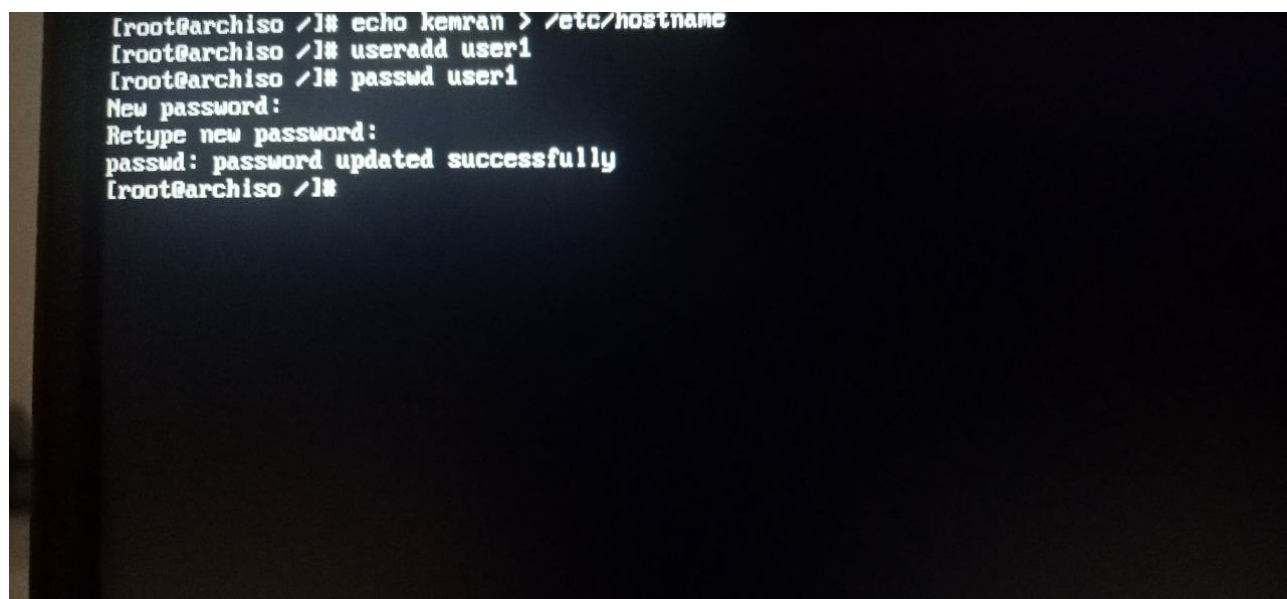


```
GNU nano 8.8
# This is the fallback vconsole configuration provided by
KEYMAP=ru
FONT=cyr-sun16
```

/etc/locale.conf



Далее настроим сеть записав имя хоста и создам пользователя user1



Установлю граб(у меня нет другого скрина, потому что я делал это в 2 часа ночи и под конец не смотрел на качество скринов, простите меня пожалуйста)


```
root@archiso /# grub-install --target=x86_64-efi --efi-directory=/boot --bootloader=/usr/sbin/grub2-install
Installing for x86_64-efi platform.
Installation finished. No error reported.
root@archiso /# grub-mkconfig -o /boot/grub/grub.cfg
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-linux
Found initrd image: /boot/initramfs-linux.img
Found fallback initrd image(s) in /boot: initramfs-linux-fallback.img
Warning: os-prober will not be executed to detect other bootable partitions.
Systems on them will not be added to the GRUB boot configuration.
Check GRUB_DISABLE_OS_PROBER documentation entry.
Adding boot menu entry for UEFI Firmware Settings ...
done
root@archiso /#
```

Выполним вход от рута в установленную систему

```
Arch Linux 6.8.9-arch1-2 (tty1)

kernan login: root
Password:
[root@kernan ~]#
```

2. Выполните вход в установленную систему с учётной записью администратора и настройте систему.
 - i. Установите следующее программное обеспечение:
 - SSH сервер и клиент
 - утилиты Midnight Commander, vi improved, http

```

warning: openssh-9.7p1-2 is up to date -- reinstalling
error: target not found: vi-improved
error: target not found: midnight-commander
[root@kemran ~]# pacman -S openssh vimproved htop midnight-commander
warning: openssh-9.7p1-2 is up to date -- reinstalling
error: target not found: vimproved
error: target not found: midnight-commander
[root@kemran ~]# pacman -S openssh vim htop mc_

```

- ii. Добавьте в систему пользователей user2, user3 со следующими параметрами:
 - Основная группа: users
 - Присутствует в группах: audio, video

```

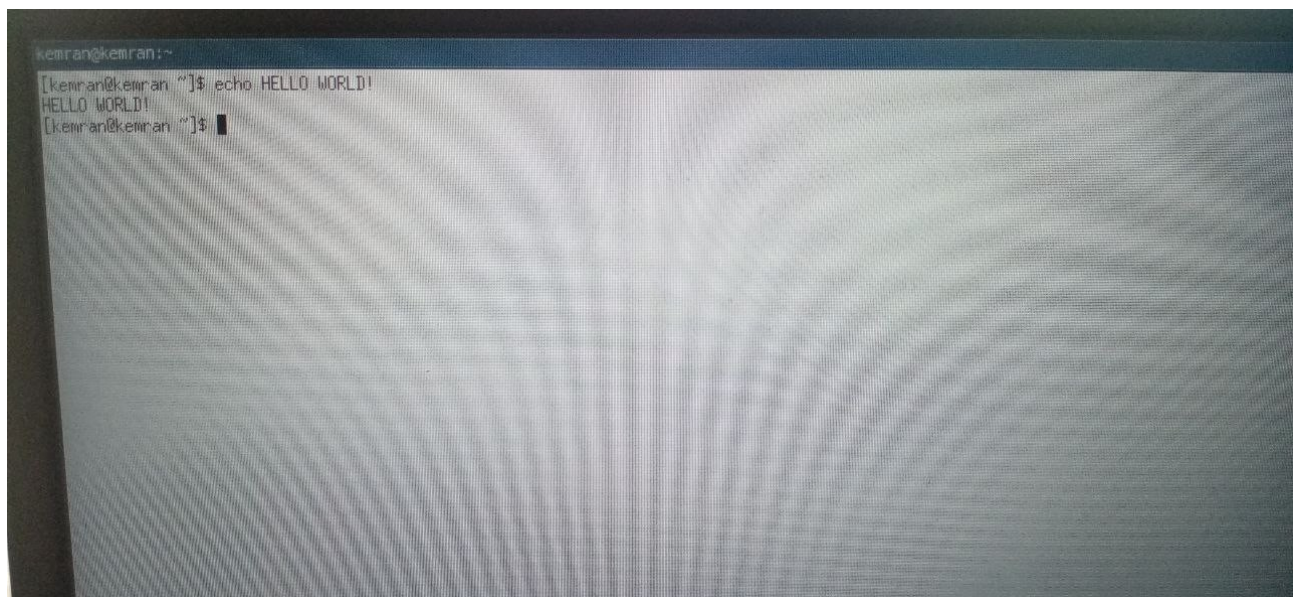
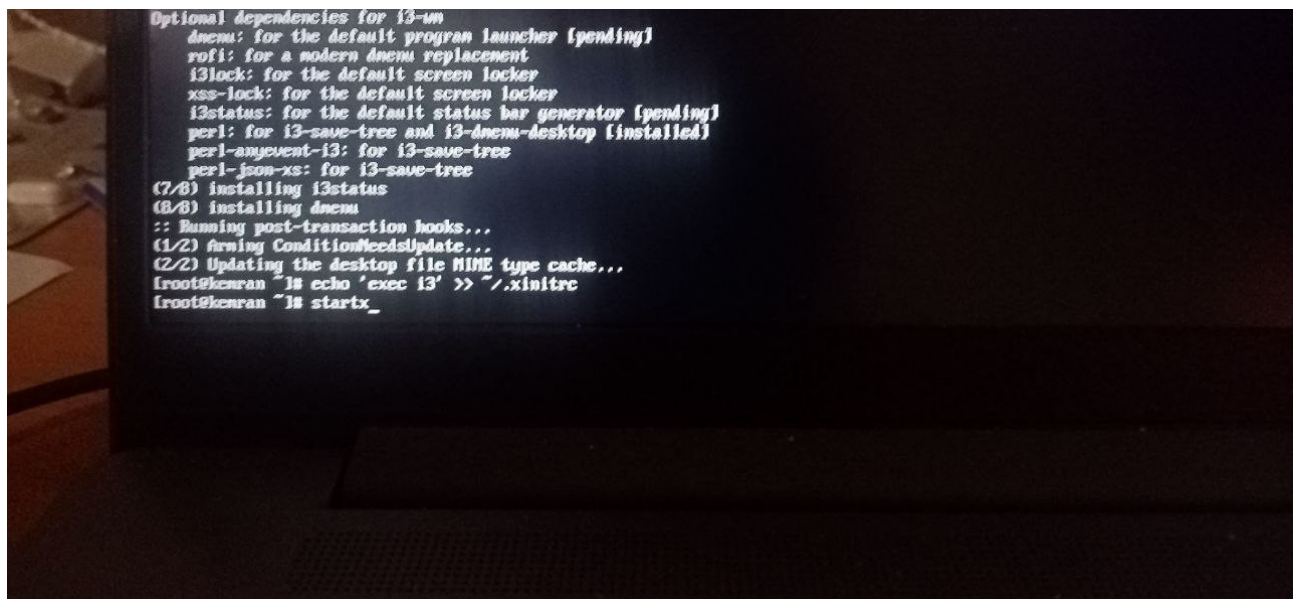
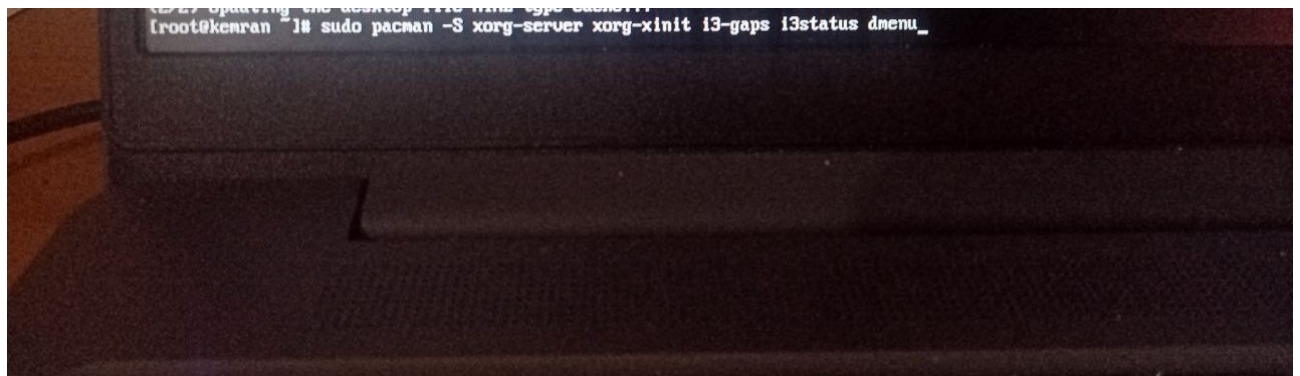
-O, --user-group          create a group with the same name as the user

[root@kemran ~]# useradd user2
[root@kemran ~]# useradd user3
[root@kemran ~]# usermod -a -G audio user2
[root@kemran ~]# usermod -a -G audio user3
[root@kemran ~]# usermod -a -G video user2
[root@kemran ~]# usermod -a -G video user3
[root@kemran ~]# addgroup users
-bash: addgroup: command not found
[root@kemran ~]# groupadd users
groupadd: group 'users' already exists
[root@kemran ~]# usermod -a -G users user2
[root@kemran ~]# usermod -a -G users user3
[root@kemran ~]# _

```

- iii. Установите графический интерфейс пользователя.

Установил i3wm (сначала попытался пансеон, но как то всё сломал... решил не морочить голову и просто поставить тайловый менеджер)



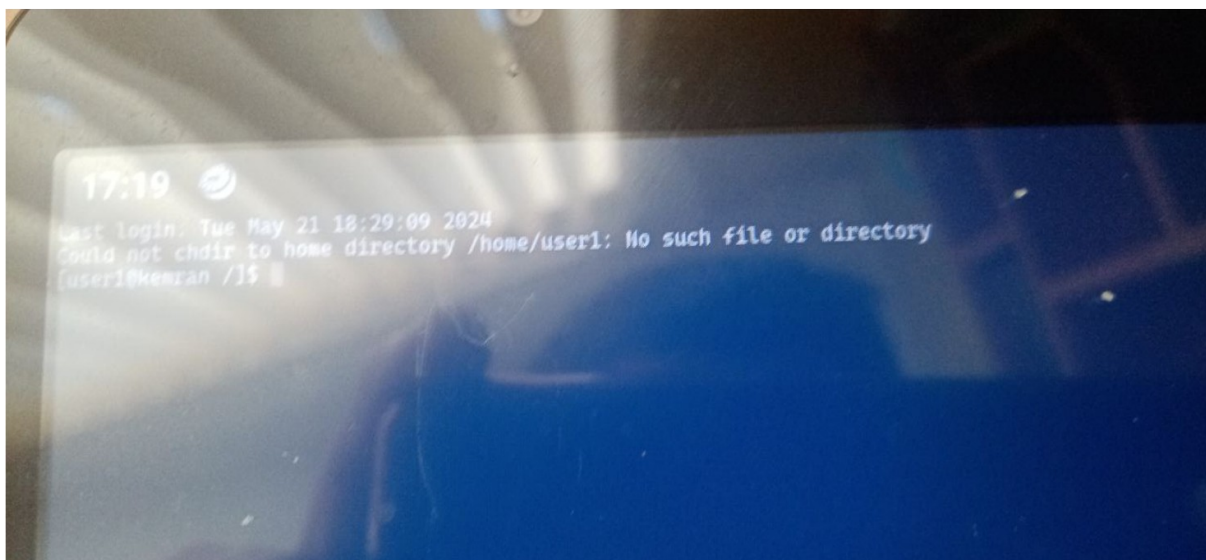
3. Создайте общую сеть Wi-Fi на другом устройстве или используйте существующую, подключитесь к ней с установленной системы.

Общая сеть будет точка доступа моего телефона, а подключаться к компьютеру я буду с помощью планшета.

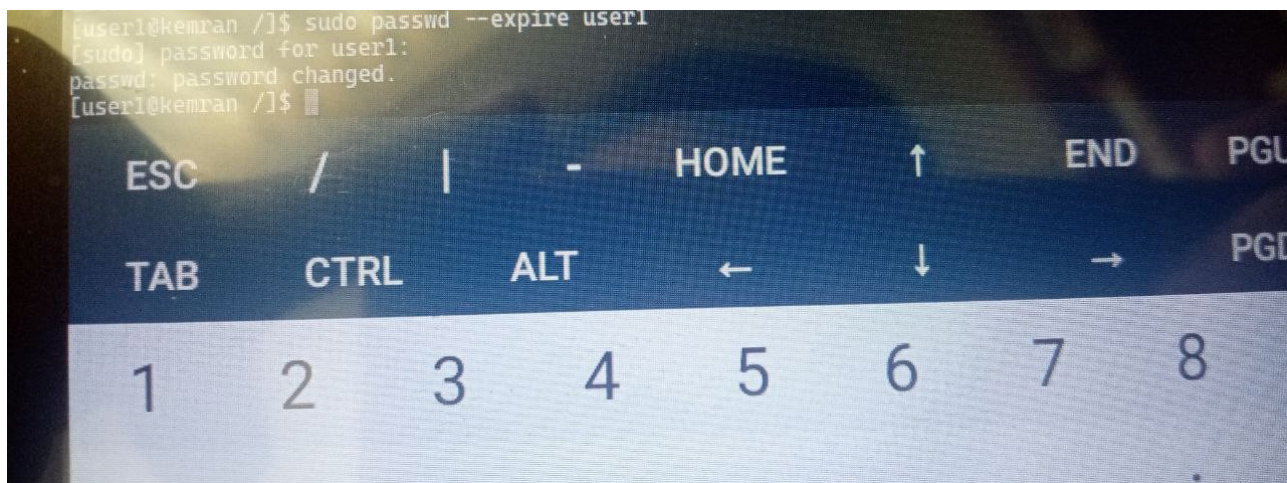
4. Если настройка системы произведена верно, тогда Вы сможете выполнить подключение к удалённому терминалу установленной операционной системы по протоколу ssh с устройства находящегося в локальной сети, созданной на предыдущем пункте, с помощью команды:

```
ssh -p <порт> root@ip_адрес
```

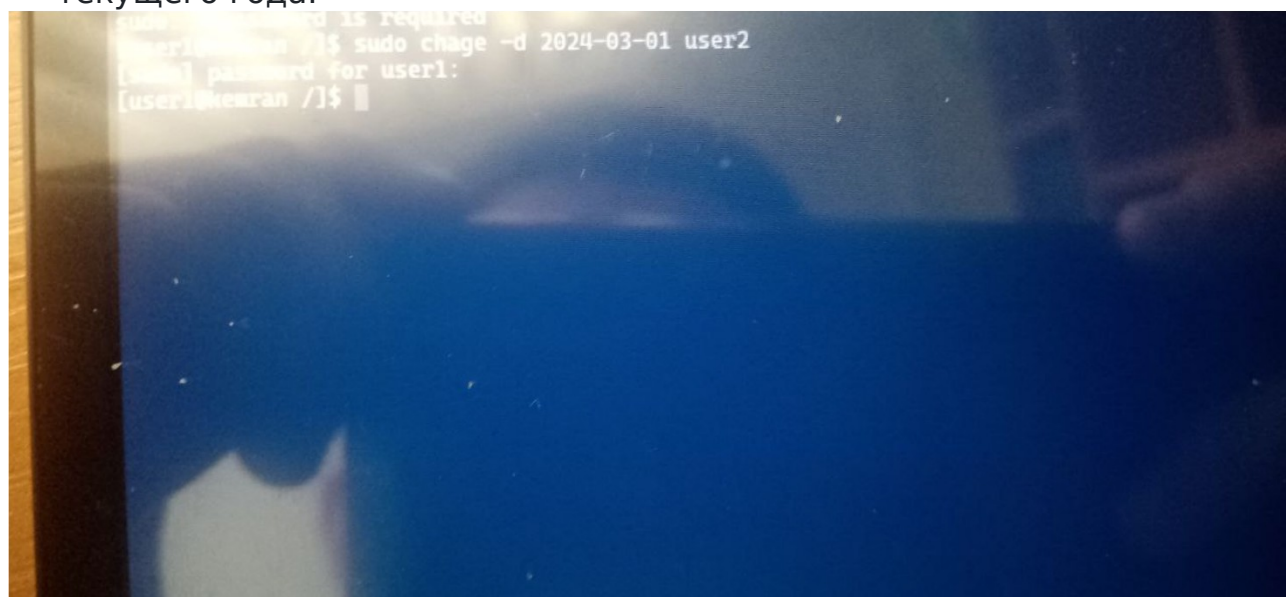
После входа Вы можете выполнять команды на удалённой операционной системе. Такой способ управления машиной является рекомендуемым, т.к. более удобен в использовании. Последующие настройки, указанные в пунктах ниже, необходимо выполнять удалённо по протоколу ssh.



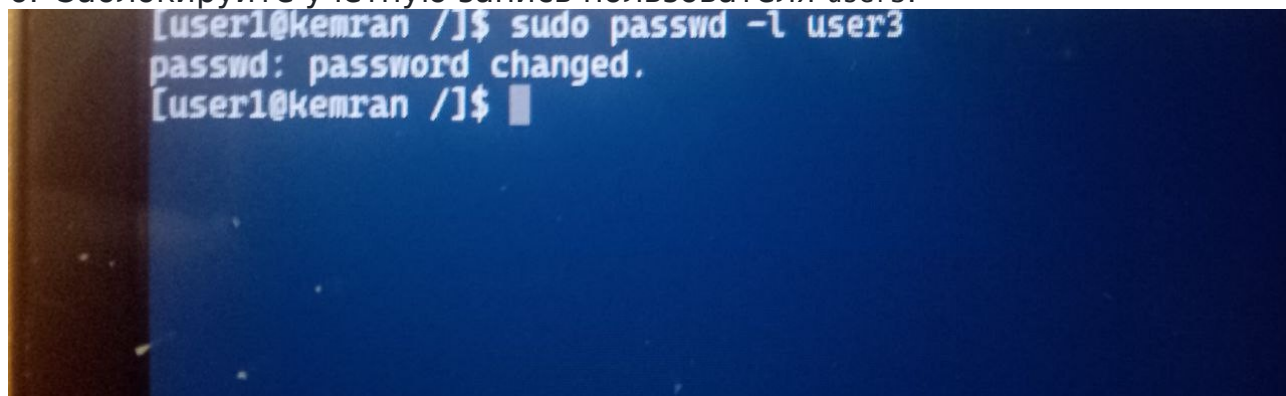
Настройте пароль пользователя user1 (не изменяя его) таким образом, чтобы пользователь был обязан сменить пароль при следующем входе.



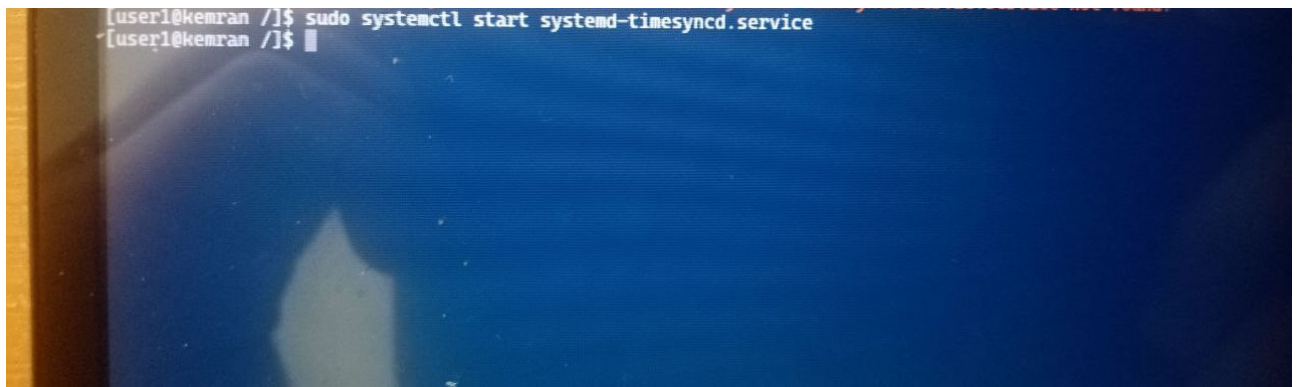
5. Учётная запись пользователя user2 должна устареть после 1 марта текущего года.



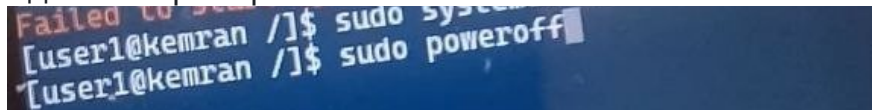
6. Заблокируйте учётную запись пользователя user3.



7. Настройте синхронизацию времени по протоколу NTP.



8. Выключите удалённую машину исполнив команду `poweroff` от имени администратора.



P.S (написано после сдачи, у меня даже видео есть с подключением через ssh. я до сих пор ничего не понимаю, потому что я **ВООБЩЕ НИЧЕГО НЕ МЕНЯЛ** с момента моего подключения через ssh
<https://youtube.com/shorts/W05bzlzIKDM?feature=share>)

Ответы на вопросы.

Какой тип распространения использует Arch GNU/Linux?

Arch Linux использует тип распространения, называемый "rolling release" (непрерывное обновление). В этой модели обновления и улучшения пакетов происходят непрерывно, без необходимости переустановки операционной системы для получения новейших версий программного обеспечения.

1. Что такое GRUB?

GRUB (Grand Unified Bootloader) — это загрузчик, который используется для загрузки операционных систем. Он поддерживает множество операционных систем и позволяет пользователю выбирать между ними во время загрузки компьютера. GRUB может быть настроен для загрузки различных конфигураций ядер и параметров.

2. С помощью какой команды происходит установка нового ПО в операционной системе Arch GNU/Linux?

В Arch Linux установка нового программного обеспечения производится с помощью пакетного менеджера `pacman`. Основная команда для установки пакетов:

- `sudo pacman -S package_name`
- **Какие способы установки ПО (форматы пакетов) существуют в сообществе GNU/Linux?**

В сообществе GNU/Linux существует несколько способов установки программного обеспечения, включая:

- **Бинарные пакеты:** Такие как `.deb` (Debian, Ubuntu) и `.rpm` (Fedora, CentOS).
- **Исходные коды:** Программное обеспечение компилируется из исходных кодов, например, используя команды `./configure`, `make`, `make install`.

- **Универсальные форматы:** Например, AppImage, Flatpak, Snap, которые могут быть использованы на разных дистрибутивах.
- **AUR (Arch User Repository):** Специфичный для Arch Linux репозиторий, в котором пользователи могут публиковать PKGBUILD-скрипты для сборки пакетов.
- **Какая команда используется для установки пароля пользователю?**

Команда для установки или изменения пароля пользователя в Linux:

- `passwd`
- **Для чего в процессе установки используются команды:**
- **`pacstrap`**

Команда `pacstrap` используется для первоначальной установки базовой системы Arch Linux на новую файловую систему. Обычно она применяется во время установки на новый диск или раздел.

- **`arch-chroot`**

Команда `arch-chroot` используется для смены корневого окружения на другой каталог (обычно новый корневой каталог системы). Это позволяет выполнять команды и настраивать новую систему как если бы она была загружена.

- **`locale-gen`**

Команда `locale-gen` генерирует локализации (наборы языковых параметров) для системы, основываясь на конфигурации в файле `/etc/locale.gen`. Это необходимо для корректной поддержки различных языков и региональных настроек.

- **`mkinitcpio`**

Команда `mkinitcpio` создаёт начальный RAM-диск (`initramfs`), который используется для загрузки системы. Этот RAM-диск содержит минимально необходимое окружение для того, чтобы ядро могло монтировать корневую файловую систему и продолжить загрузку.

Вывод:

Я познакомился с основами установки ОС Arch GNU/Linux, а также подготовил базу для выполнения следующих лабораторных и практических занятий.