



پروژه سفارشی‌سازی و ساخت یک سیستم‌عامل لینوکس سبک وزن
استاد درس: دکتر علی بهلوانی

نام و نامخانوادگی: عرفان فرهمند‌نژاد
شماره دانشجویی: 4013613056

۱- با استفاده از Ctrl + Alt + T یک ترمینال باز کرده و دستور sudo apt update && sudo apt upgrade وارد میکنیم:

```
erf@erf-VMware-Virtual-Platform: ~ $ sudo apt update && sudo apt upgrade
[sudo] password for erf:
[sudo] password for erf:
Hit:1 http://ir.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://ir.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://ir.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
6 packages can be upgraded. Run 'apt list --upgradable' to see them.
sudo: upgrade: command not found
erf@erf-VMware-Virtual-Platform: ~
```

پس از وارد کردن دستور بالا و انجام شدن بروزرسانی ها، محیط پایاده سازی کاملاً آماده است و می توان به مرحله بعد دستور کار رفت.

نکته: همانطور که اشاره شد، برای ادامه پایاده سازی نیازی به نصب برنامه خاصی نیست اما برای راحتی در محیط لینوکس می تواند برنامه ویرایش من دخواه با برنامه دیگر دخواه خودتان را نصب کنید (دستور sudo به بورز معلی امتحانات بورز روت را می دهد).

بخش دوم: بیلدروت و ساخت لینوکس

قبل از نصب برنامه بیلدروت، باید برنامه ها و ابزارهای مورد نیاز و پیش نیازهای بیلدروت نصب شود. این پیش نیازها به احتمال زیاد در سیستم عامل شما از قبیل وجود دارد اما برای اطمینان از دستور زیر برای نصب برنامه ها استفاده شود.

```
sudo apt install build-essential binutils rsync wget libncurses-dev
libelf-dev libssl-dev
```

نصب بیلدروت

۲- دستور sudo apt install build-essential binutils rsync wget libncurses-dev libelf-dev libssl-dev را استفاده کنید تا برنامه ها و ابزارهای مورد نیاز و پیش نیازهای بیلدروت نصب شود.

```
erf@erf-VMware-Virtual-Platform: ~ $ sudo apt install build-essential binutils rsync wget libncurses-dev libelf-dev libssl-dev
[sudo] password for erf:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
binutils is already the newest version (2.42-4ubuntu2.5).
binutils set to manually installed.
rsync is already the newest version (3.2.7-1ubuntu1.2).
rsync set to manually installed.
wget is already the newest version (1.21.4-1ubuntu4.1).
wget set to manually installed.
The following additional packages will be installed:
bzip2 dkpg-dev fakeroot g++-13 g++-13-x86-64-linux-gnu
g++-x86-64-linux-gnu libalgroithm-diff-perl libalgroithm-diff-xs-perl
libalgroithm-merge-perl libdpkg-perl libfakeroot libfile-fcntllock-perl
libstdc++-13-dev libbzstd-dev lto-disabled-list make zlib1g-dev
Suggested packages:
bzip2-doc debian-keyring g++-multilib g++-13-multilib gcc-13-doc git bzip2
libbz2-doc libbz2-doc libstdc++-13-doc make-doc
The following NEW packages will be installed:
build-essential bzip2 dkpg-dev fakeroot g++-13 g++-13-x86-64-linux-gnu
g++-x86-64-linux-gnu libalgroithm-diff-perl libalgroithm-diff-xs-perl
libalgroithm-merge-perl libdpkg-perl libelf-dev libfakeroot
libfile-fcntllock-perl libncurses-dev libbss1-dev libstdc++-13-dev
```

پس از وارد کردن دستور بالا و انجام شدن بروزرسانی ها، محیط پایاده سازی کاملاً آماده است و می توان به مرحله بعد دستور کار رفت.

نکته: همانطور که اشاره شد، برای ادامه پایاده سازی نیازی به نصب برنامه خاصی نیست اما برای راحتی در محیط لینوکس می تواند برنامه ویرایش من دخواه با برنامه دیگر دخواه خودتان را نصب کنید (دستور sudo به بورز معلی امتحانات بورز روت را می دهد).

بخش دوم: بیلدروت و ساخت لینوکس

قبل از نصب برنامه بیلدروت، باید برنامه ها و ابزارهای مورد نیاز و پیش نیازهای بیلدروت نصب شود. این پیش نیازها به احتمال زیاد در سیستم عامل شما از قبیل وجود دارد اما برای اطمینان از دستور زیر برای نصب برنامه ها استفاده شود.

```
sudo apt install build-essential binutils rsync wget libncurses-dev
libelf-dev libssl-dev
```

نصب بیلدروت

پس از نصب ابزارهای مورد نیاز باید بیلدروت نصب شود. برای نصب بیلدروت در روش وجود دارد.

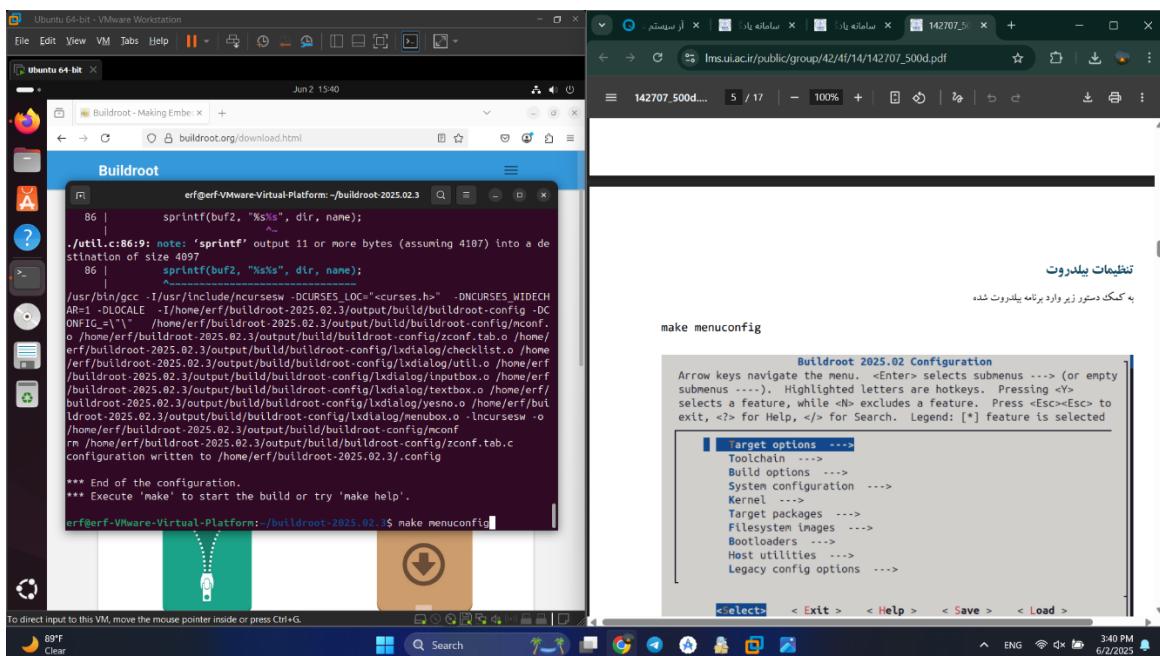
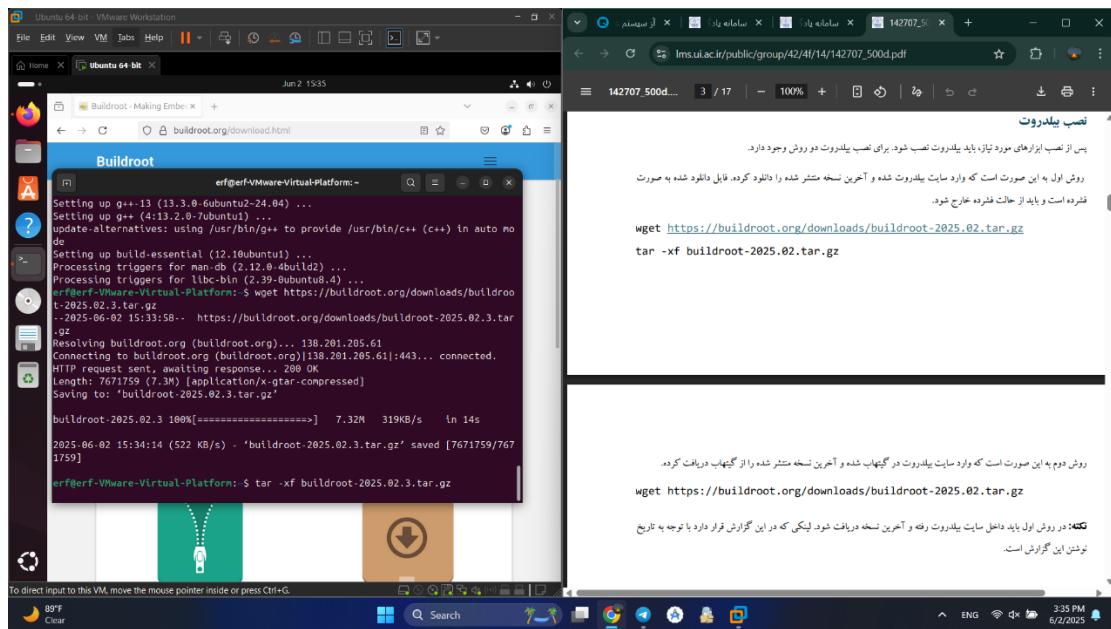
روش اول به این صورت است که وارد سایت بیلدروت شده و آخرین نسخه منتشر شده را دانلود کرده، قابل دانلود شده به صورت فشرده است و باید از حالت فشرده خارج شود.

```
wget https://buildroot.org/downloads/buildroot-2025.02.tar.gz
tar -xf buildroot-2025.02.tar.gz
```

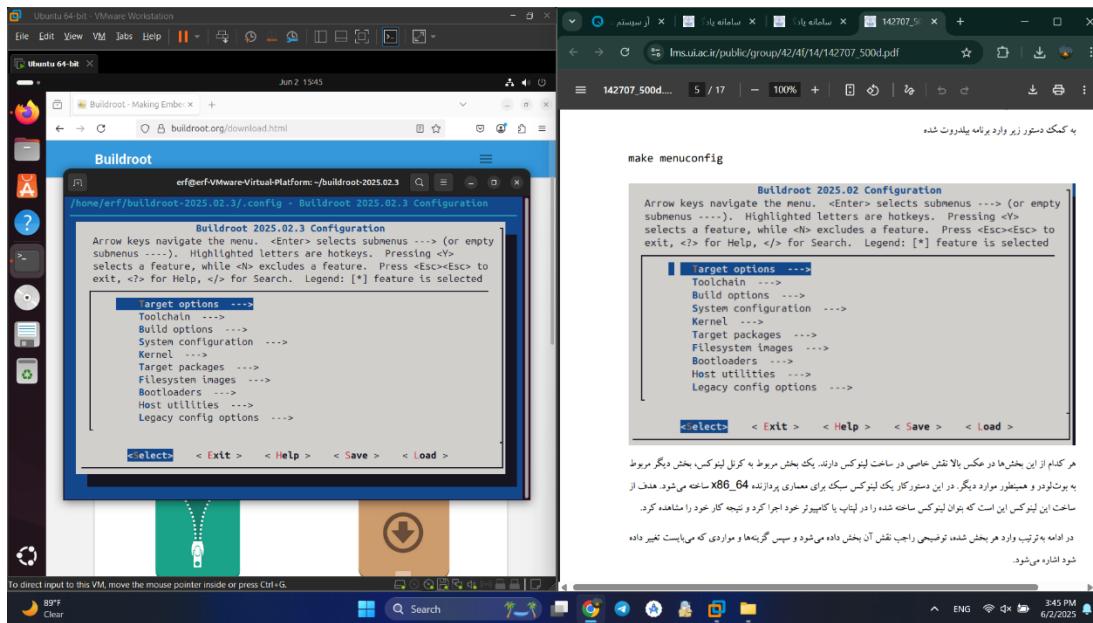
3- برای نصب بیلدروت اقدام می‌کنیم و آخرین نسخه منتشر شده را دانلود می‌کنیم و از حالت فشرده خارج می‌کنیم:

wget <https://buildroot.org/downloads/buildroot-2025.02.3.tar.gz>

tar -xf buildroot-2025.02.3.tar.gz



4- با زدن دستور make menuconfig (که البته فقط در آدرس /buildroot-2025.02.3 عمل میکرد) وارد برنامه بیلدروت میشویم:

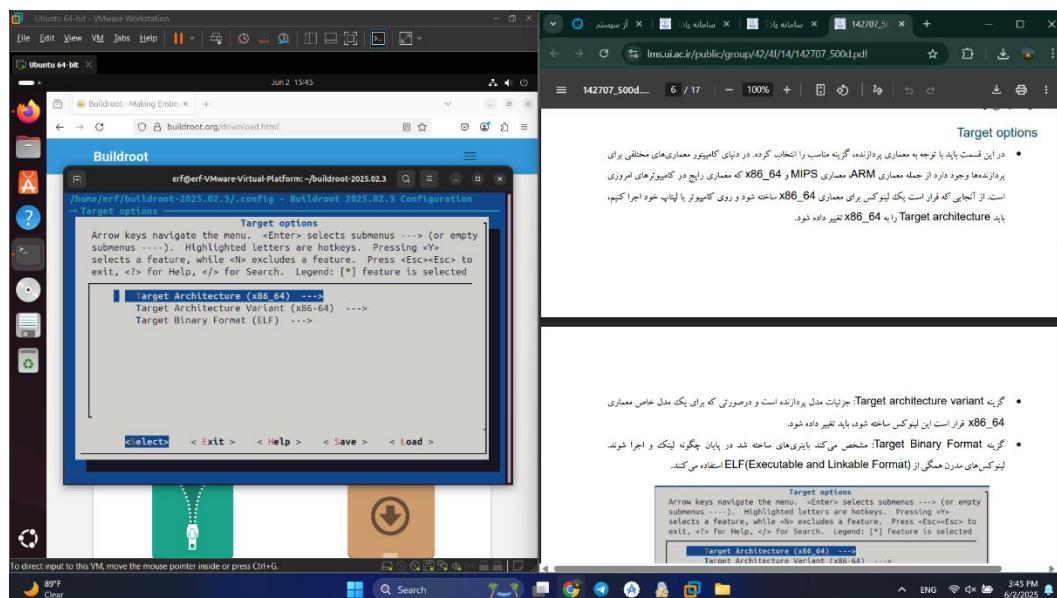


5- در مرحلهی بعد وارد قسمت Target options میشویم و تنظیمات زیر را طبق دستور کار انجام میدهیم:

Target Architecture → x86-64

Target Architecture Variant → x86-64

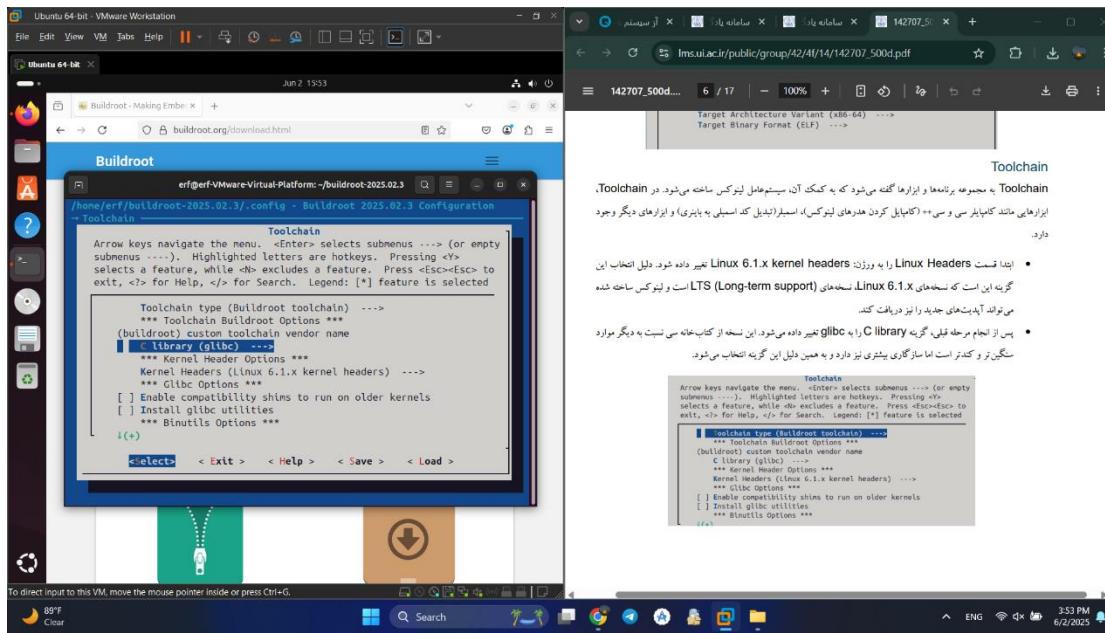
Target Binary Format → ELF



6- وارد قسمت Toolchain شده و تنظیمات زیر را طبق دستور کار انجام می‌دهیم:

Linux Headers → Linux 6.1.x kernel headers

(که البته بود) C library → glibc



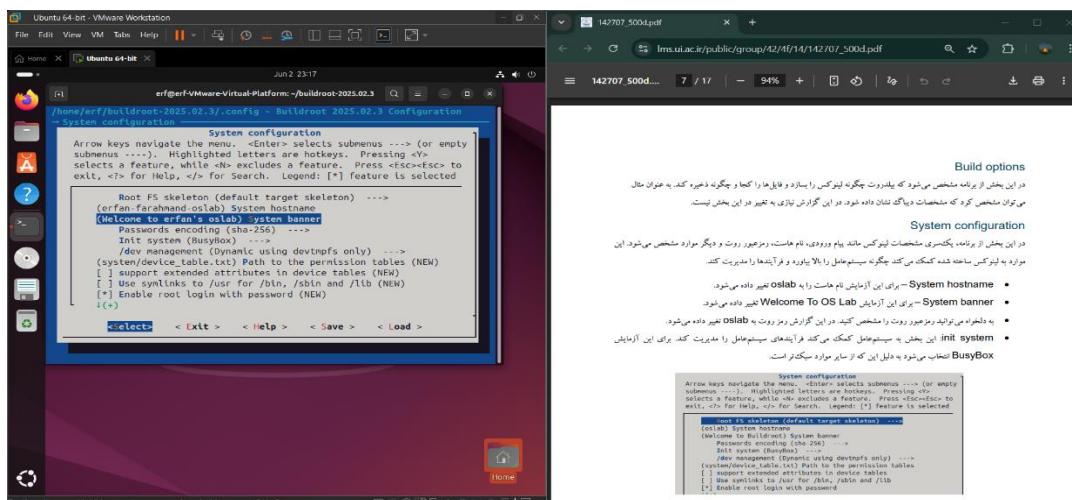
7- وارد قسمت System configuration شده و تنظیمات زیر را انجام می‌دهیم:

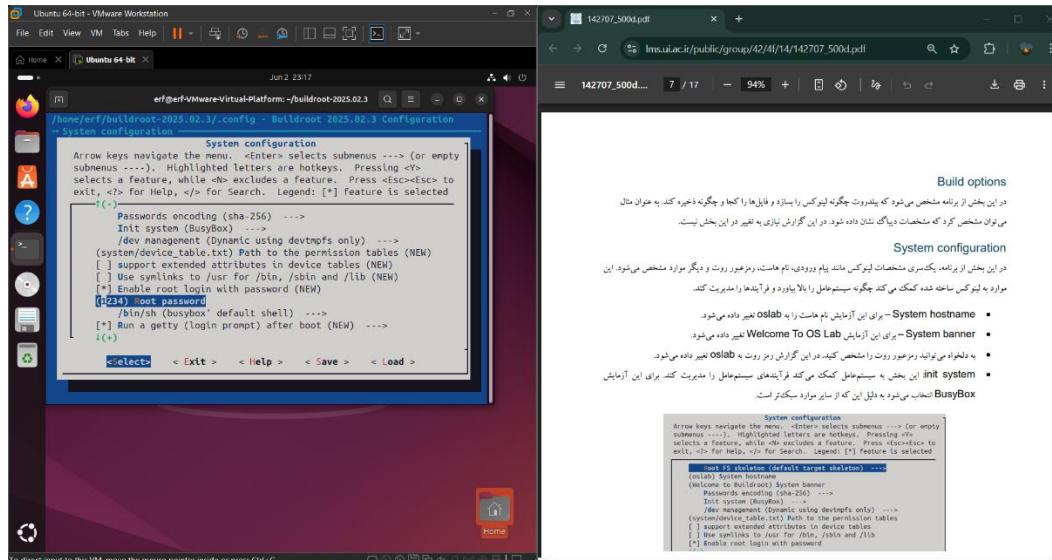
System hostname → erfan-farahmand-oslab

System banner → Welcome to erfan's oslab

Init system → BusyBox

Root password → 1234





8- وارد بخش Kernel میشویم و تنظیمات زیر را انجام می‌دهیم:

Linux Kernel → On

Kenel Version → Custom version → 6.1.24

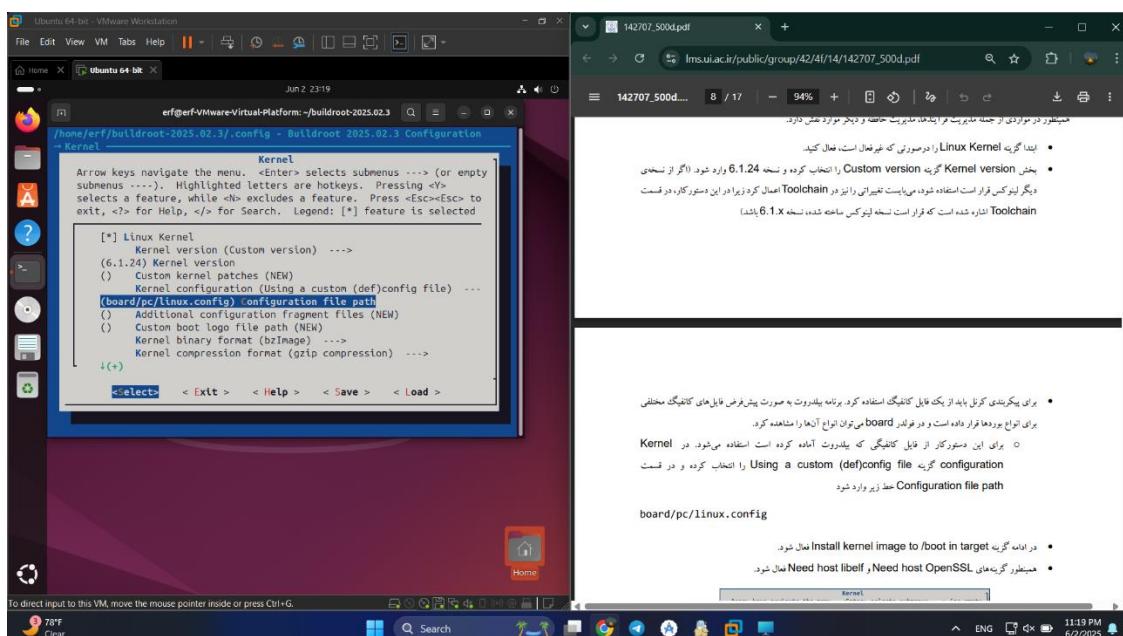
Kernel configuration → Using a custom (def) config file

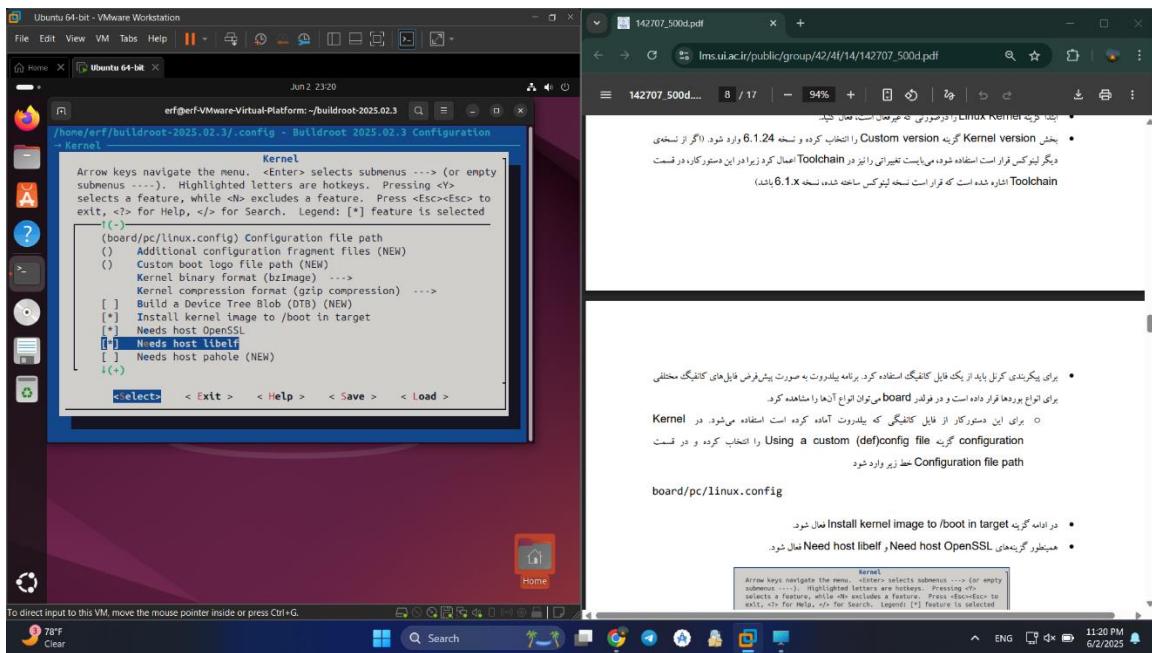
Configuration file pat → board/pc/linux.config

Install kernel image to /boot in target → On

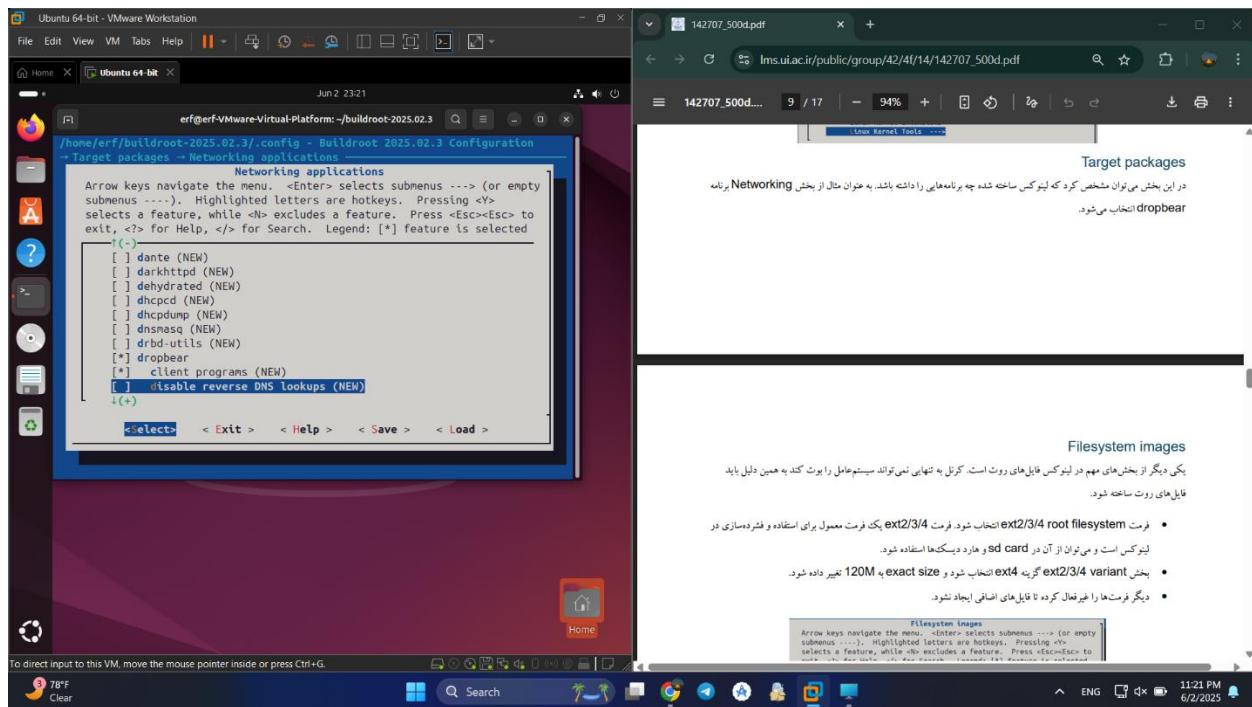
Need host OpenSSL → On

Need host libelf → on





9- وارد بخش Target packages میشویم و از بخش Networking applications گزینه dropbear را فعال میکنیم.

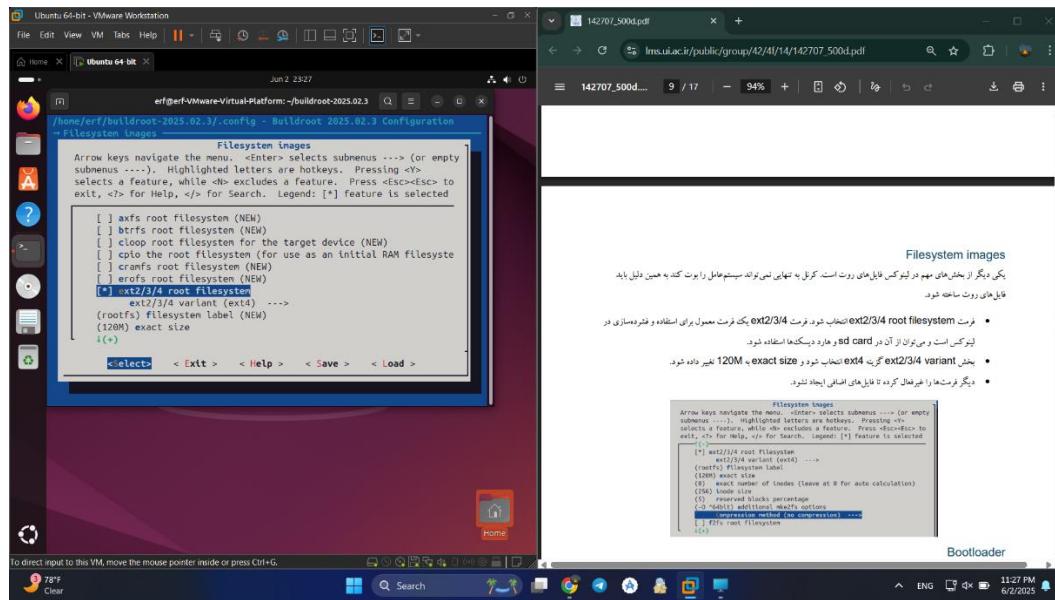


10- وارد قسمت شده و تنظیمات زیر را انجام می‌دهیم:

ext2/3/4 root filesystem → on

ext2/3/4 variant (ext4) → ext4

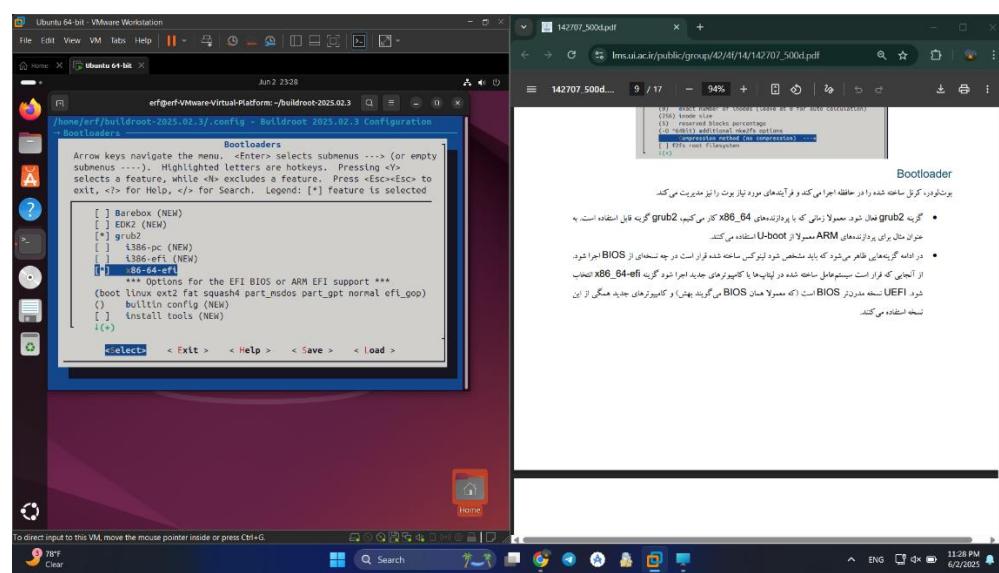
exact size → 120M



11- وارد بخش شده و تنظیمات زیر را انجام می‌دهیم:

Grub → on

x86-64-efi → on

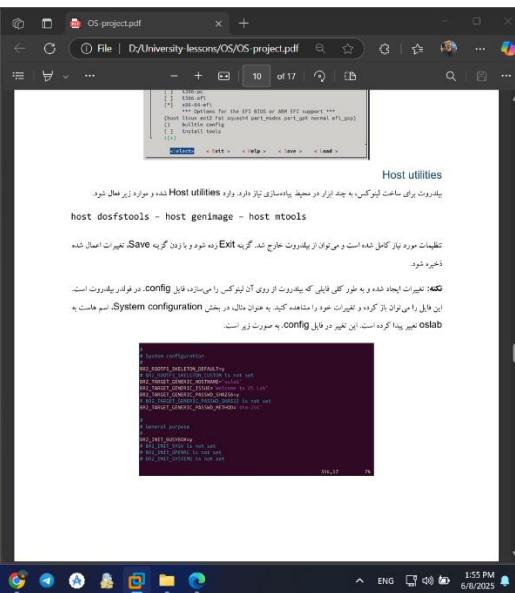
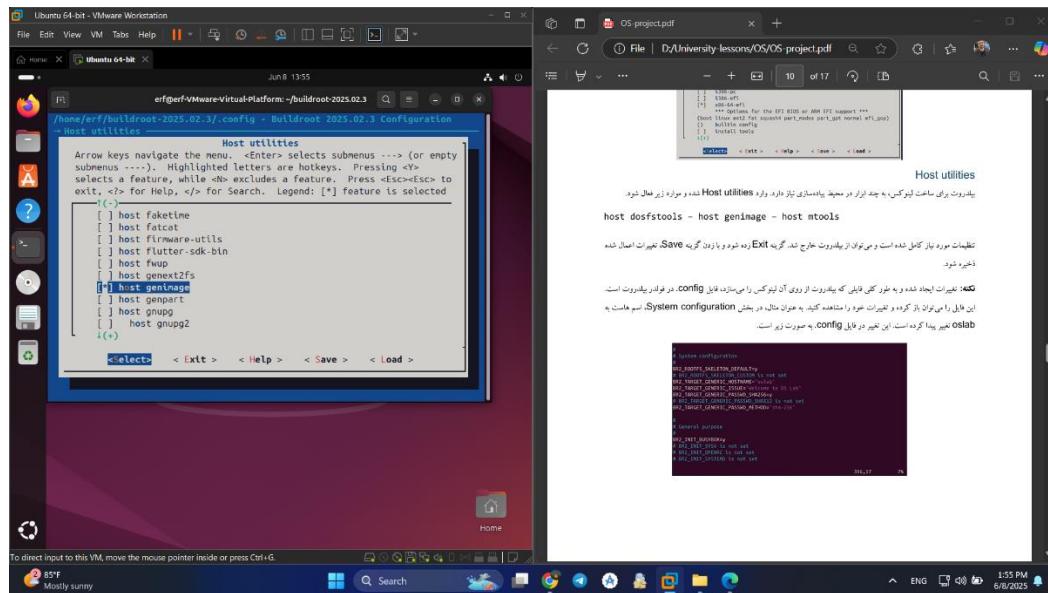
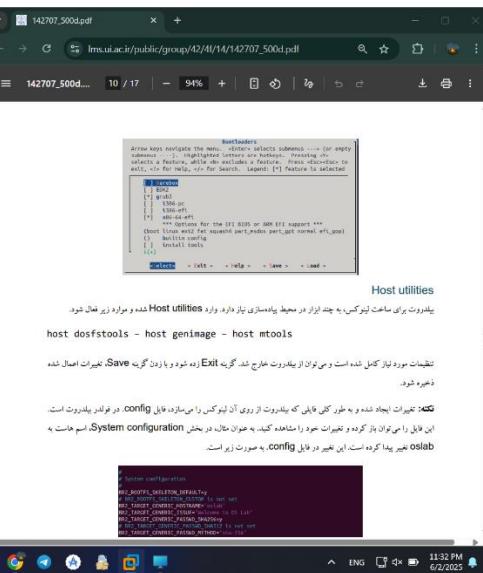
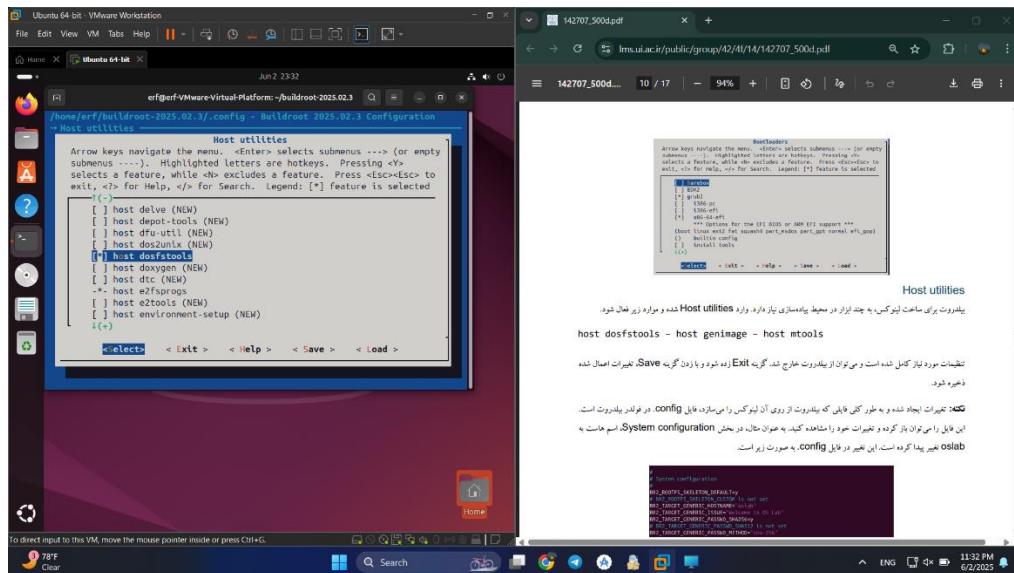


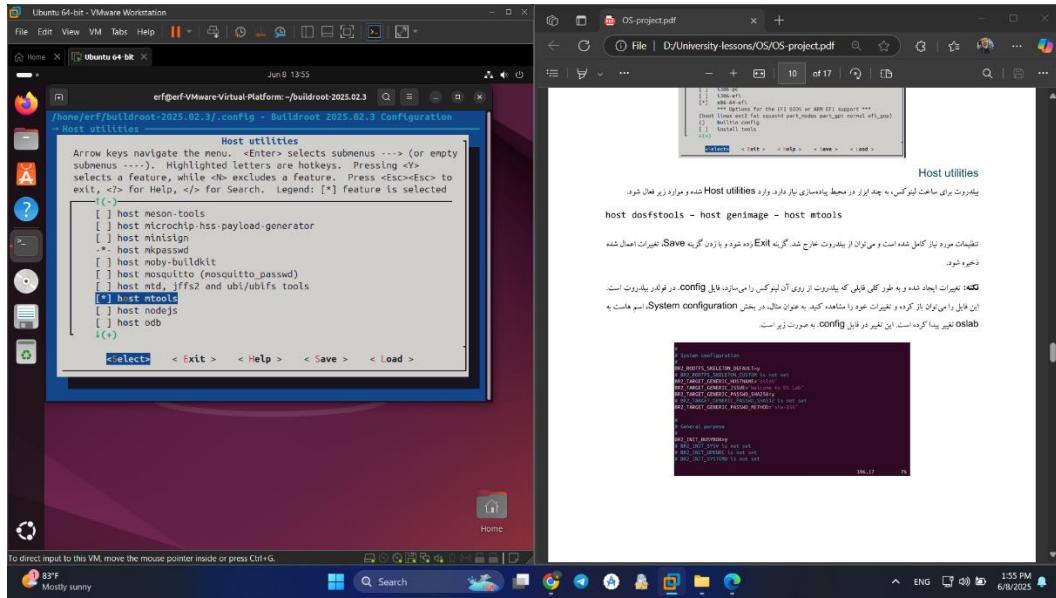
12- وارد قسمت Host utilities میشویم و تنظیمات زیر را اعمال میکنیم:

host dosfstools → on

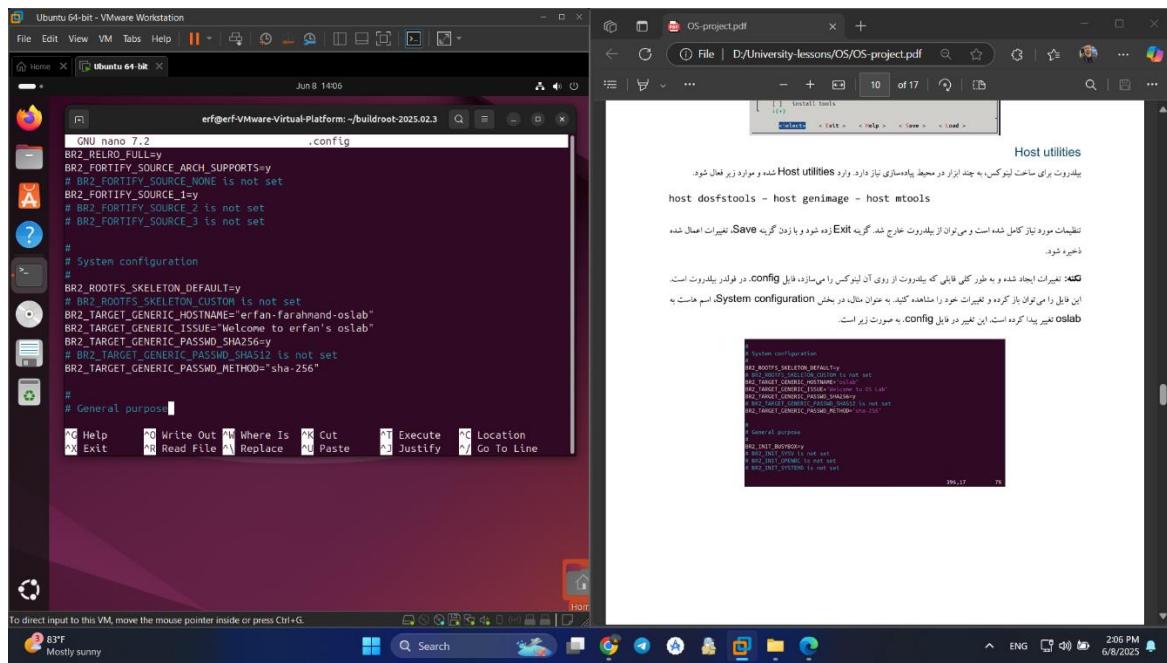
host genimage → on

host mtools → on

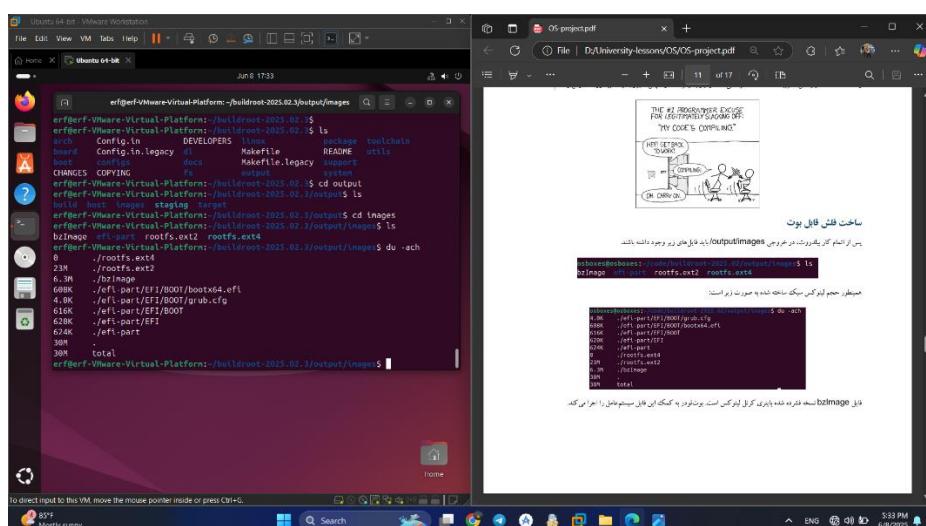
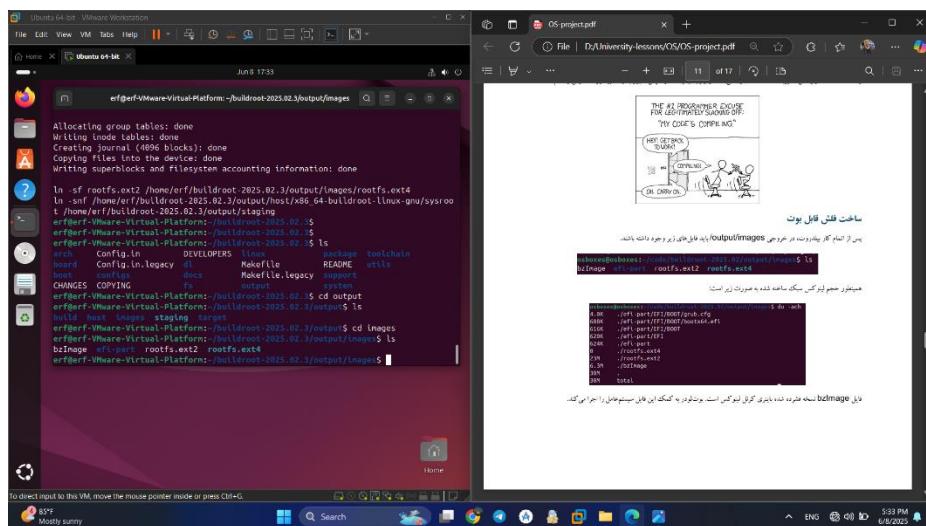
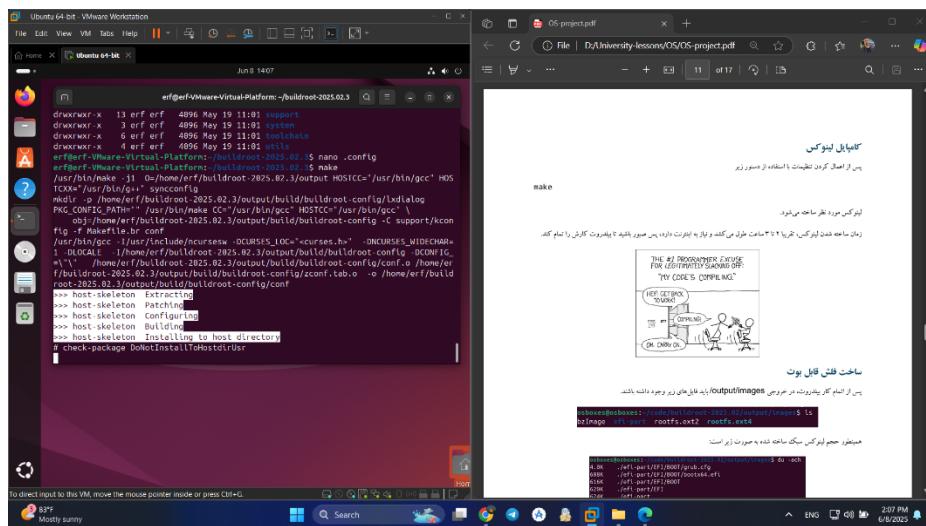




13- بعد از save فایل config. که در فولدر /buildroot-2025.02.3 باز میکنیم تا تغییرات انجام شده را مشاهده و از ذخیره‌ی آنها مطمئن بشیم:



14- پس از آن دستور make را وارد میکنیم تا لینوکس مورد نظرمان ساخته شود:



15- USB را میزنیم و سپس دستور lsblk را میکنیم:

```

erf@erf-VMware-Virtual-Platform:~/buildroot-2025.02.3/output/images$ lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0    2:0    1  1.4M  0 disk /media/erf/disk
loop0    7:0    0  4K  1 loop /snap/bare/5
loop1    7:1    0 258M  1 loop /snap/firefox/5751
loop2    7:2    0 73.9M  1 loop /snap/core22/1748
loop3    7:3    0 11.1M  1 loop /snap/firmware-updater/167
loop4    7:4    0 516M  1 loop /snap/gnome-42-2204/262
loop5    7:5    0 91.7M  1 loop /snap/gtk-common-themes/1535
loop6    7:6    0 10.8M  1 loop /snap/snap-store/1248
loop7    7:7    0 44.4M  1 loop /snap/snapd/23545
loop8    7:8    0 568K  1 loop /snap/snapd-desktop-integration/253
loop9    7:9    0 58.9M  1 loop /snap/snapd/24505
loop10   7:10   0 73.9M  1 loop /snap/core22/1963
sda     8:0    0 48G  0 disk
└─sda1   8:1    0  1M  0 part /
└─sda2   8:2    0 48G  0 part /
  └─sdb1  8:16   1 28.9G 0 disk
    └─sdb1  8:17   1 299M 0 part
  └─sdb2  8:18   1 28.6G 0 part /media/erf/f95595c-5ade-4dc0-8098-6a76bda9c025
sr0    11:0   1 89.4M 0 rom /media/erf/CDROM
sr1    11:1   1 1024M 0 rom

erf@erf-VMware-Virtual-Platform:~/buildroot-2025.02.3/output/images$
```

فولدر اطلاعات لازم برای بوت کردن سیستم‌های UEFI را دارد، همینطور یک فایل grub.cfg در آین فولدر وجود دارد که اطلاعات لازم برای بوت‌لودر grub2 در آن قرار دارد.

قابل image rootfs.ext2 است که فایل‌های روت نوبت‌گذاری شده در آن قرار دارد.

اکنون USB خود را وارد دستگاه کنید (فایل‌های USB یکتاپ گرفته شود زیرا فرآور است ظمامی فایل‌ها یکتاپ شود).

ابتدا با کنکنک دستور

```
lsblk
```

USB خود را بیدا کرده که معمولاً به صورت /dev/sdX وارد شده و دستگاه است. در آین مثال USB وارد شده دستگاه /dev/sdb است.

sdb	8:16	1	15G	0	disk
└─sdb1	8:17	1	299M	0	part
└─sdb2	8:18	1	1000M	0	part
sr0	11:0	1	1024M	0	rom

حال باید پارسین‌های موجود بایزی بوت کردن سیستم عامل درست شود برای آین کار به داریتین انجام می‌شود. یعنی پارسین با فرمت FAT که پایس بوت را تشخیص دهد و یک پارسین ext4 که فایل‌های روت در آن قرار می‌گیرد.

برای انجام این کار با دستور زیر وارد برنامه parted که می‌توان پارسین‌های سیستم را مدیریت کرد.

```

sudo parted /dev/sdb
mklabel gpt
mkpart ESP fat32 1MIB 300MIB
set 1 esp on
mkpart primary ext4 300MIB 100%
```

ابتدا پارسین ۰-۳۰۰ مگا با فرمت FAT و یک پارسین با فرمت ext4 که اندیح حافظه USB است درست می‌شود.

۴۰۰ در مجموعی که حافظه USB را می‌داند، پیشنهاد می‌شود پارسین ۳۰۰ مگا حافظه درست کرده تا در مرحله بعد رامزه فرمت کردن کوئید شود. برای آین کار می‌توان از دستور زیر بحثی دستور را لاله‌ساز کرد. در آین دستور، برای پارسین دو یک گلک درنظر گرفته شده است.

16- از اینجا به بعد را دو بار انجام دادم چون به شک درباره‌ی مرتبه اول خوردم و دوباره انجامش دادم اما در دفعه‌ی دوم همه چیز درست شد که اسم USB به sdc تغییر کرد، ابتدا سراغ بار اول میرویم که مشکلات به وجود آمدند و سپس تصاویر دومین تلاش را میبینیم که مشکلات برطرف شدند:

```

erf@erf-VMware-Virtual-Platform:~/buildroot-2025.02.3/output/images$ lsblk
loop6    7:6    0 10.8M 1 loop /snap/snap-store/1248
loop7    7:7    0 44.4M 1 loop /snap/snapd/23545
loop8    7:8    0 568K 1 loop /snap/snapd-desktop-integration/253
loop9    7:9    0 58.9M 1 loop /snap/snapd/24505
loop10   7:10   0 73.9M 1 loop /snap/core22/1963
sda     8:0    0 48G 0 disk
└─sda1   8:1    0 1M 0 part /
└─sda2   8:2    0 48G 0 part /
  └─sdb1  8:16   1 28.9G 0 disk
    └─sdb1  8:17   1 299M 0 part
  └─sdb2  8:18   1 28.6G 0 part /media/erf/f95595c-5ade-4dc0-8098-6a76bda9c025
sr0    11:0   1 89.4M 0 rom /media/erf/CDROM
sr1    11:1   1 1024M 0 rom

erf@erf-VMware-Virtual-Platform:~/buildroot-2025.02.3/output/images$ sudo parted /dev/sdb
mklabel gpt
mkpart ESP fat32 1MIB 300MIB
set 1 esp on
mkpart primary ext4 300MIB 100%
[sudo] password for erf:
GNU Parted 3.6
Using /dev/sdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted)

mklab gpt
mkpart ESP fat32 1MIB 300MIB
set 1 esp on
mkpart primary ext4 300MIB 100%
[sudo] password for erf:
GNU Parted 3.6
Using /dev/sdb
Welcome to GNU Parted! Type 'help' to view a list of commands.

mkpart primary ext4 300MIB 1300MIB

dr. مرحله بعد باید پارسین‌ها را فرمت کرد.

sudo mkfs.vfat -F32 /dev/sdb1
sudo mkfs.ext4 /dev/sdb2

در مرحله بعد آین دو پارسین ساخته شده را mount کرده و فایل‌های موردنیاز باید اضافه می‌شود.

برای mount کردن از دستورهای زیر استفاده شود.

sudo mkdir -p /mnt/efi /mnt/root
```

The screenshot shows a dual-boot environment. On the left, a terminal window on an Ubuntu 64-bit VM displays the command `mkpart primary ext4 300MB 1300MB` followed by `mkfs.vfat -F32 /dev/sdb1`. On the right, a Microsoft Edge browser window shows a PDF titled 'OS-project.pdf' with the text 'در مرحله بعد باید پارتیشن را فرست کرد.' (In the next step, you must send the partition). Below the browser is a taskbar with icons for File Explorer, Task View, Start, and other system icons.

```

mkpart primary ext4 300MB 1300MB
در مرحله بعد باید پارتیشن را فرست کرد.

sudo mkfs.vfat -F32 /dev/sdb1

```

** همانطور که در تصویر بالا مشاهده میکنیم لینوکس دستورات mklabel و mkpart را شناسایی نکرده است.

(که احتمالا به دلیل کپی کردن همه دستورات و پیست کردن آنها در ترمینال به صورت همزمان است در مرتبه دوم اینکار انجام نشده)

17- و در ادامه دستورات پس از ساخته شدن پارتیشن های مورد نیاز را در ترمینال وارد کردم:

The screenshot shows a dual-boot environment. On the left, a terminal window on an Ubuntu 64-bit VM displays the command `mkfs.vfat -F32 /dev/sdb1` followed by `mkfs.ext4 /dev/sdb2` and `mkfs.fat 4.2 (2021-01-31)`. On the right, a Windows File Explorer window shows a folder named 'temp' containing files 'boot', 'efi', and 'rootfs'. A context menu is open over the 'rootfs' file, with the option 'Copy' highlighted. Below the file explorer is a taskbar with icons for File Explorer, Task View, Start, and other system icons.

```

mkfs.vfat -F32 /dev/sdb1
mkfs.ext4 /dev/sdb2
mkfs.fat 4.2 (2021-01-31)

```

18- سیس در فایل grub.cfg موجود در /buildroot-2025.02.3/output/images/efi-part/EFI/BOOT

زیر را ایجاد میکنیم:

Sda1 → sda2

```

set default=0
set timeout=5
menuentry "Buildroot" {
    linux /boot/bzImage root=/dev/sda1 rootwait console=tty1
}

```

فستت نیز کرده و پاک گردید و موردنیزه شده از مستورات زیر استفاده شود.

```

sudo rm -rf /mnt/efi
sudo rm -rf /mnt/root
sync

```

19- و ادامه دستورات:

```

set timeout=5
menuentry "Buildroot" {
    linux /boot/bzImage root=/dev/sda2 root=UUID=00000000-0000-0000-0000-000000000000 rootwait console=tty1
}


```

در نهایت باید فایل زیر در پارتیشن گول پانداشته شود:

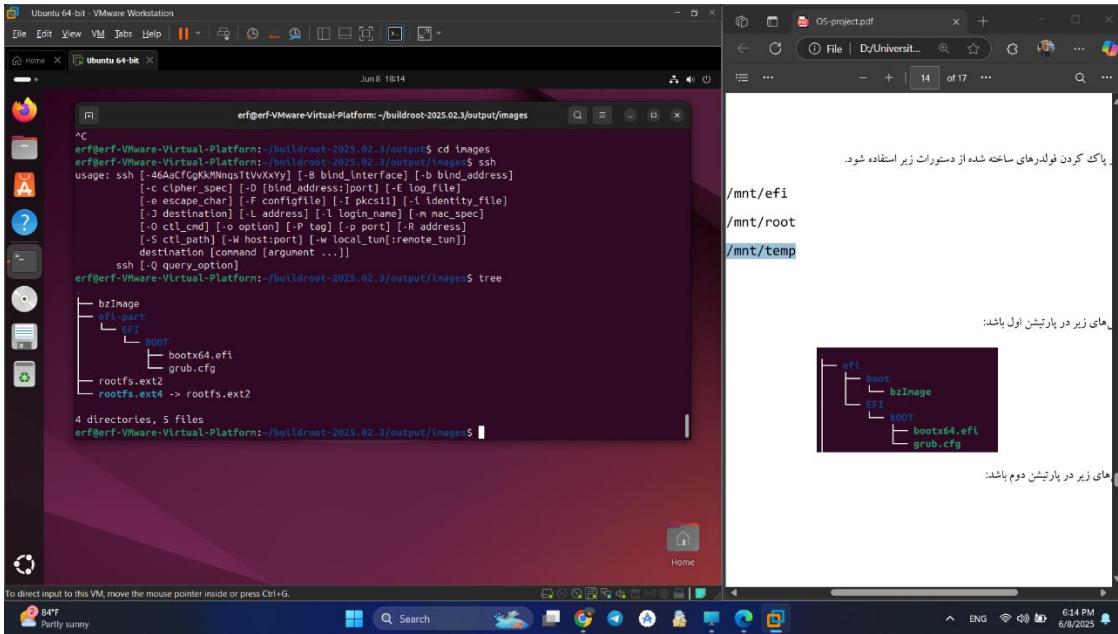
```

efi/
└── boot
    └── bzImage
    └── efi
        └── bootx64.efi
        └── grub.cfg

```

همینطور باید فایل زیر در پارتیشن گول پانداشته شود:

و ساختار فایل‌ها:



20- تکرار مراحل 17 تا 19 با : sdc



```

erf@erf-Virtual-Platform:/buildroot-2025.02.3/output/images$ sudo mkfs.vfat -F32 /dev/sdc1
mkfs.fat 4.2 (2021-01-31)
erf@erf-Virtual-Platform:/buildroot-2025.02.3/output/images$ sudo mkfs.ext4 /dev/sdc2
mke2fs 1.47.0 (5-Feb-2023)
/dev/sdc2 is mounted; will not make a filesystem here!
erf@erf-Virtual-Platform:/buildroot-2025.02.3/output/images$ sudo mkdir -p /mnt/efi /mnt/root
erf@erf-Virtual-Platform:/buildroot-2025.02.3/output/images$ sudo mount /dev/sdc1 /mnt/efi
erf@erf-Virtual-Platform:/buildroot-2025.02.3/output/images$ sudo mount /dev/sdc2 /mnt/root
erf@erf-Virtual-Platform:/buildroot-2025.02.3/output/images$ sudo mkdir -p /mnt/efi/boot
erf@erf-Virtual-Platform:/buildroot-2025.02.3/output/images$ sudo cp bzImage /mnt/efi/boot
erf@erf-Virtual-Platform:/buildroot-2025.02.3/output/images$ sudo cp vmlinuz /mnt/efi/boot
erf@erf-Virtual-Platform:/buildroot-2025.02.3/output/images$ sudo cp initrd /mnt/efi/boot
erf@erf-Virtual-Platform:/buildroot-2025.02.3/output/images$ sudo mkdir -p /mnt/temp/
erf@erf-Virtual-Platform:/buildroot-2025.02.3/output/images$ sudo cp -a /mnt/temp/* /mnt/root/
erf@erf-Virtual-Platform:/buildroot-2025.02.3/output/images$ sudo umount /mnt/temp/
erf@erf-Virtual-Platform:/buildroot-2025.02.3/output/images$ 

```

سیستم عاملی که باید در پارتیشن بوت ایجاد شود.

```

sudo mkdir -p /mnt/efi /mnt/root
sudo mount /dev/sdb1 /mnt/efi
sudo mount /dev/sdb2 /mnt/root

sudo mkdir -p /mnt/efi/boot
sudo cp bzImage /mnt/efi/boot
sudo cp -r efi-part/* /mnt/efi

sudo mkdir -p /mnt/temp/
sudo mount rootfs-ext2 /mnt/temp/
sudo cp -a /mnt/temp/* /mnt/root/
sudo umount /mnt/temp/

```

سیستم عاملی که دوست نیوکس را در پارتیشن بوت قرار داده می شود.

```

erf@erf-Virtual-Platform:/buildroot-2025.02.3/output/images$ sudo mount /dev/sdc2 /mnt/root
erf@erf-Virtual-Platform:/buildroot-2025.02.3/output/images$ sudo mkdir -p /mnt/efi/boot
erf@erf-Virtual-Platform:/buildroot-2025.02.3/output/images$ sudo cp bzImage /mnt/efi/boot
erf@erf-Virtual-Platform:/buildroot-2025.02.3/output/images$ sudo cp vmlinuz /mnt/efi/boot
erf@erf-Virtual-Platform:/buildroot-2025.02.3/output/images$ sudo cp initrd /mnt/efi/boot
erf@erf-Virtual-Platform:/buildroot-2025.02.3/output/images$ sudo mkdir -p /mnt/temp/
erf@erf-Virtual-Platform:/buildroot-2025.02.3/output/images$ sudo cp -a /mnt/temp/* /mnt/root/
erf@erf-Virtual-Platform:/buildroot-2025.02.3/output/images$ sudo umount /mnt/temp/
erf@erf-Virtual-Platform:/buildroot-2025.02.3/output/images$ ls
bzImage  efi-part  rootfs-ext2  rootfs.ext4
erf@erf-Virtual-Platform:/buildroot-2025.02.3/output/images$ ls
erf@erf-Virtual-Platform:/buildroot-2025.02.3/output/images$ cd efi-part
erf@erf-Virtual-Platform:/buildroot-2025.02.3/output/images$ cd EFI/BOOT
erf@erf-Virtual-Platform:/buildroot-2025.02.3/output/images$ ls
bootx64.efi  grub.cfg
erf@erf-Virtual-Platform:/buildroot-2025.02.3/output/images$ nano grub.cfg
erf@erf-Virtual-Platform:/buildroot-2025.02.3/output/images$ sudo umount /mnt/
erf@erf-Virtual-Platform:/buildroot-2025.02.3/output/images$ sync
erf@erf-Virtual-Platform:/buildroot-2025.02.3/output/images$ sudo rm -rf /mnt/efi
erf@erf-Virtual-Platform:/buildroot-2025.02.3/output/images$ sudo rm -rf /mnt/root
erf@erf-Virtual-Platform:/buildroot-2025.02.3/output/images$ sudo rm -rf /mnt/temp

```

نیز برای داده های رایانه که در پارتیشن بوت قرار دارند، سیستم عاملی که داده های رایانه را در پارتیشن بوت قرار داده است.

```

set default="0"
set timeout="5"
menuentry "Buildroot" {
    linux /boot/bzImage root=/dev/sda1 rootwait console=tty1
}

برای تسبیح و پاک کردن فولدرهای ساخت شده از دستورات زیر استفاده شود.

sudo rm -rf /mnt/efi
sudo rm -rf /mnt/root
sudo rm -rf /mnt/temp

```

در غایب باید فایل های زیر در پارتیشن اول باشد.

و طبق قبل دستورات زیر رو هم زدیم:

`sudo rm -rf /mnt/efi`

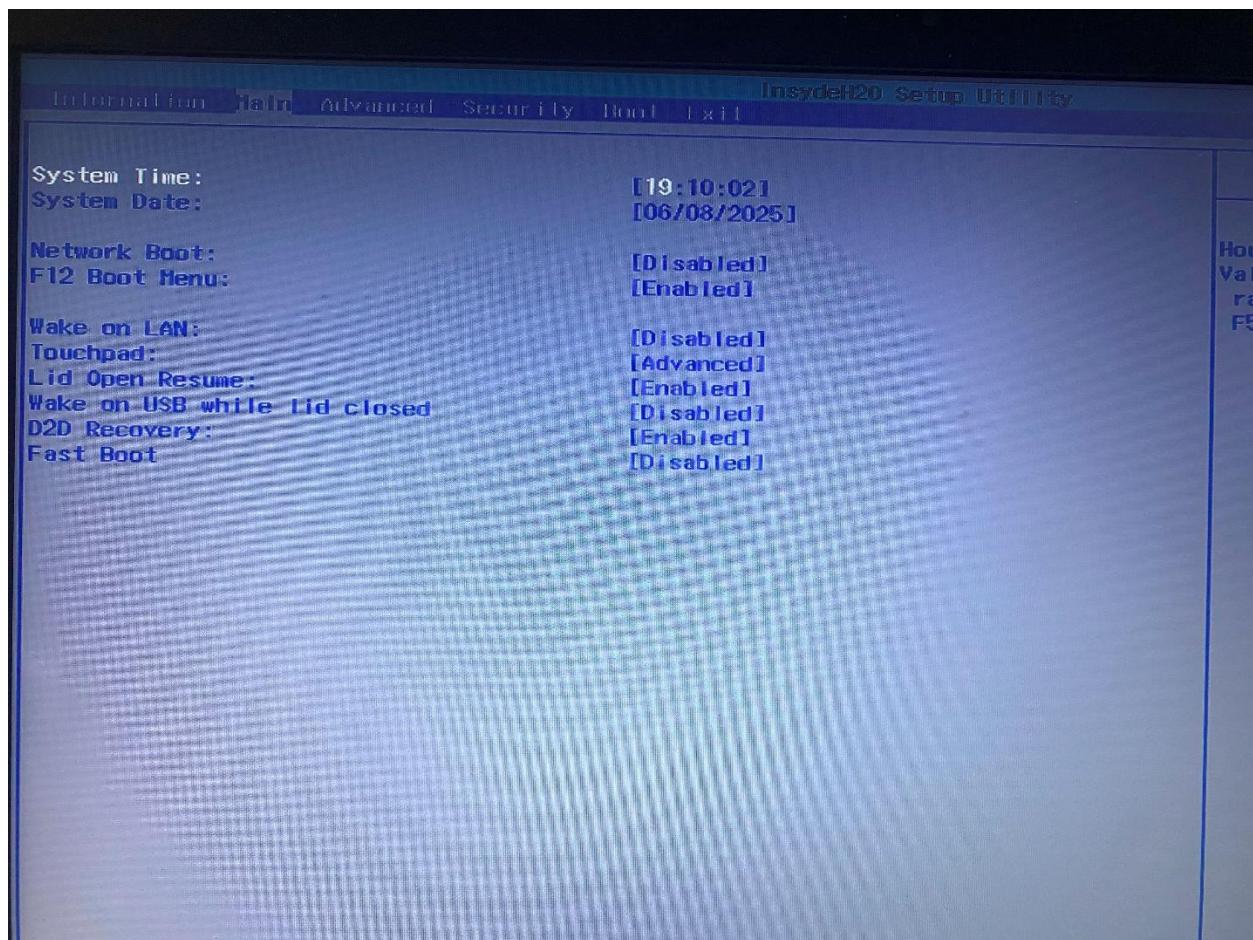
`sudo rm -rf /mnt/root`

`sudo rm -rf /mnt/temp`

مراحل بوت

-20

تنظیمات boot قابل مشاهده است:



InsydeH2O Setup Utility

Information Main Advanced Security Boot Exit

Intel VTX:

[Enabled]

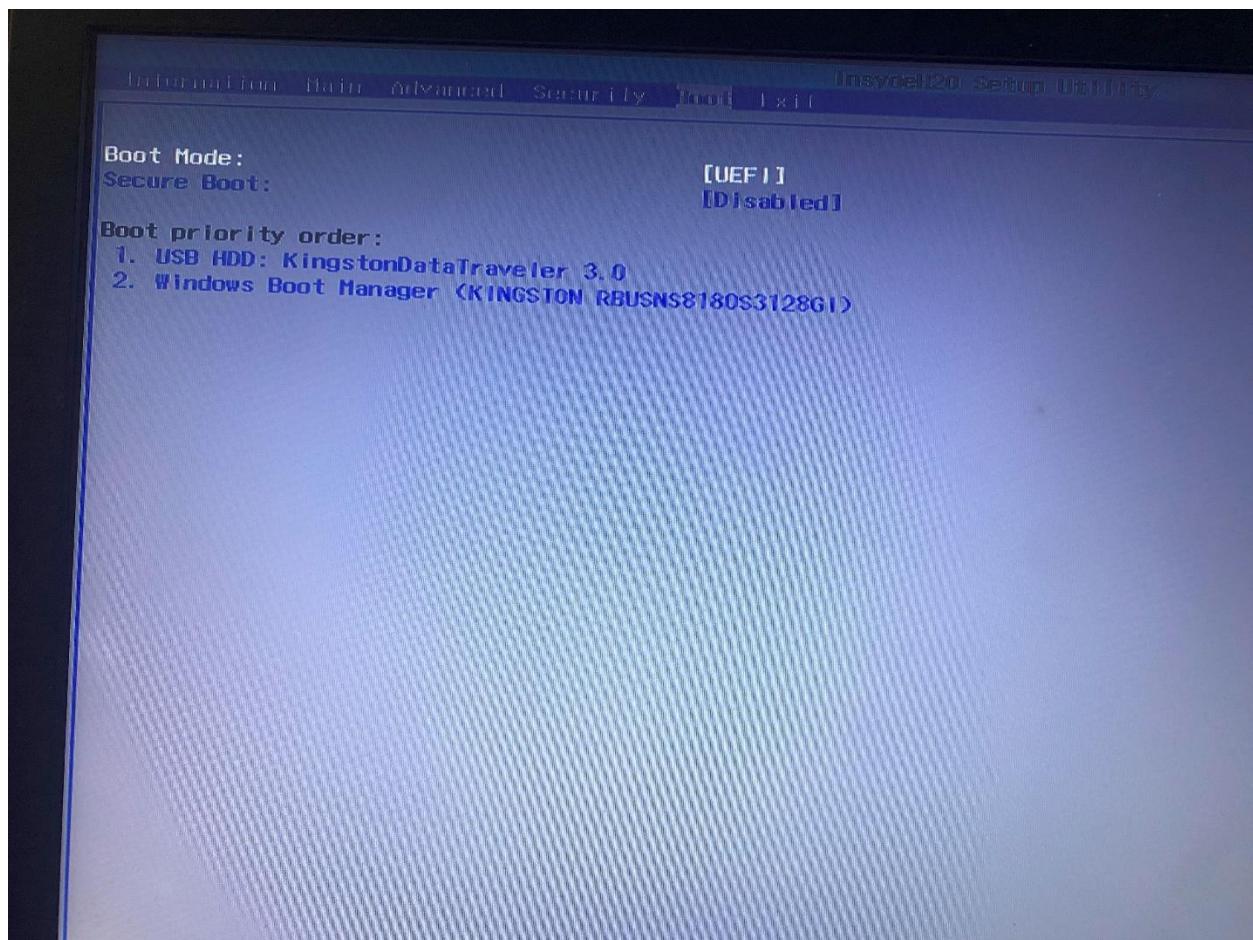
Intel VTD:

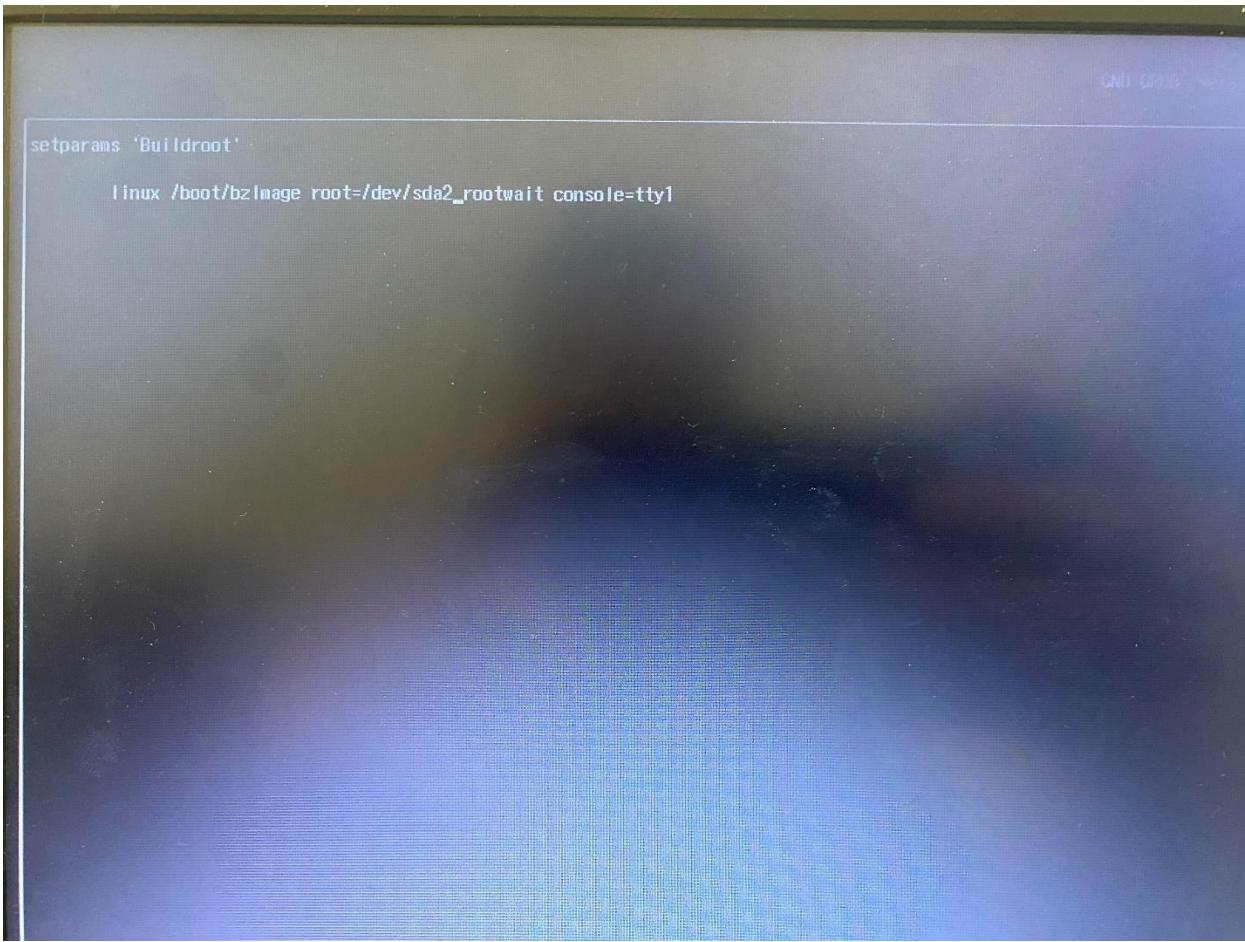
[Enabled]

SATA Configuration

Information Main Advanced Security Boot Exit

Supervisor Password Is:	Set
User Password Is:	Clear
HDD0 Password Is:	Frozen
HDD1 Password Is:	Frozen
Set Supervisor Password:	[Enter]
Set User Password:	[Enter]
Set HDD0 Password:	[Enter]
Set HDD1 Password:	[Enter]
Password on Boot:	[Disabled]
Secure Boot Mode:	Standard
Erase all Secure Boot Setting:	[Enter]
Select an UEFI file as trusted for executing:	[Enter]
Restore Secure Boot to Factory Default:	[Enter]
Current TPM (TCM) State:	Installed
Change TPM (TCM) State:	[Enabled]
Clear TPM (TCM):	[Clear]





هنگام بوت شدن دکمه E را مرتبا میزنیم و در این قسمت sda2 را به sda1 تبدیل کردیم سپس با زدن F10 آن را سیو و بوت به درستی انجام شد.

```
scsi host0: usb-storage 1-5:1.0
usb 1-?: new full-speed USB device number 3 using xhci_hcd
usb 1-9: new high-speed USB device number 4 using xhci_hcd
usb 1-10: new full-speed USB device number 5 using xhci_hcd
scsi 0:0:0:0: Direct-Access      Kingston DataTraveler 3.0 0000 PQ: 0 ANSI: 4
sd 0:0:0:0: [sda] 60538881 512-byte logical blocks: (31.0 GB/28.9 GiB)
sd 0:0:0:0: [sda] Write Protect is off
sd 0:0:0:0: [sda] Write cache: disabled, read cache: enabled, doesn't support DPO or FUA
  sda: sda1 sda2
sd 0:0:0:0: [sda] Attached SCSI removable disk
EXT4-fs (sda2): INFO: recovery required on readonly filesystem
EXT4-fs (sda2): write access will be enabled during recovery
EXT4-fs (sda2): recovery complete
EXT4-fs (sda2): mounted filesystem with ordered data mode. Quota mode: disabled.
UFS: Mounted root (ext4 filesystem) readonly on device 8:2.
devtmpfs: mounted
Freeing unused kernel image (initmem) memory: 1220K
Write protecting the kernel read-only data: 18432k
Freeing unused kernel image (text/rodata gap) memory: 2044K
Freeing unused kernel image (rodata/data gap) memory: 1632K
Run /sbin/init as init process
EXT4-fs (sda2): re-mounted. Quota mode: disabled.
Seeding 256 bits and crediting
Saving 256 bits of creditable seed for next boot
Starting syslogd: OK
Starting klogd: OK
Running sysctl: OK
Starting network: OK
Starting crond: OK
Starting dropbear sshd: OK

Welcome to erfan's oslab
erfan-farahmand-oslab login: _
```

```
autoconfig for Generic: line_outs=1 (0x14/0x0/0x0/0x0/0x0) type:special
snd_hda_codec_generic hdaudioC000: speaker_outs=0 (0x0/0x0/0x0/0x0/0x0)
snd_hda_codec_generic hdaudioC000: hp_outs=1 (0x21/0x0/0x0/0x0/0x0)
snd_hda_codec_generic hdaudioC000: mono: mono_out=0x0
snd_hda_codec_generic hdaudioC000: inputs:
snd_hda_codec_generic hdaudioC000: Mic=0x12
Input: HDA Intel PCH Front Headphone at /devices/pci0000:00/0000:00:1f.3/sound/card0/input8
usb 1-6: new high-speed USB device number 2 using xhci_hcd
usb-storage 1-6:1.0: USB Mass Storage device detected
scsi host0: usb-storage 1-6:1.0
usb 1-7: new full-speed USB device number 3 using xhci_hcd
usb 1-9: new high-speed USB device number 4 using xhci_hcd
usb 1-10: new full-speed USB device number 5 using xhci_hcd
scsi 0:0:0:0: Direct-Access Kingston DataTraveler 3.0 0000 PQ: 0 ANSI: 4
sd 0:0:0:0: [sda] 60538881 512-byte logical blocks: (31.0 GB/28.9 GiB)
sd 0:0:0:0: [sda] Write Protect is off
sd 0:0:0:0: [sda] Write cache: disabled, read cache: enabled, doesn't support DPO or FUA
sda: sda1 sda2
sd 0:0:0:0: [sda] Attached SCSI removable disk
EXT4-fs (sda2): INFO: recovery required on readonly filesystem
EXT4-fs (sda2): write access will be enabled during recovery
EXT4-fs (sda2): recovery complete
EXT4-fs (sda2): mounted filesystem with ordered data mode. Quota mode: disabled.
UFS: Mounted root (ext4 filesystem) readonly on device 8:2.
devtmpfs: mounted
Freeing unused kernel image (initmem) memory: 1220K
Write protecting the kernel read-only data: 18432k
Freeing unused kernel image (text/rodata gap) memory: 2044K
Freeing unused kernel image (rodata/data gap) memory: 1632K
Run /sbin/init as init process
EXT4-fs (sda2): re-mounted. Quota mode: disabled.
Seeding 256 bits and crediting
Saving 256 bits of creditable seed for next boot
Starting syslogd: OK
Starting klogd: OK
Running sysctl: OK
Starting network: OK
Starting crond: OK
Starting dropbear sshd: OK

Welcome to erfan's oslab
erfan-farhamand-oslab login: root
Password:
#
```

```

Usage: ssh [OPTIONS] [user@]host[:port] [command ...]

  -o <option> Set option in OpenSSH-like format ('-o help' to list options)
  -i <identityfile> (multiple allowed, default '~/.ssh/id_dropbear')
  -A   Enable agent auth forwarding
  -L <[listenaddress:]listenport:remotehost:remoteport> Local port forwarding
  -g   Allow remote hosts to connect to forwarded ports
  -R <[listenaddress:]listenport:remotehost:remoteport> Remote port forwarding
  -W <receive_window_buffer> (default 24576, larger may be faster, max 10MB)
  -K <keepalive> (0 is never, default 0)
  -I <idle_timeout> (0 is never, default 0)
  -z   disable QoS
  -B <endhost:endport> Netcat-alike forwarding
  -J <proxy_program> Use program pipe rather than TCP connection
  -c <cipher list> Specify preferred ciphers ('-c help' to list options)
  -m <MAC list> Specify preferred MACs for packet verification (or '-m help')
  -b   [bind_address][:bind_port]
  -V   Version

ssh: Exited: Remote host needs to provided.

# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inette6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: sit0@NONE: <NOARP> mtu 1480 qdisc noop qlen 1000
    link/sit 0.0.0.0 brd 0.0.0.0
# ip
BusyBox v1.37.0 (2025-06-08 15:09:56 +0330) multi-call binary.

Usage: ip [OPTIONS] address|routet|link|tunne|neigh|rule [ARGS]

OPTIONS := -f[amily] inet|inet6|link | -o[neline]

ip addr add|del IFADDR dev IFACE | show|flush [dev IFACE] [to PREFIX]
ip route list|flush|add|del|change|append|replace|test ROUTE
ip link set IFACE [up|down] [arp on|off] [multicast on|off]
    [promisc on|off] [mtu NUM] [name NAME] [qlen NUM] [address MAC]
    [master IFACE | nomaster] [netns PID] [type TYPE ARGS]
ip tunnel add|change|del|show [NAME]
    [mode ipip|gre|sit] [remote ADDR] [local ADDR] [ttl TTL]
ip neigh show|flush [to PREFIX] [dev DEV] [nud STATE]
ip rule [list] | add|del SELECTOR ACTION
# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inette6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: sit0@NONE: <NOARP> mtu 1480 qdisc noop qlen 1000
    link/sit 0.0.0.0 brd 0.0.0.0
#

```

FULL
HD

1080

مشکلاتی که من در انجام این پروژه به آنها برخوردم شامل موارد زیر است:

- مشکل حافظه سیستم عامل (در نهایت حافظه ماشین مجازی خودم رو بیشتر کردم چون با 20 گیگ در make به مشکل میخورد)
- مشکلات پارتبیشن بندی و تکرار آنها به دفعات
- ساخтар پروژه کمی متفاوت با ساخтар پروژه ای که در دستور کار گفته شده بود، ایجاد شد در حالی که متوجه شدم درست است.
- بوت به نادرستی انجام میشد که متوجه شدم هنگام بوت شدن باید این کار را انجام داد.

لینک گوگل درایو (شامل فایل‌های ایمیج):

https://drive.google.com/file/d/1p_flyhS0f5XHOZOUpFqMHkGsLCOV8BX/view?usp=sharing