Operating System Consepts

Template Final Exam

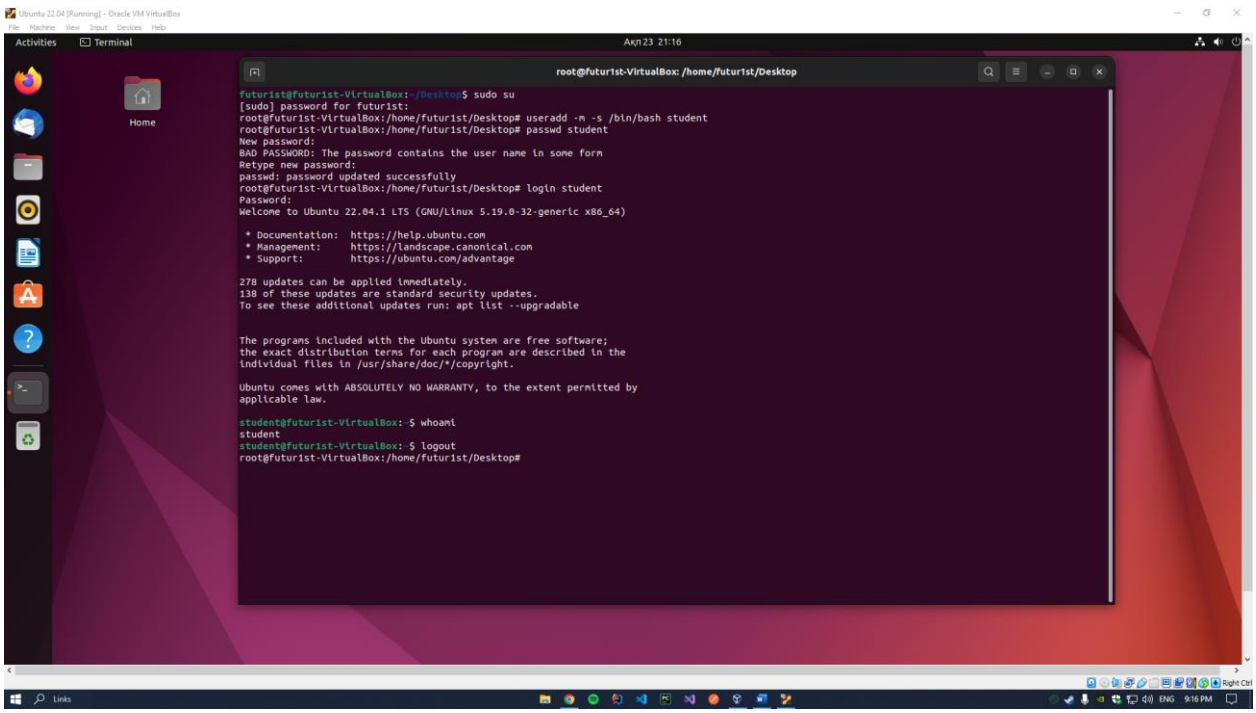
#	Full Name	Group
1	Myrzakhanov Abylaykhan	CS-2127N
2	Tuleshev Turan	CS-2127N
3	Turan Miras	CS-2129N

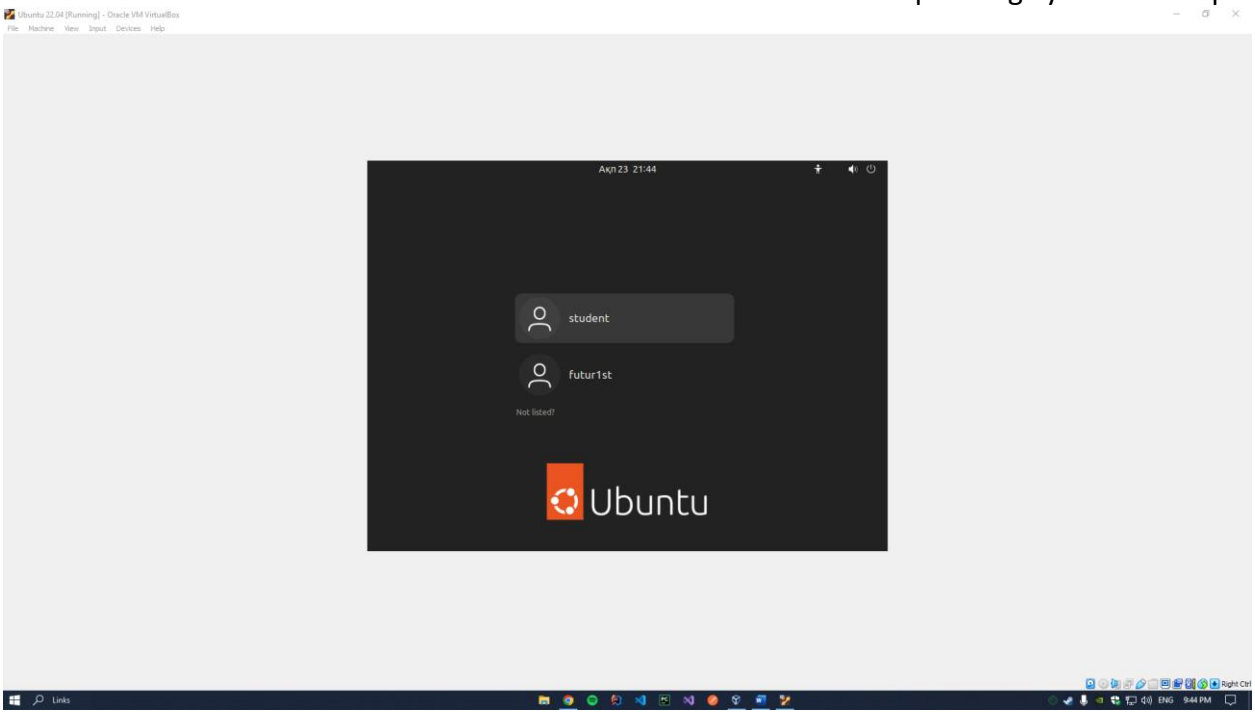
Link to the repository: https://github.com/Futur1stXD/Final_OSC.git

Step-by-step task completion:

Task 1:

1. sudo su
2. useradd -m -s /bin/bash student
3. passwd student
4. login student
5. whoami
6. logout



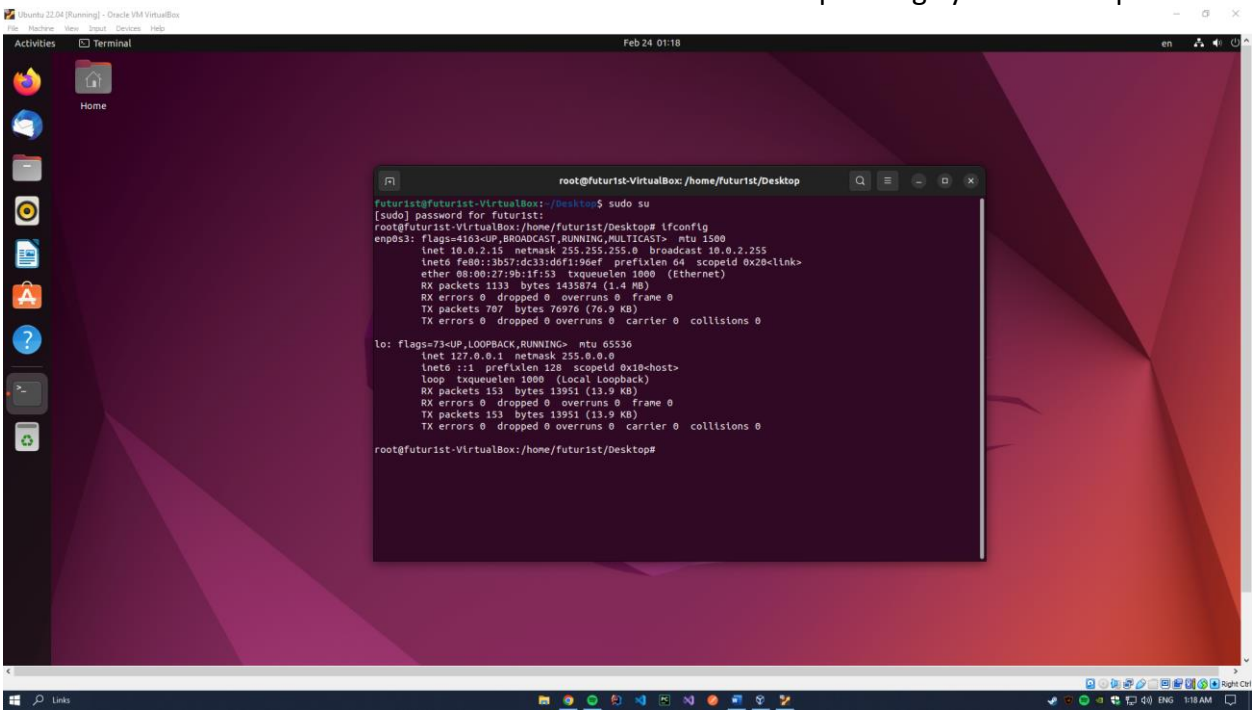


Screenshots of the code compilation result:

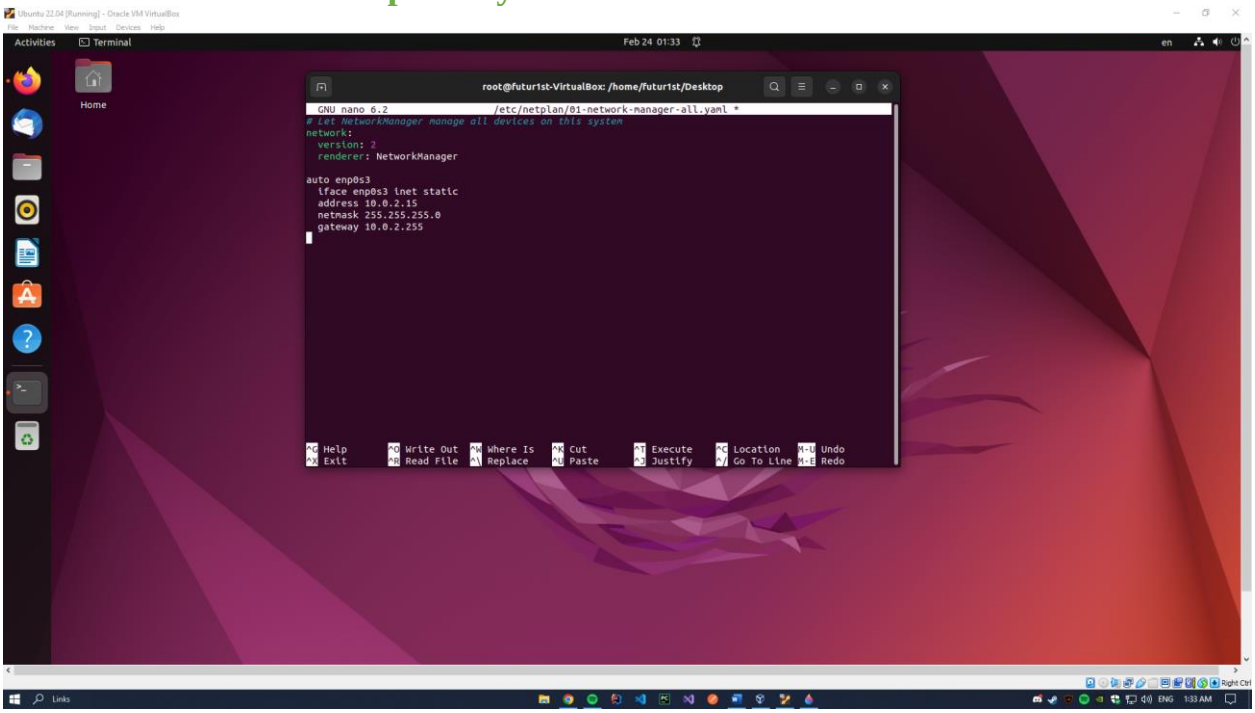
Task 2:

First method: Direct IP connection to the Internet

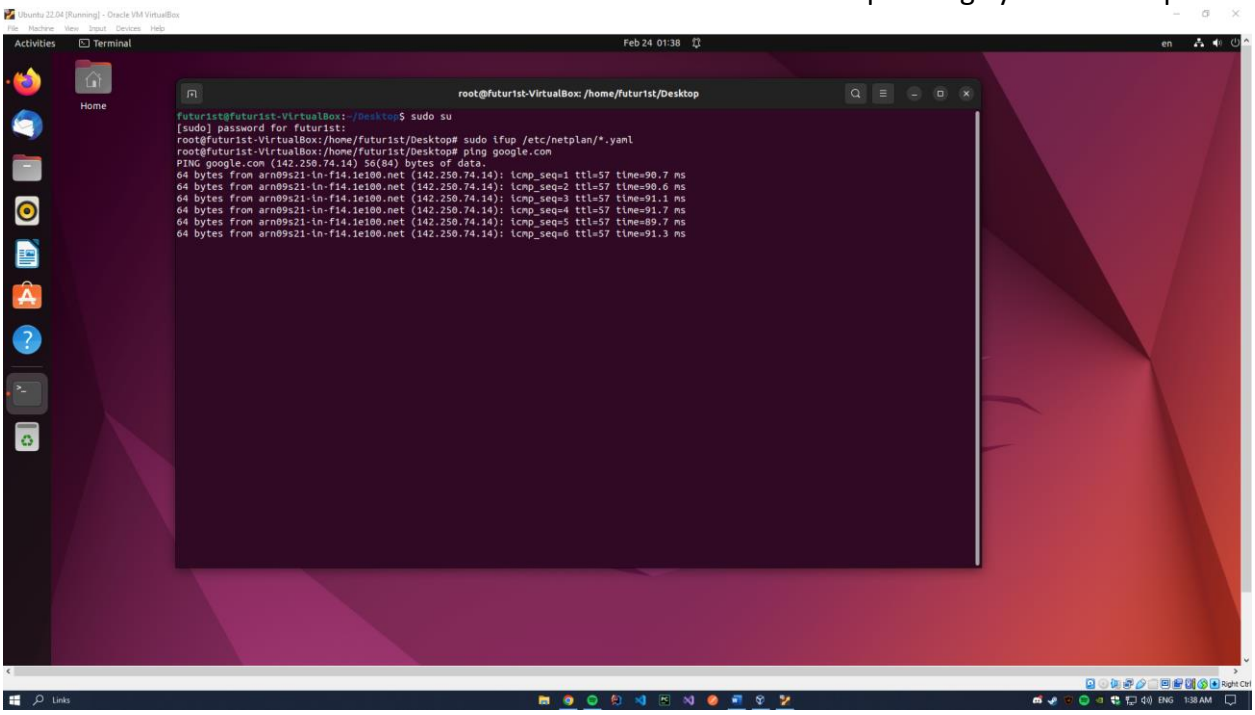
1. sudo su
2. ifconfig



3. **sudo nano /etc/netplan/*.yaml**

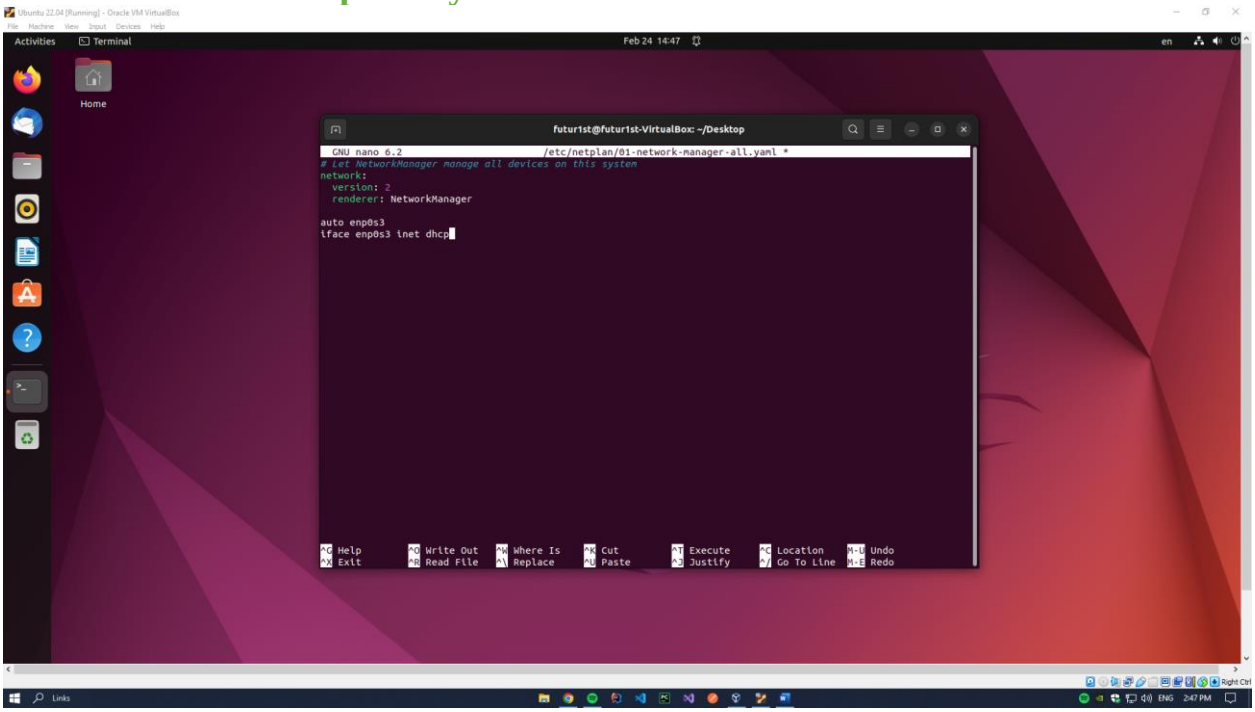


4. **sudo ifup /etc/netplan/*.yaml**
5. **ping google.com**

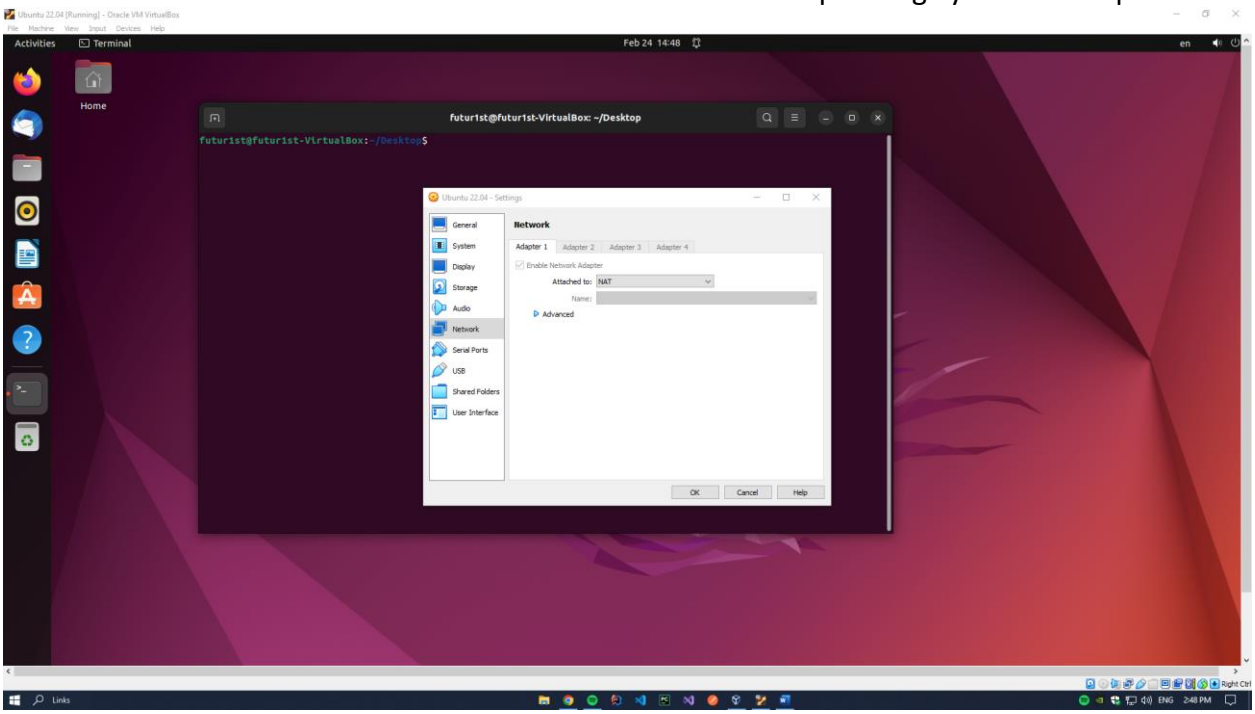


Second method: Connection to NAT

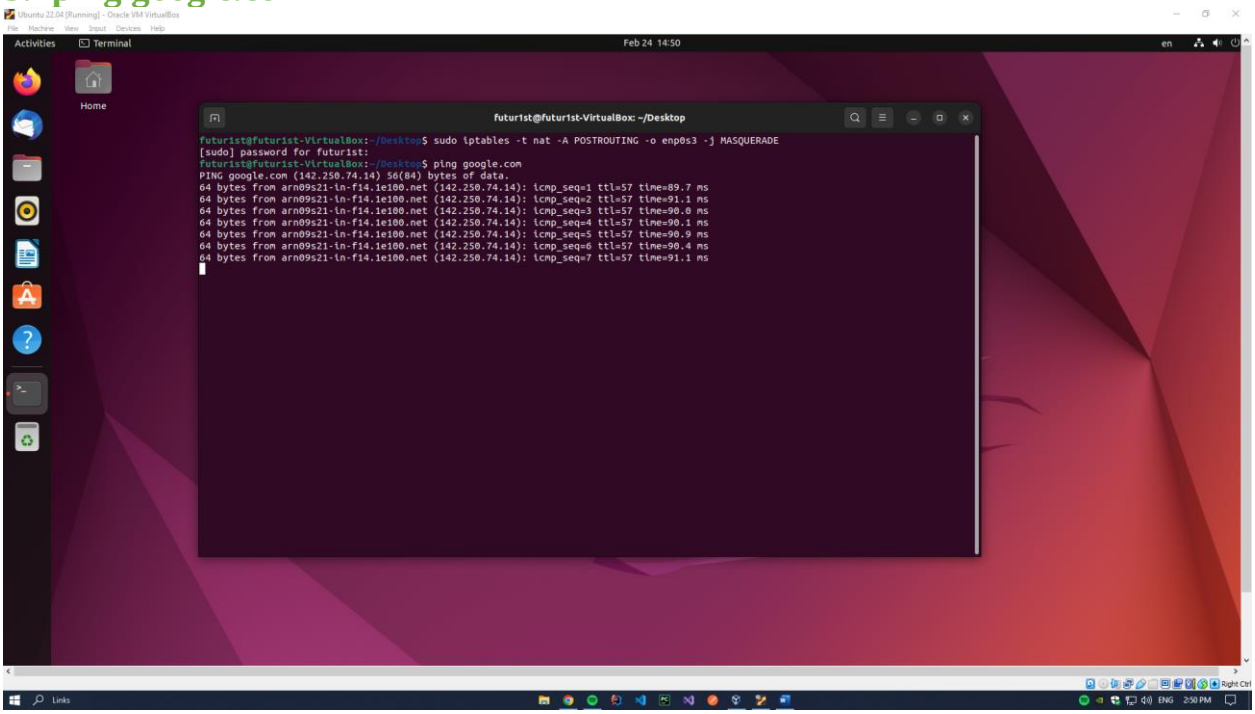
1. ifconfig
2. sudo nano /etc/netplan/*.yaml



3. Change in virtual machine setting connection to NAT

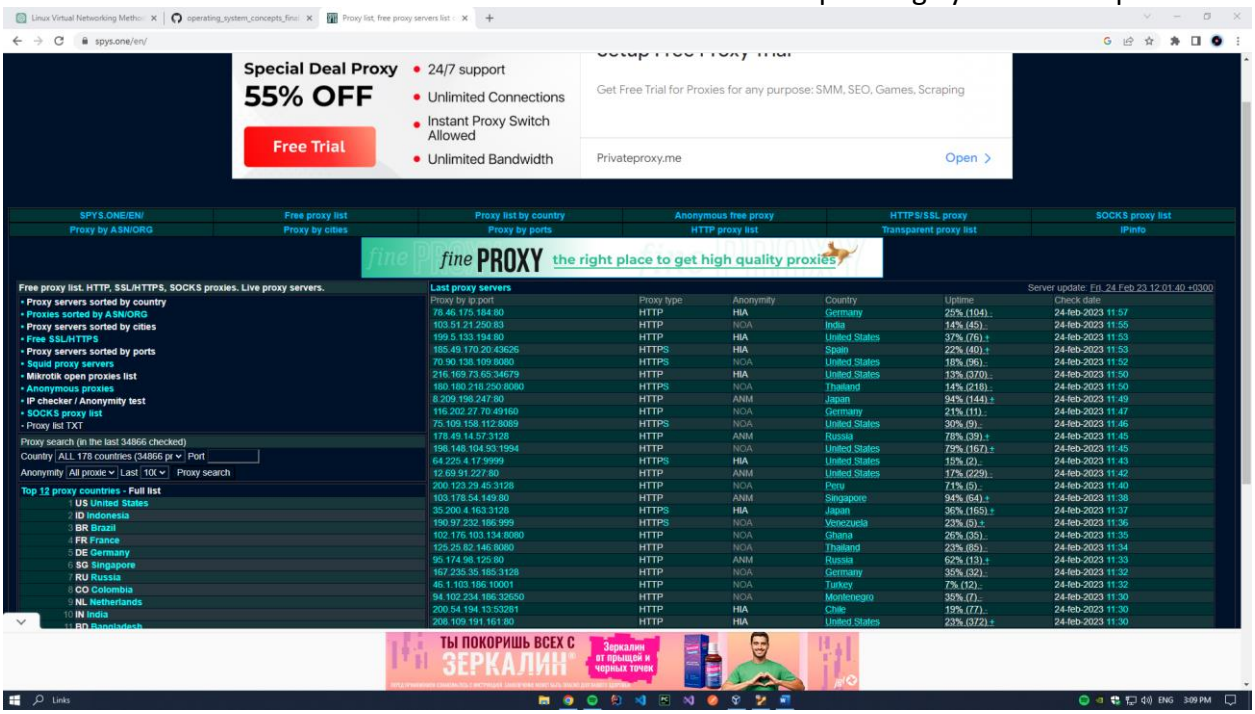


4. **sudo iptables -t nat -A POSTROUTING -o <host_interface> -j MASQUERADE**
5. **ping google.com**

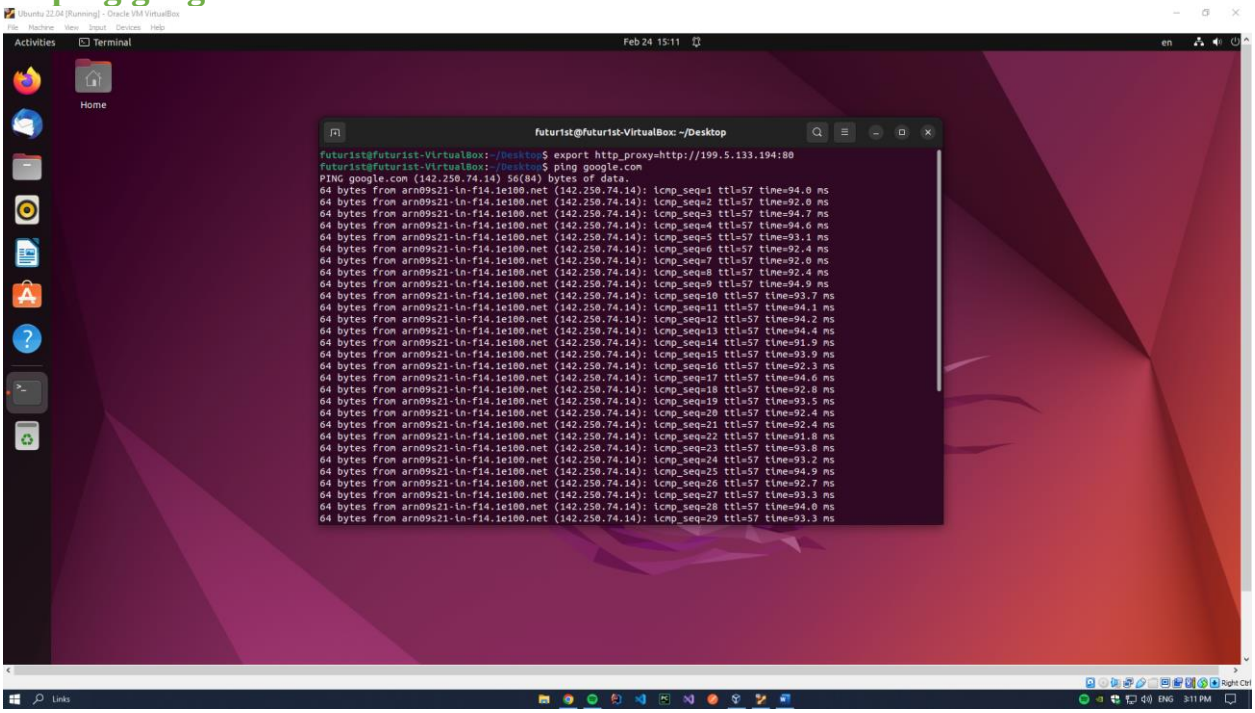


Third method: Proxy

1. **Search in browser: Free proxy servers. In my example I use this site: https://spys.one/en/**



2. **export http_proxy=http:// 199.5.133.194:80**
3. **ping google.com**

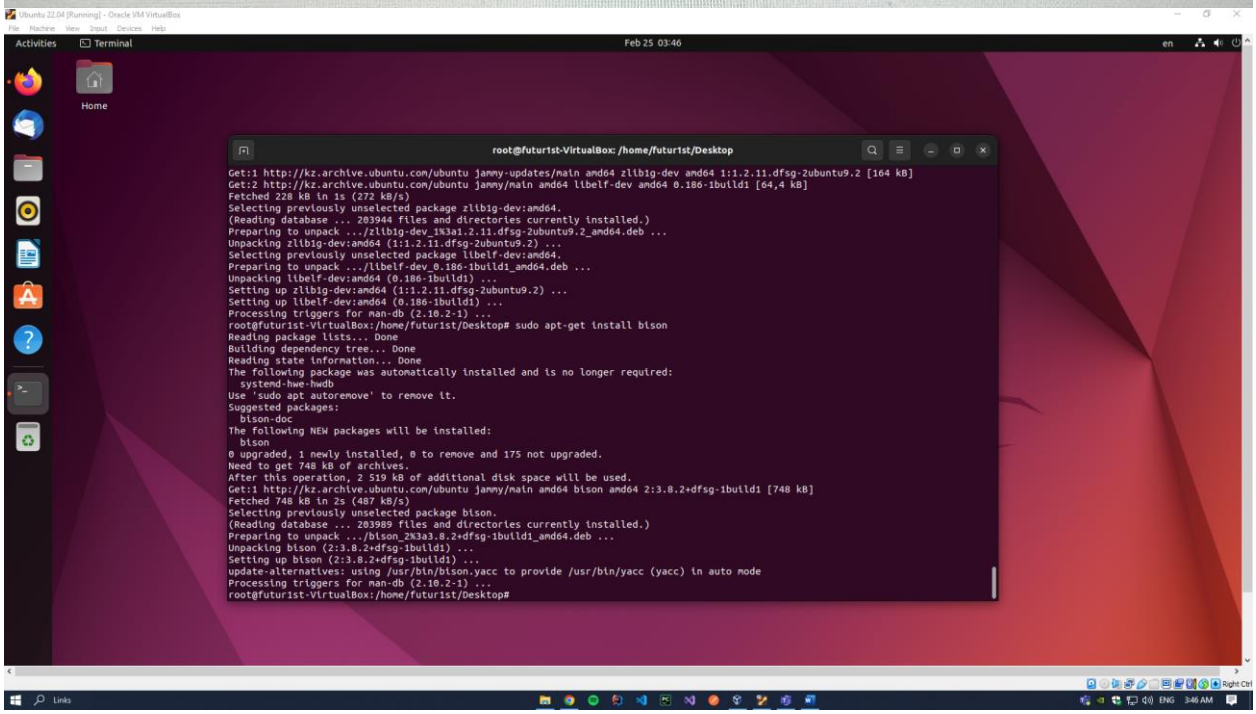


Screenshots of the code compilation result:

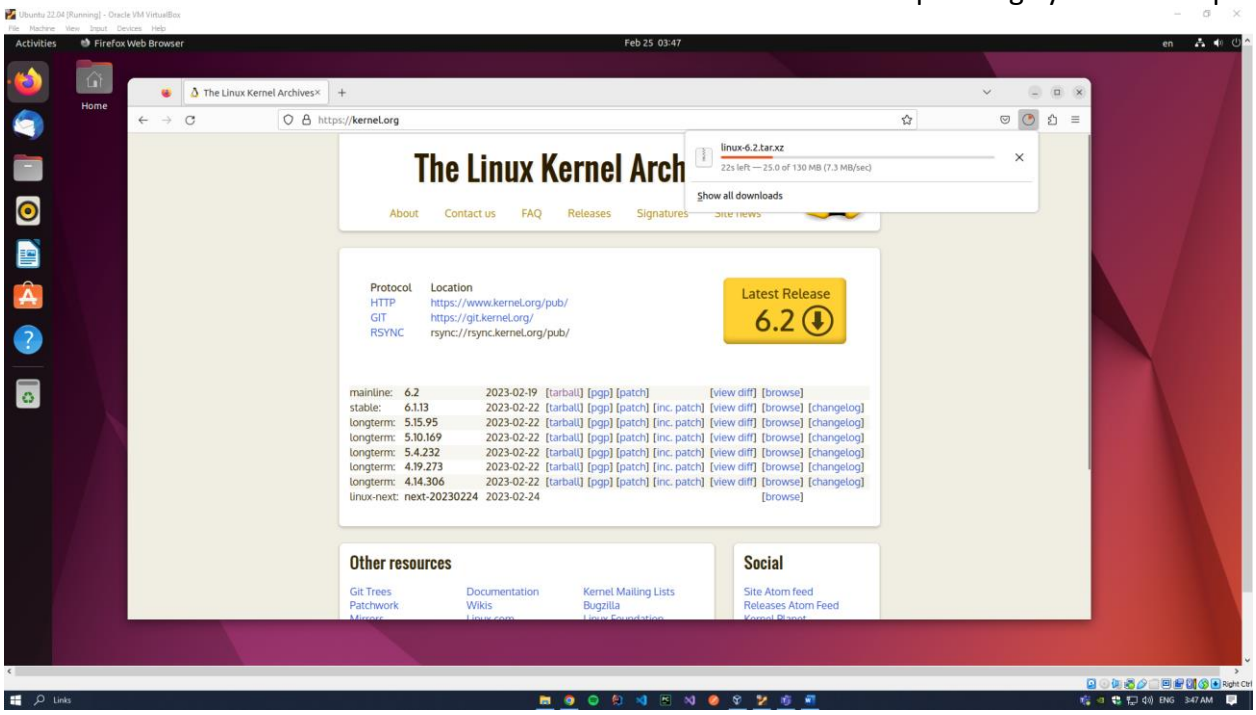
Task 3:

1. At first we need install this packages

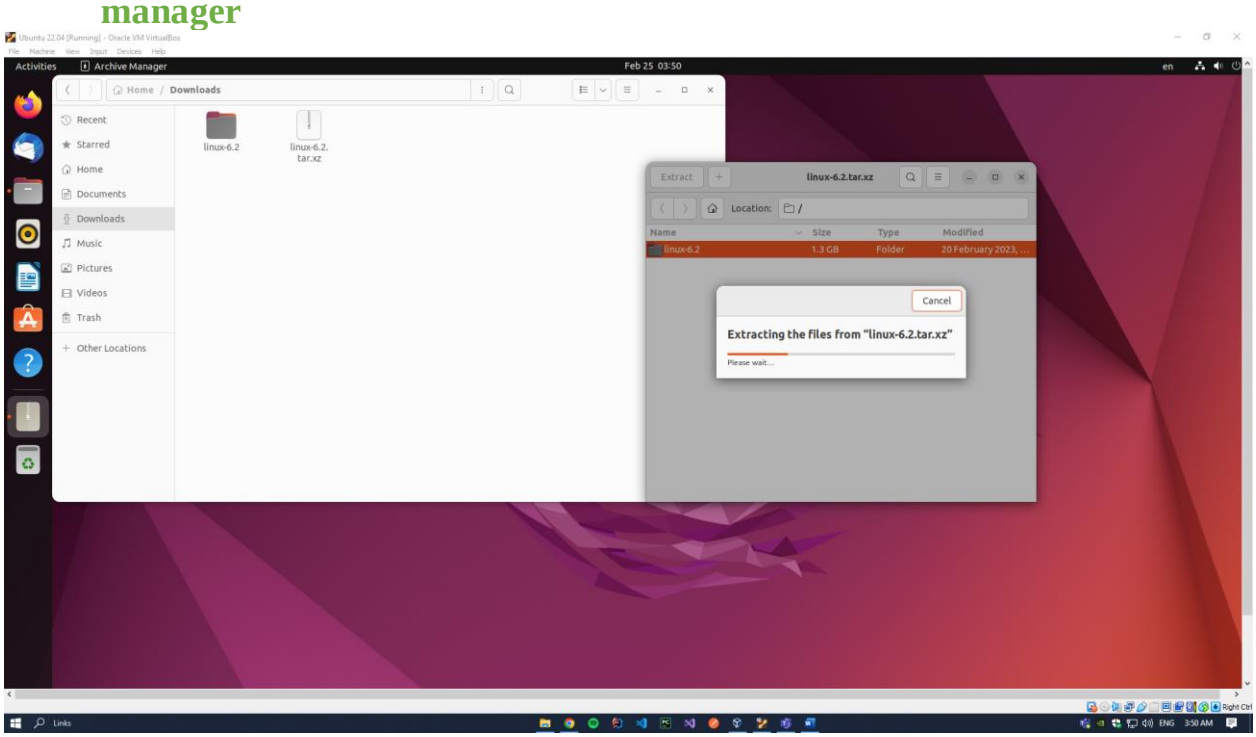
Package	Package description
git	Tracks and makes a record of all changes during development in the source code. It also allows reverting the changes.
fakeroot	Creates the fake root environment.
build-essential	Installs development tools such as C, C++, gcc, and g++.
ncurses-dev	Provides API for the text-based terminals.
xz-utils	Provides fast file compression and decompression.
libssl-dev	Supports SSL and TSL that encrypt data and make the internet connection secure.
bc (Basic Calculator)	Supports the interactive execution of statements.
flex (Fast Lexical Analyzer Generator)	Generates lexical analyzers that convert characters into tokens.
libelf-dev	Issues a shared library for managing ELF files (executable files, core dumps and object code)
bison	Converts grammar description to a C program.



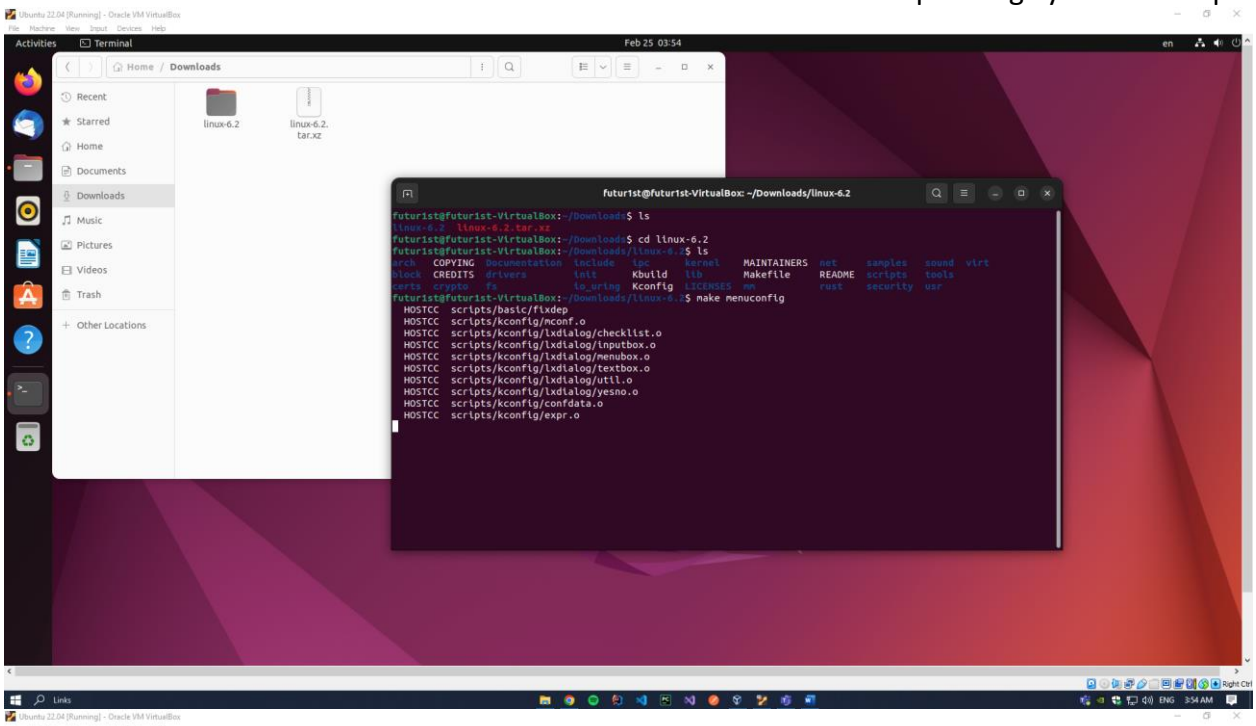
2. After that we download from **kernel.org** latest version of kernel

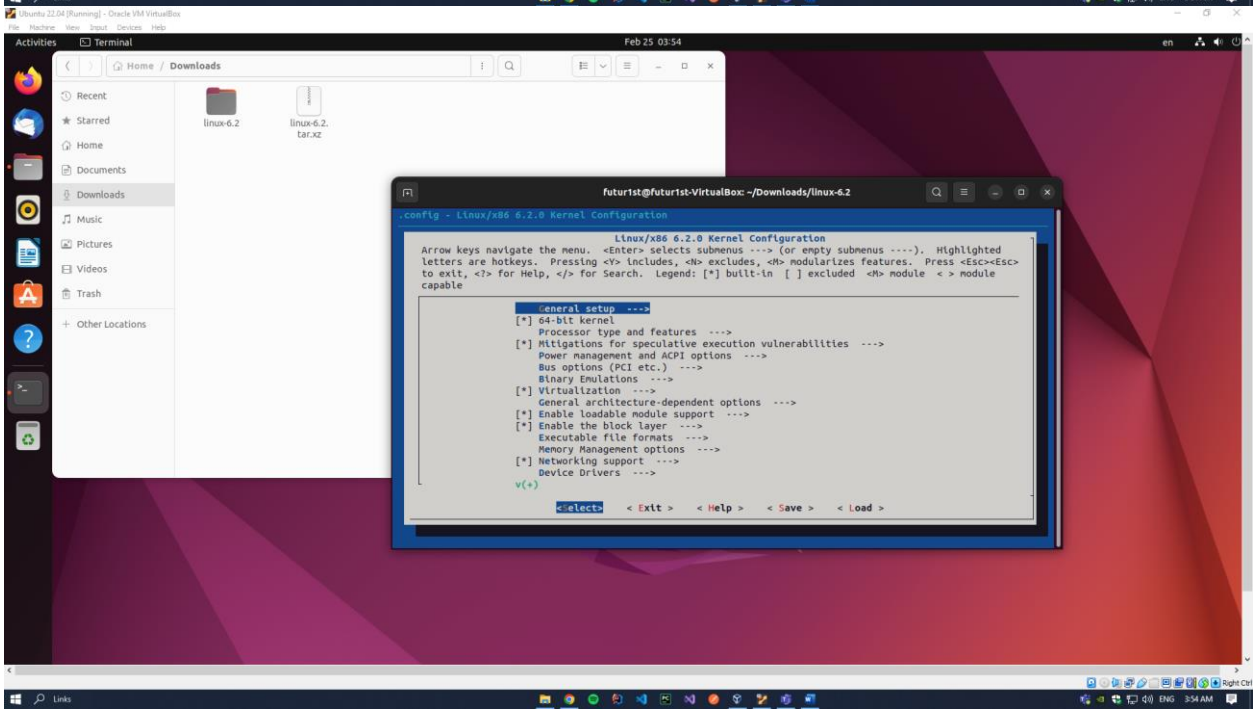


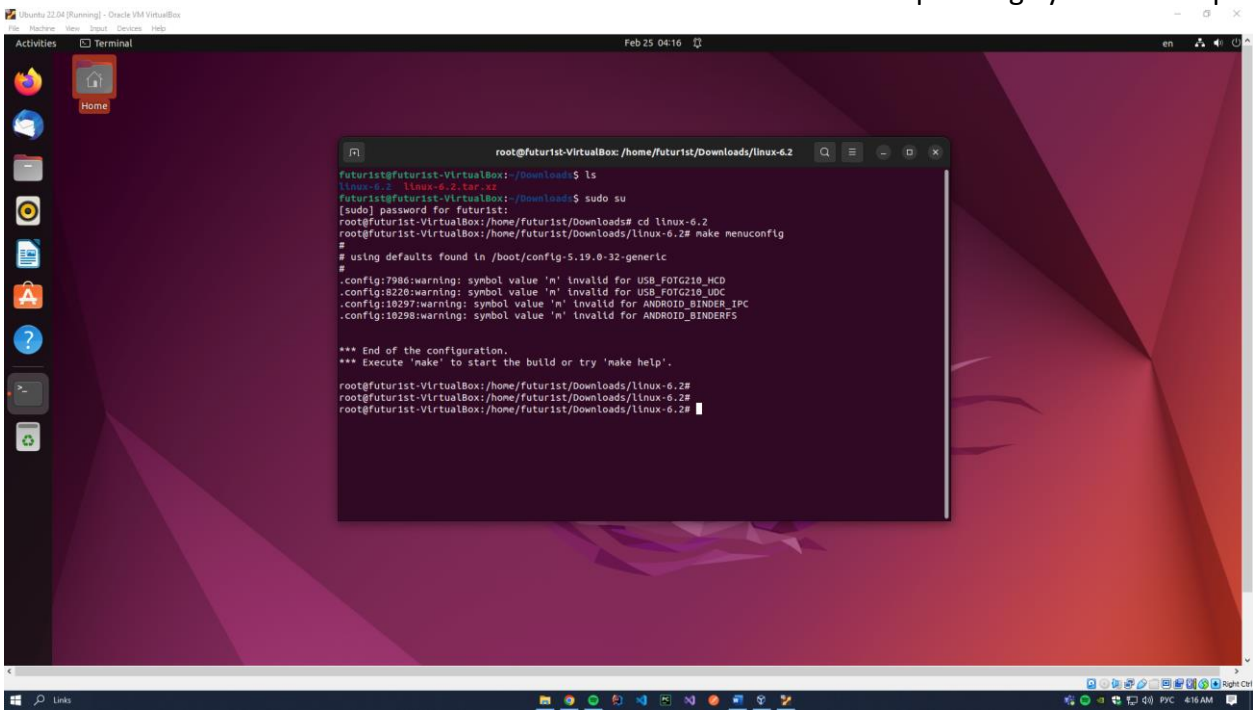
3. **Unzip this file with command: tar xvf linux-6.2.tar.xz or Archive manager**



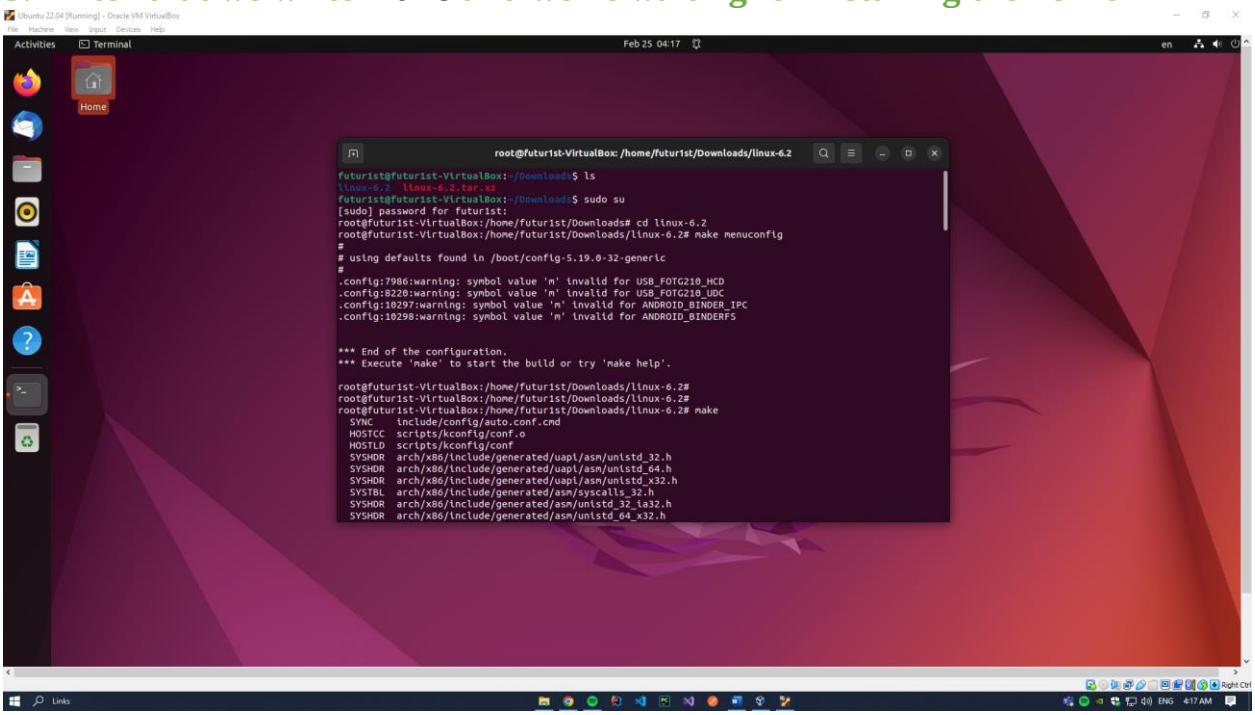
4. **After that we open this file in Terminal and do make menuconfig**

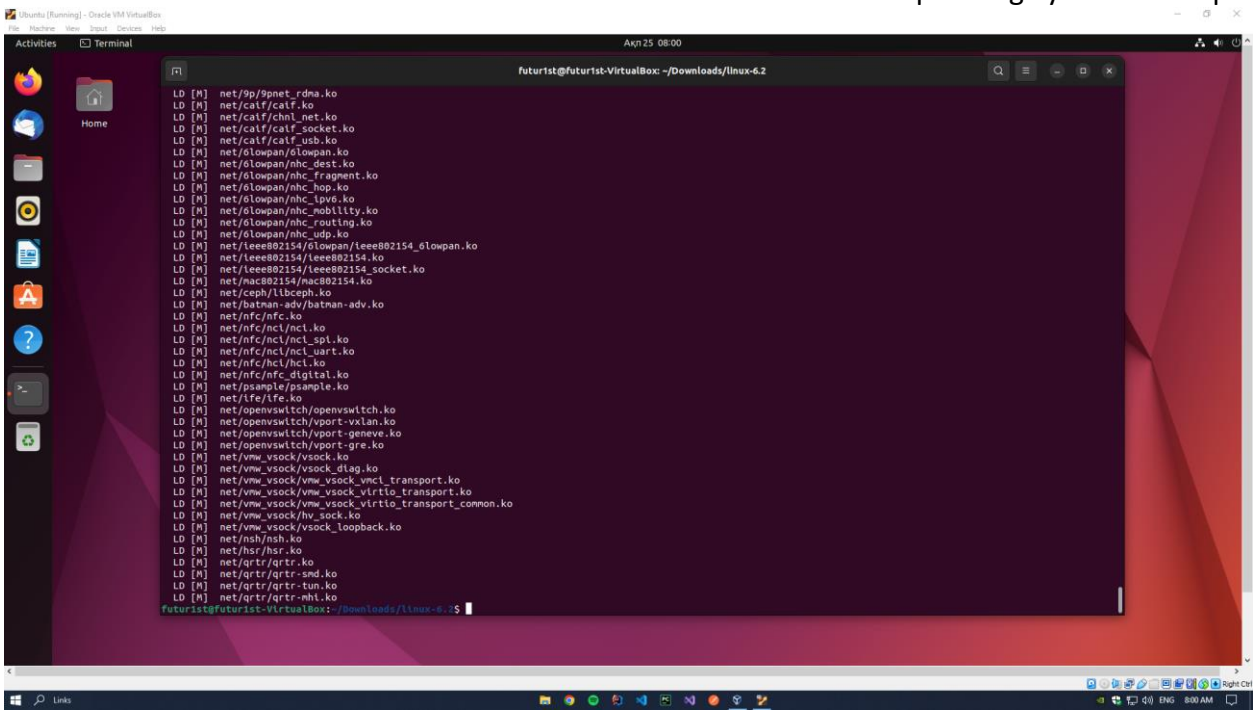

Activities
Terminal
Feb 25 03:54
Home / Downloads
Recent
Starred
Home
Documents
Downloads
Music
Pictures
Videos
Trash
Other Locations
linux-6.2
linux-6.2.tar.xz
futur1st@futur1st-VirtualBox: ~/Downloads/linux-6.2
futur1st@futur1st-VirtualBox:~/Downloads$ ls
linux-6.2 linux-6.2.tar.xz
futur1st@futur1st-VirtualBox:~/Downloads$ cd linux-6.2
futur1st@futur1st-VirtualBox:~/Downloads/linux-6.2$ ls
arch COPYING Documentation include ipc kernel MAINTAINERS net samples sound virt
block CREDITS drivers init Kbuild lib Makefile README scripts tools
certs crypto fs io_uring Kconfig LICENSES mm rust security usr
futur1st@futur1st-VirtualBox:~/Downloads/linux-6.2$ make menuconfig
HOSTCC scripts/basic/fixdep
HOSTCC scripts/kconfig/mconf.o
HOSTCC scripts/kconfig/lxdialog/checklist.o
HOSTCC scripts/kconfig/lxdialog/inputbox.o
HOSTCC scripts/kconfig/lxdialog/menubox.o
HOSTCC scripts/kconfig/lxdialog/textbox.o
HOSTCC scripts/kconfig/lxdialog/util.o
HOSTCC scripts/kconfig/lxdialog/yesno.o
HOSTCC scripts/kconfig/confdata.o
HOSTCC scripts/kconfig/expr.o


Activities
Terminal
Feb 25 03:54
Home / Downloads
Recent
Starred
Home
Documents
Downloads
Music
Pictures
Videos
Trash
Other Locations
linux-6.2
linux-6.2.tar.xz
futur1st@futur1st-VirtualBox: ~/Downloads/linux-6.2
.config - Linux/x86 6.2.0 Kernel Configuration
Linux/x86 6.2.0 Kernel Configuration
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ----). Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded <M> module < > module capable
General setup ---->
[*] 64-bit kernel
Processor type and features ---->
[*] Mitigations for speculative execution vulnerabilities ---->
Power management and ACPI options ---->
Bus options (PCI etc.) ---->
Binary Emulations ---->
[*] Virtualization ---->
General architecture-dependent options ---->
[*] Enable loadable module support ---->
[*] Enable the block layer ---->
Executable file formats ---->
Memory Management options ---->
[*] Networking support ---->
Device Drivers ---->
v(+)
<Select> < Exit > < Help > < Save > < Load >

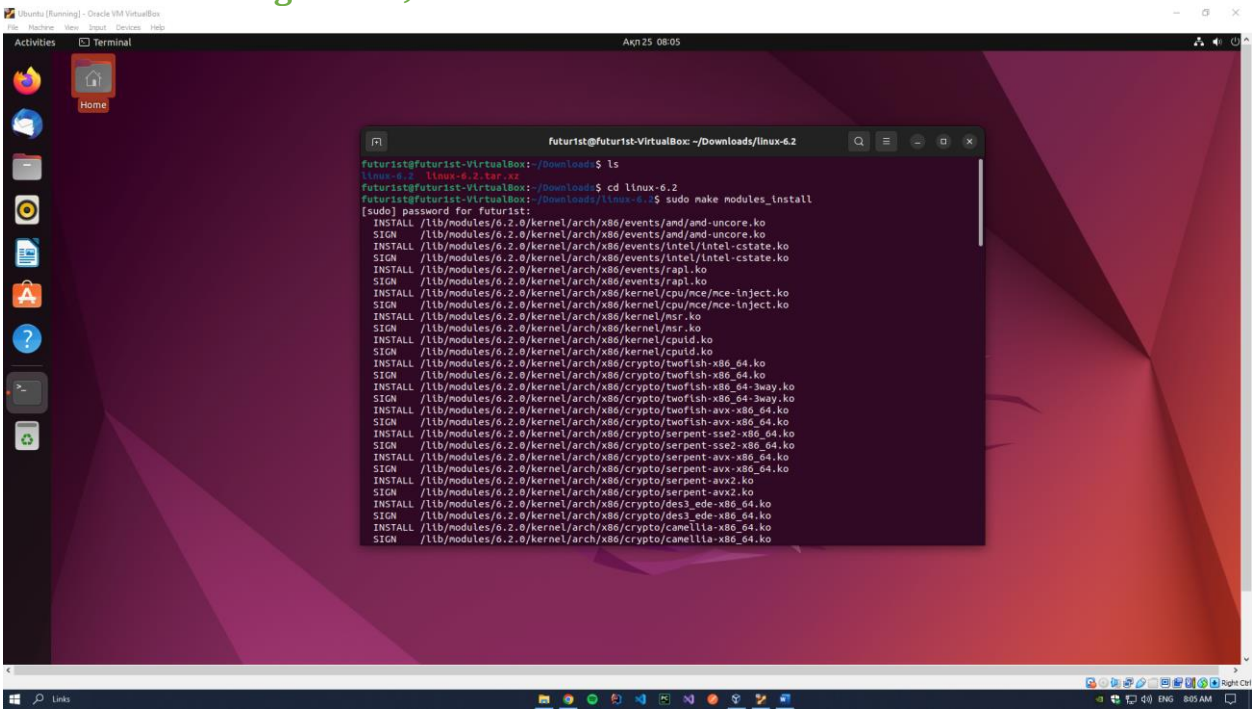


5. After that we write **make** and we`re waiting for installing the kernel

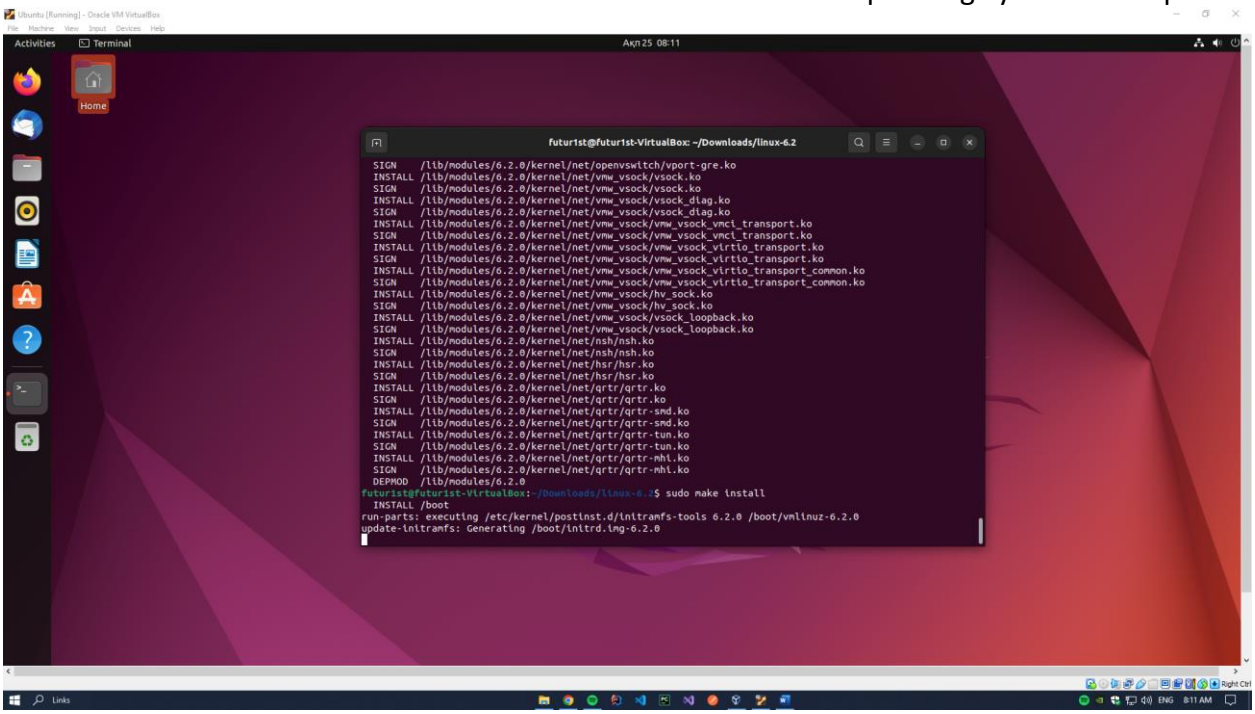




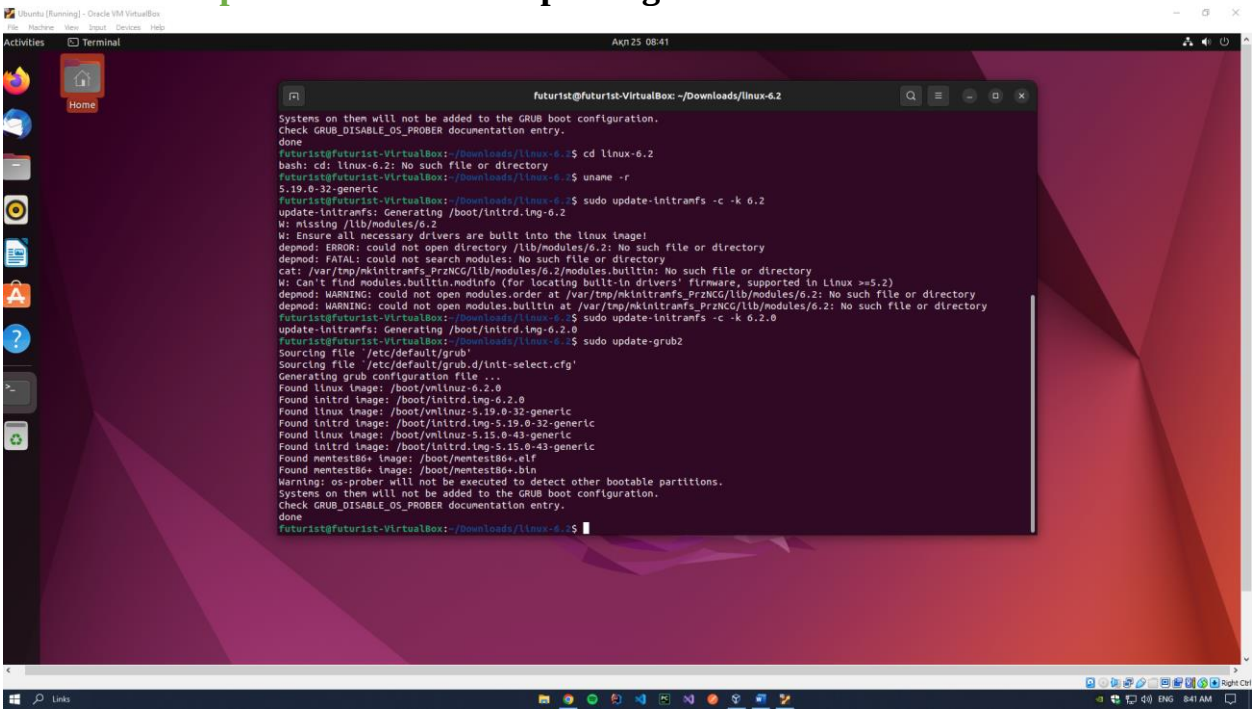
6. After installing kernel, we need install the modules



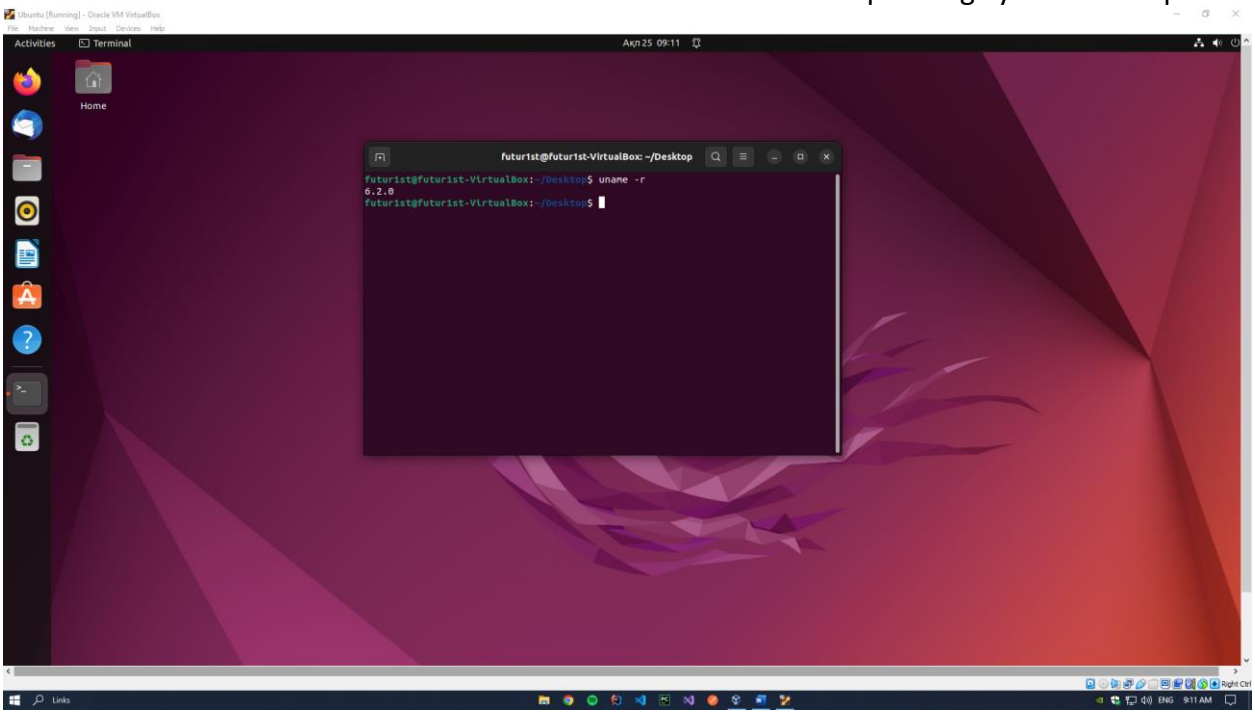
7. After installing modules, we need install new this kernel



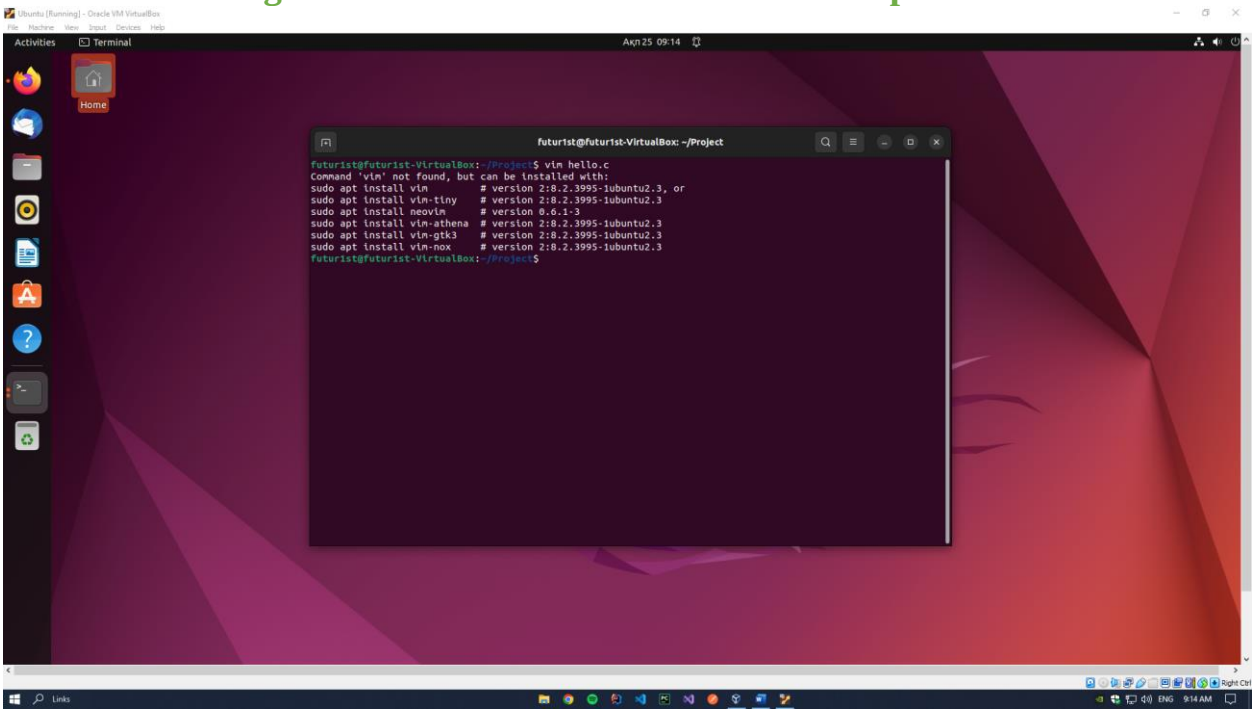
8. We need update GRUB **sudo update-grub2**



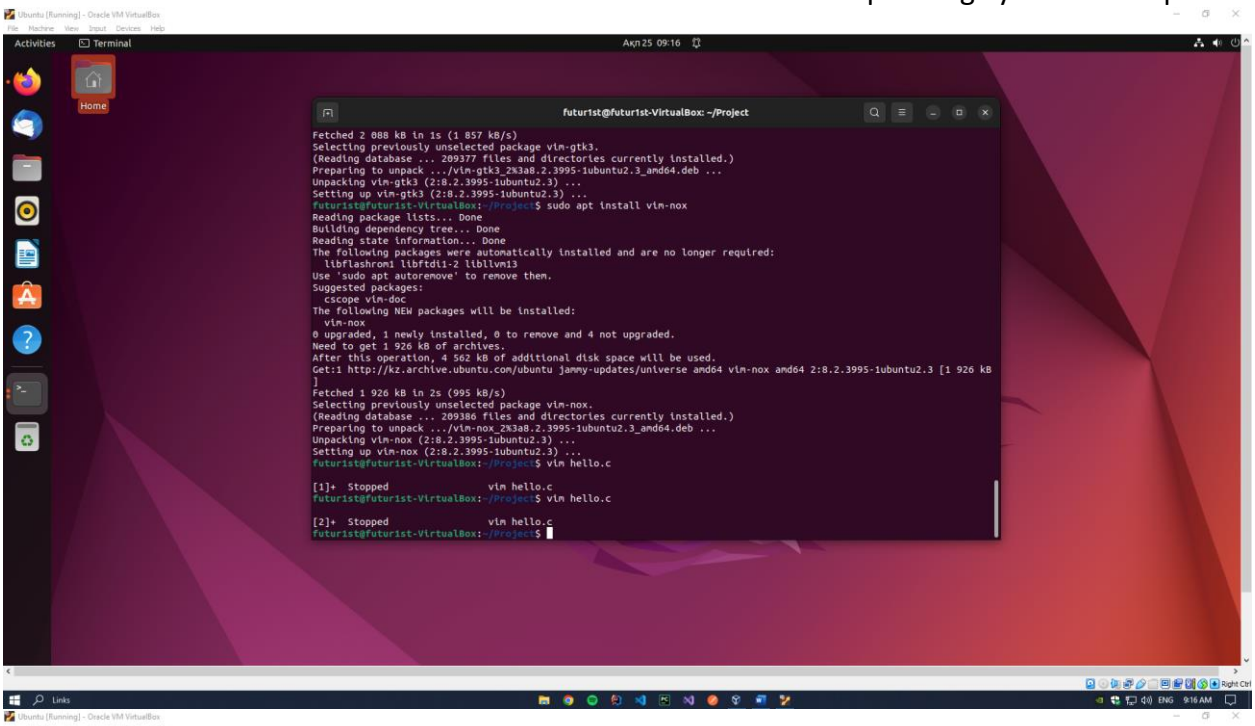
9. After that we restart our Linux and check the version of Kernel **uname -r**

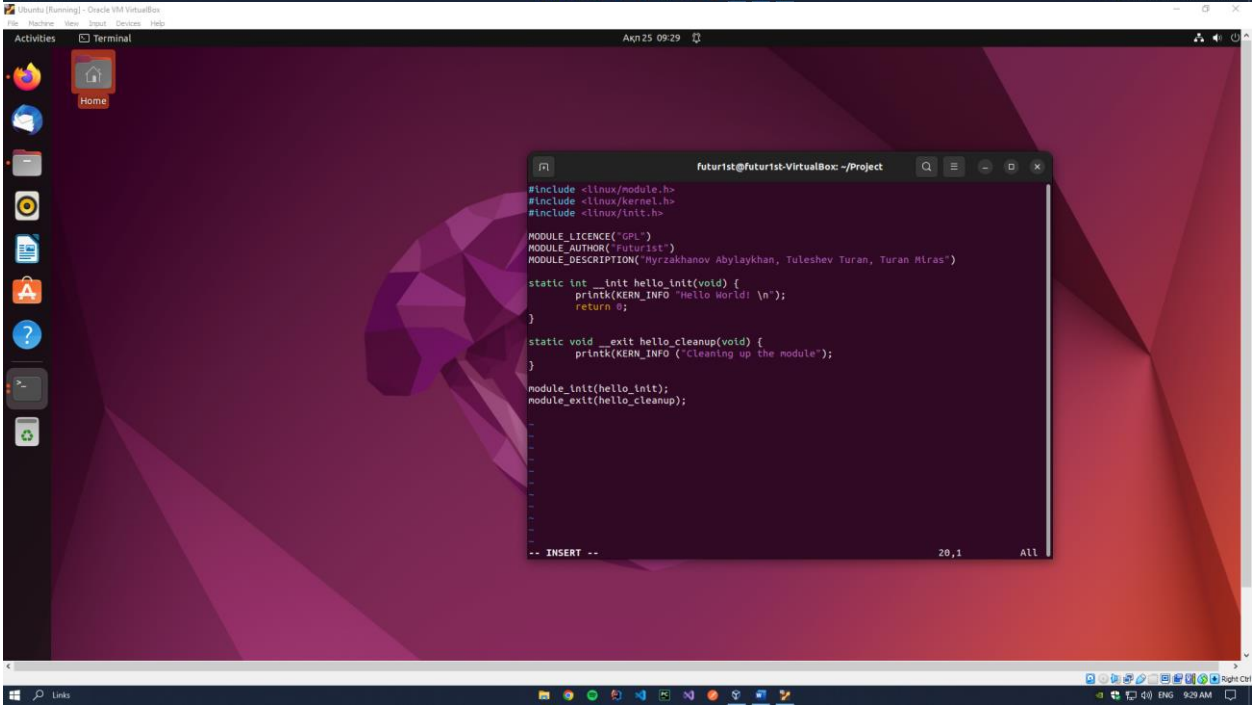


10.Before writing basic Hello world we need install this packets

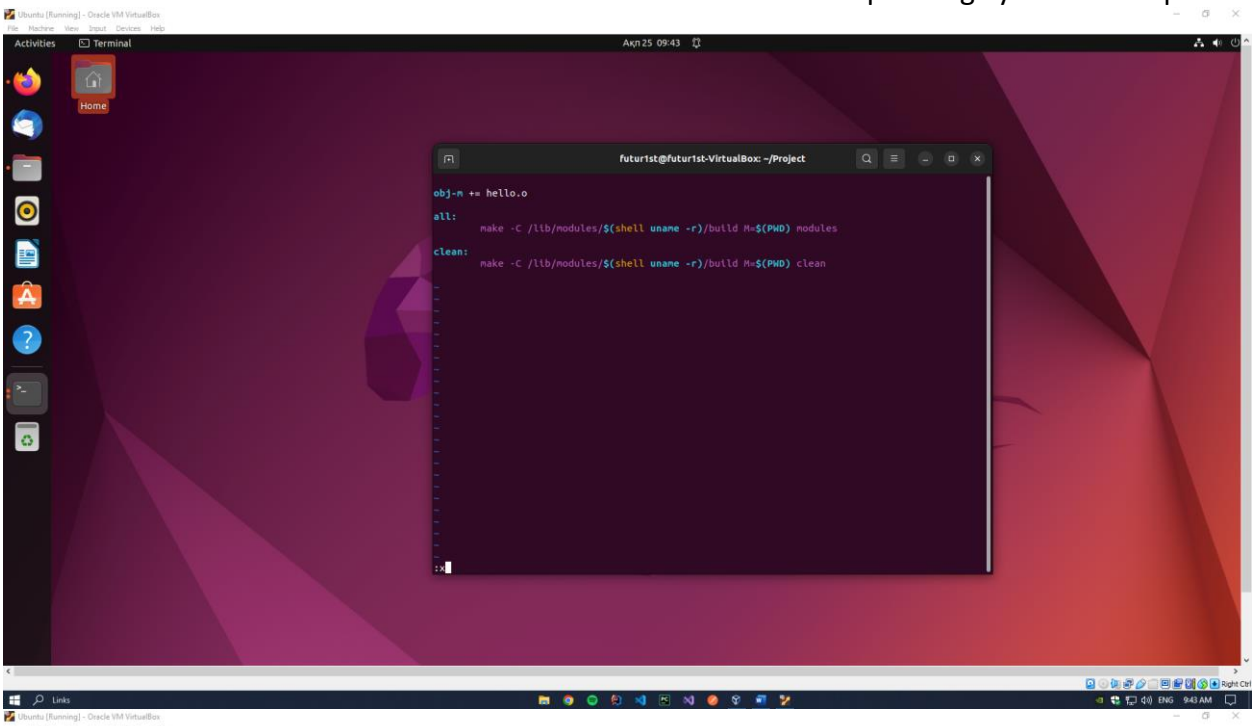


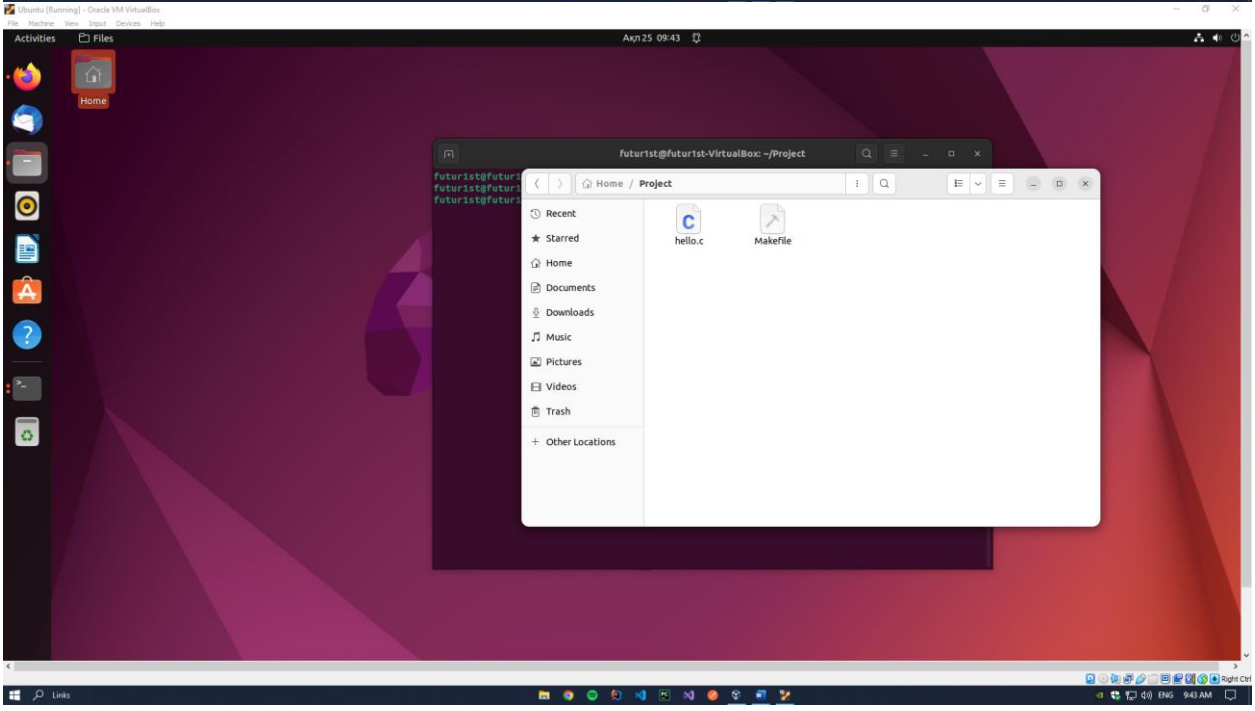
11.After we can start write Hello world.c with command vim hello.c



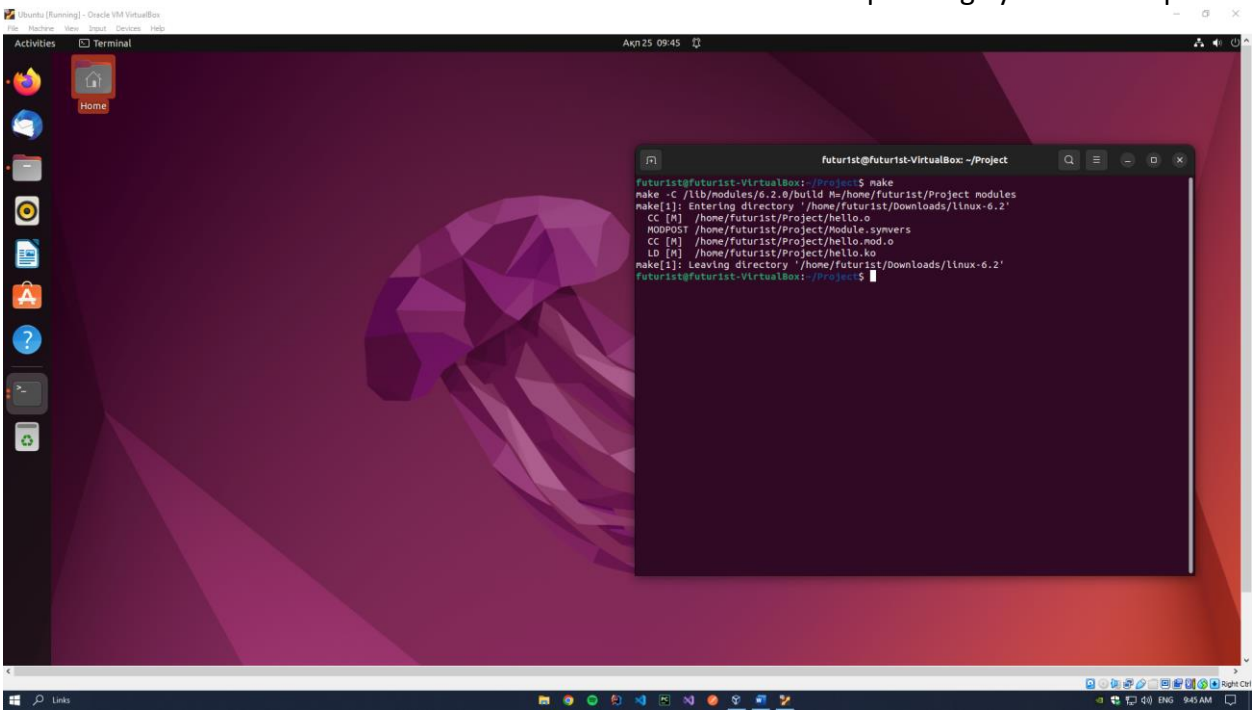


12.After we create file that will compile our code **vim Makefile**

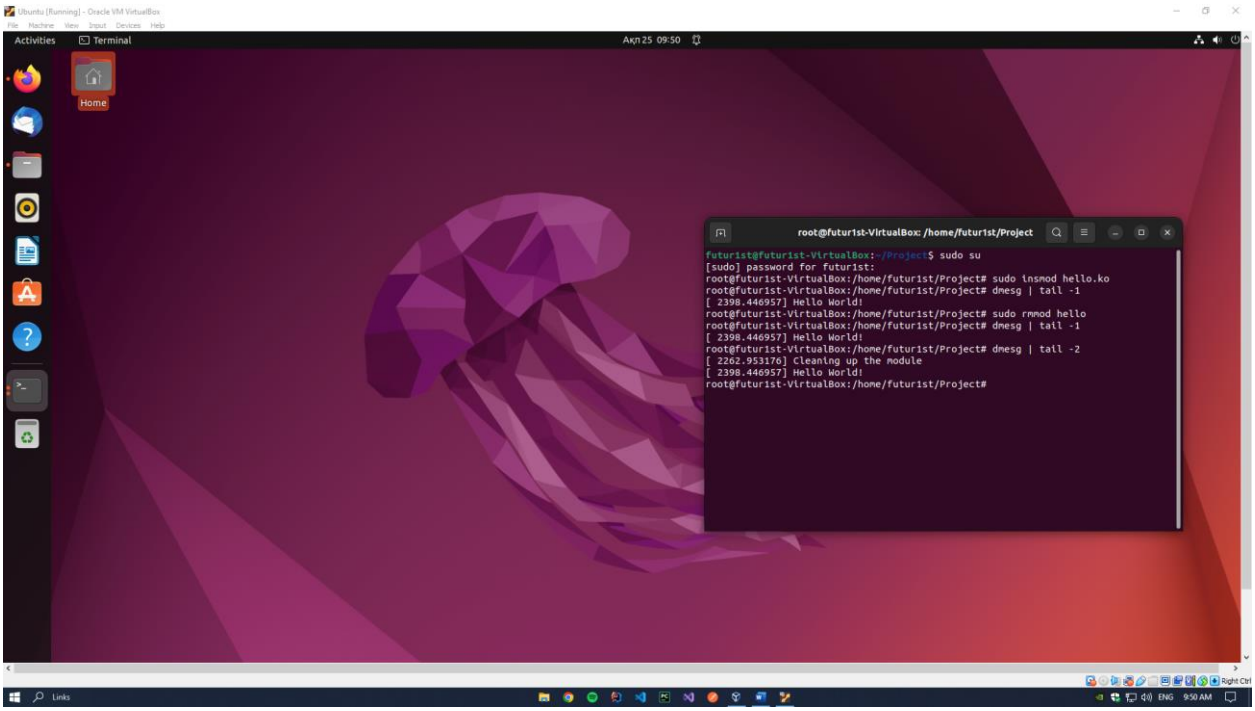




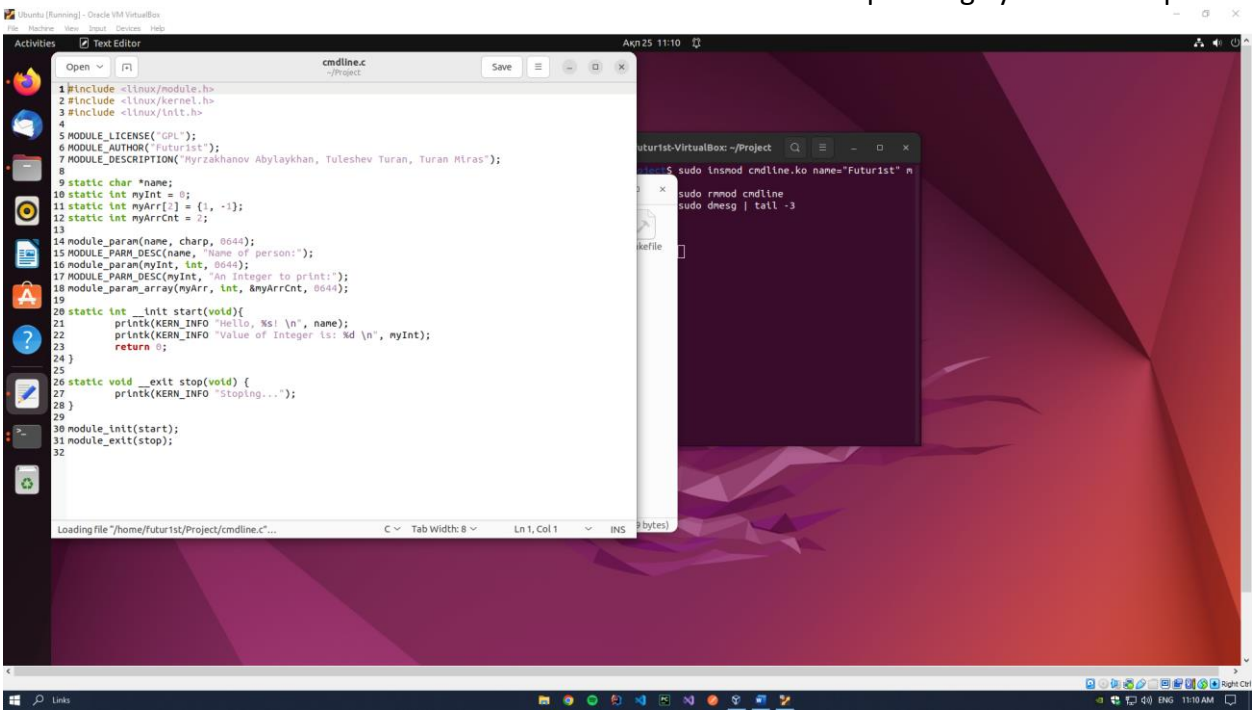
13. Write make for find some errors



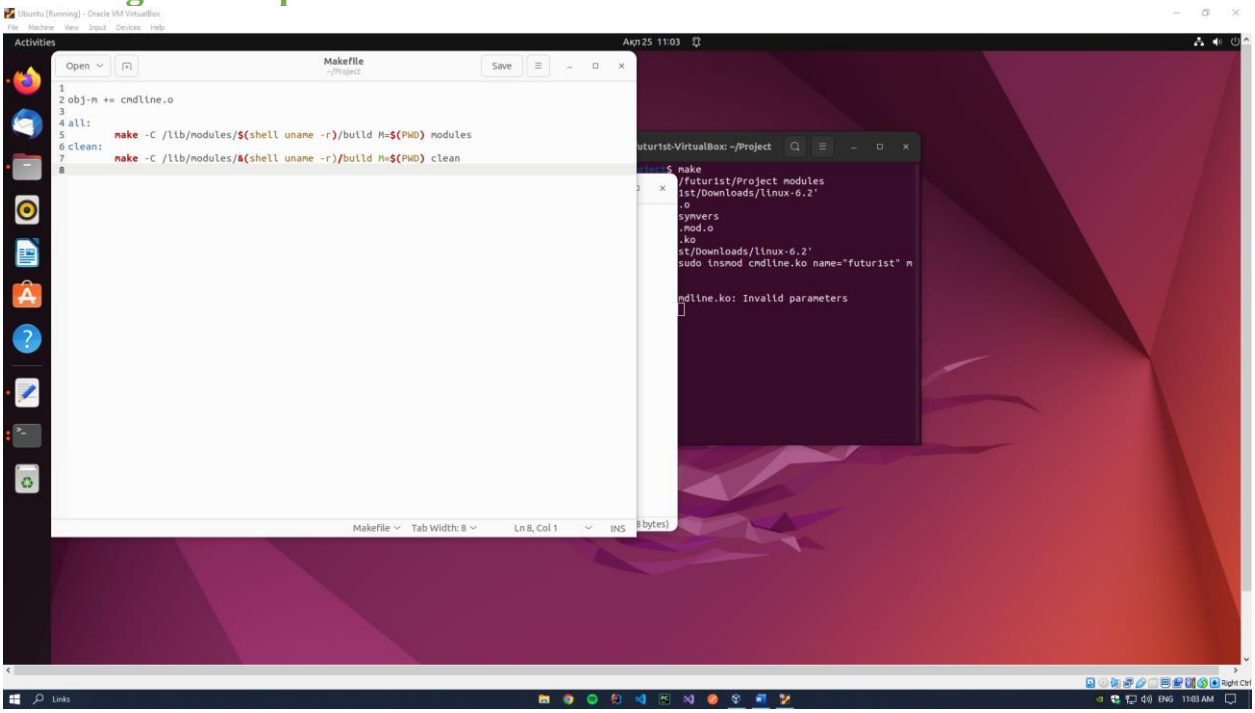
14.After that we turn on our code

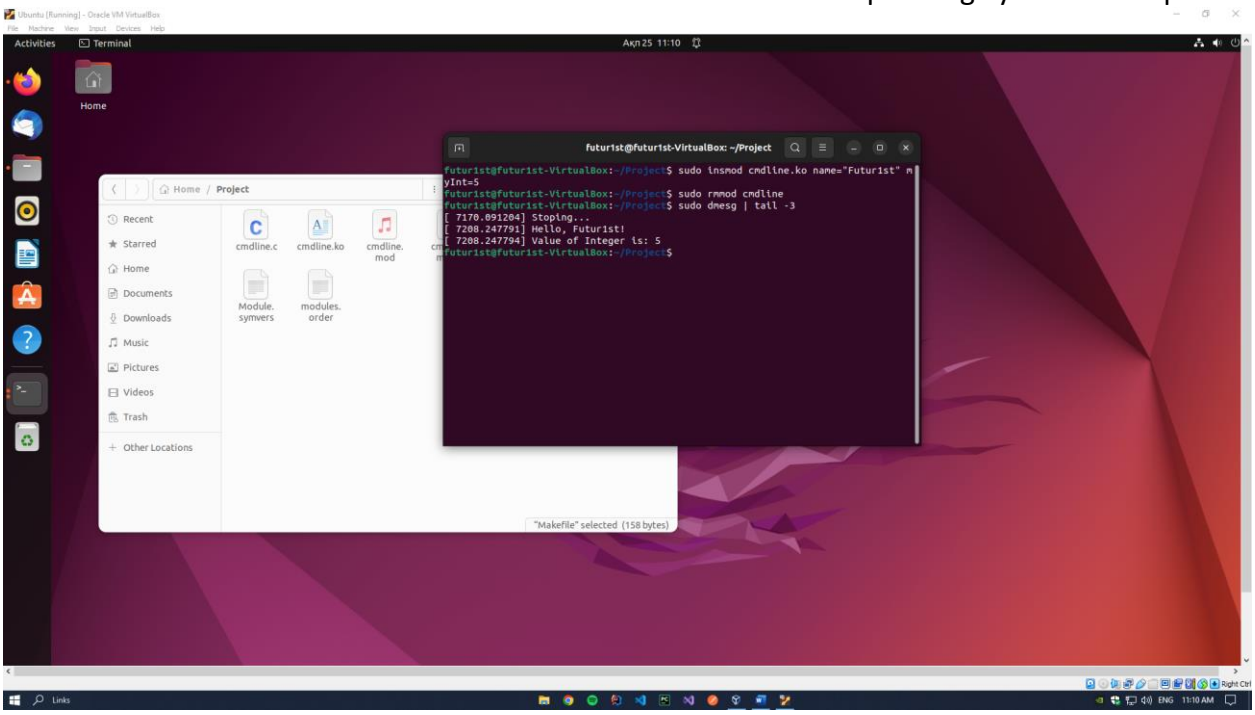


Now we will write code on kernel, that will accept our values from CMD line



Creating the compiler of this code





Conclusion:

We learned how to update and create our own kernel. They remembered how to install libraries on Linux OS and wrote their codes in C using the Linux Kernel. We think it was an unforgettable experience for us, thank you for such a wonderful course.