

زهره صداقت (۹۹۰۱۲۶۸۱۰۰۳) - گروه ۱۳
تمرین ۱

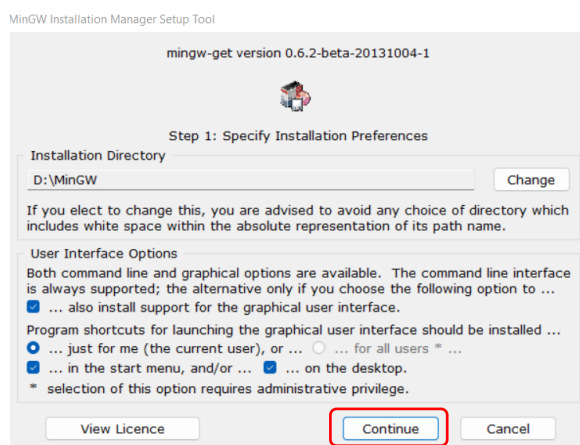
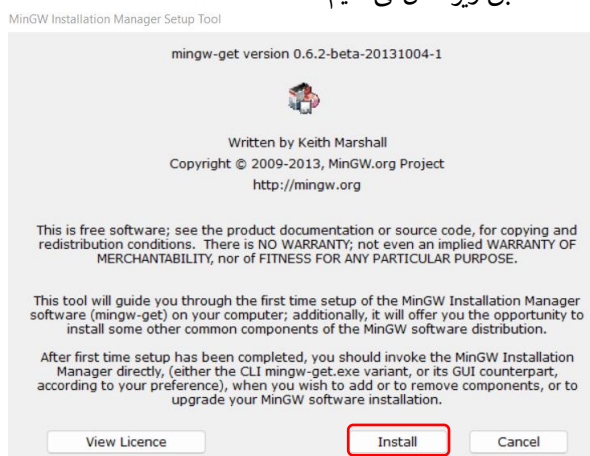
۱. کدهای نوشته شده به زبان C را می توان هم از طریق IDE ها و هم از طریق ترمینال (CMD در ویندوز) کامپایل و اجرا کرد. لازم است عمل، نصب کامپایلر زبان C یا GCC می باشد.

Windows:

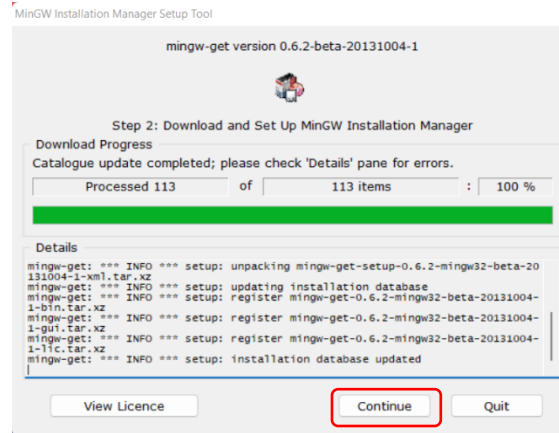
در ویندوز، با نصب کامپایلر MinGW GCC به کمک رابط های گرافیکی به صورت زیرو انجام عملیات لازم، میتوانیم صحت نصب را در CMD با دستورات مرتبط بررسی کنیم. برای نصب، ابتدا فایل exe. را از مراجع مربوطه دانلود می کنیم:



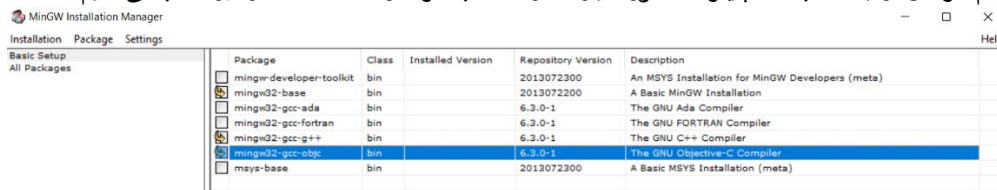
سپس برای نصب فایل دانلود شده مطابق زیر عمل می کنیم:



زهره صداقت (۹۹۰۱۲۲۶۸۱۰۰۳) – گروه ۱۳
تمرین ۱



پس از اتمام مراحل اولیه نصب، کامپایلرهای مورد نیازمان را انتخاب کرده و مجدداً آن‌ها را نیز نصب می‌کنیم:

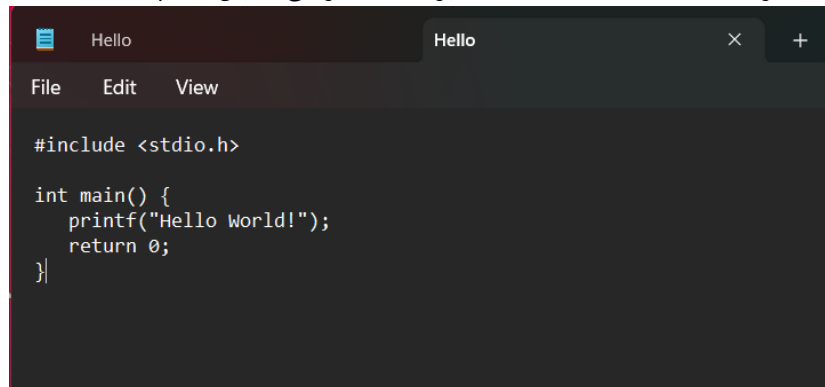


در این مرحله به سراغ تست صحت نصب کامپایلر به کمک دستورات در CMD می‌رویم:

```
PS C:\Users\User> gcc -v
Using built-in specs.
COLLECT_GCC=D:\Program Files\CodeBlocks\MinGW\bin\gcc.exe
COLLECT_LTO_WRAPPER=D:\Program Files\CodeBlocks\MinGW\bin\..\libexec/gcc/x86_64-w64-mingw32/8.1.0/lto-wrapper.exe
Target: x86_64-w64-mingw32
Configured with: ../../src/gcc-8.1.0/configure --host=x86_64-w64-mingw32 --build=x86_64-w64-mingw32 --target=x86_64-w64-mingw32 --prefix=/mingw64 --with-sysroot=/c:/mingw810/x86_64-810-posix-seh-rt_v6-rev0/mingw64 --enable-shared --enable-static --disable-multilib --enable-languages=c,c++,fortran,lto --enable-libstdcxx-time=yes --enable-threads=posix --enable-libgomp --enable-libatomic --enable-lto --enable-graphite --enable-checking=release --enable-fully-dynamic-string --enable-version-specific-runtime-libs --disable-libstdcxx-pch --disable-libstdcxx-debug --enable-bootstrap --disable-rpath --disable-win32-registry --disable-nls --disable-werror --disable-symvers --with-gnu-as --with-gnu-ld --with-arch=nocona --with-tune=core2 --with-libiconv --with-system-zlib --with-gmp=/c:/mingw810/prerequisites/x86_64-w64-mingw32-static --with-mpfr=/c:/mingw810/prerequisites/x86_64-w64-mingw32-static --with-mpc=/c:/mingw810/prerequisites/x86_64-w64-mingw32-static --with-isl=/c:/mingw810/prerequisites/x86_64-w64-mingw32-static --with-pkgversion='x86_64-posix-seh-rev0, Built by MinGW-W64 project' --with-bugurl=https://sourceforge.net/projects/mingw-w64 CFLAGS='-O2 -pipe -fno-ident -I/c:/mingw810/x86_64-810-posix-seh-rt_v6-rev0/mingw64/opt/include -I/c:/mingw810/prerequisites/x86_64-zlib-static/include -I/c:/mingw810/prerequisites/x86_64-w64-mingw32-static/include' CXXFLAGS='-O2 -pipe -fno-ident -I/c:/mingw810/x86_64-810-posix-seh-rt_v6-rev0/mingw64/opt/include -I/c:/mingw810/prerequisites/x86_64-zlib-static/include -I/c:/mingw810/prerequisites/x86_64-w64-mingw32-static/include' LDLAGS='-pipe -fno-ident -L/c:/mingw810/x86_64-810-posix-seh-rt_v6-rev0/mingw64/opt/lib -L/c:/mingw810/prerequisites/x86_64-zlib-static/lib -L/c:/mingw810/prerequisites/x86_64-w64-mingw32-static/lib'
Thread model: posix
gcc version 8.1.0 (x86_64-posix-seh-rev0, Built by MinGW-W64 project)
PS C:\Users\User>
```

به کمک دستور gcc -v درخواست پرنیت ورژن کامپایلر gcc نصب شده را اعلام می‌کنیم. در این جا اگر مراحل نصب با موفقیت انجام نشده بود، با ارور نامفهوم بودن دستور gcc مواجه می‌شدیم.

سپس به کمک یک Text Editor، کدی ساده به زبان C می‌نویسیم در زمان ذخیره در انتهای نام فایل، پسوند c. را اضافه می‌کنیم. (محتوای کد، قرار است رشته Hello World! را برای ما در خروجی نمایش دهد.)

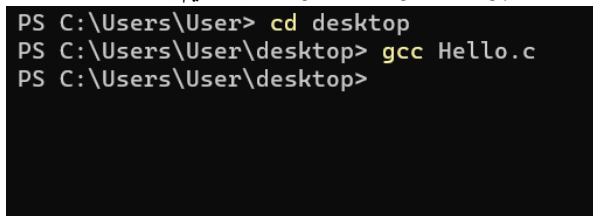


```

#include <stdio.h>

int main() {
    printf("Hello World!");
    return 0;
}
    
```

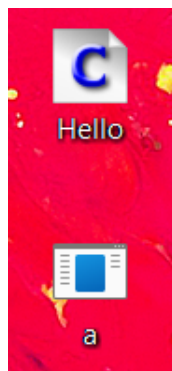
پس از ذخیره‌سازی فایل، در CMD دستوری مطابق زیر را جهت کامپایل کردن کد نوشته شده، می‌دهیم. (بدیهی است که قبل صدور دستور کامپایل، ابتدا باید به پوشه محتوی فایل مورد نظر برویم.)



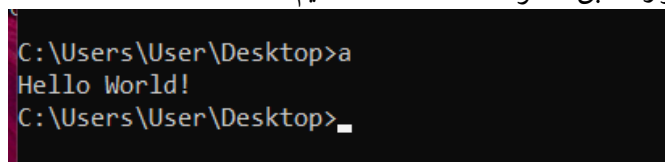
```

PS C:\Users\User> cd desktop
PS C:\Users\User\desktop> gcc Hello.c
PS C:\Users\User\desktop>
    
```

در این مرحله، در صورتی که دستور مورد نظر بدون خطا و به درستی اجرا شده باشد، با هیچ خروجی‌ای مواجه نخواهیم شد و اگر به واسطه گرافیکی پوشه محتوی فایل Hello.c برویم، با فایلی جدید رو به رو خواهیم شد که فایل اجرایی و کامپایل شده کد نوشته شده می‌باشد.



حال برای اجرای فایل ساخته شده توسط کامپایلر می‌توانیم با وارد کردن نام فایل ساخته شده توسط کامپایلر در CMD، خروجی را مطابق تصویر زیر و مطابق کد نوشته شده مشاهده کنیم.



```

C:\Users\User\Desktop>a
Hello World!
C:\Users\User\Desktop>_
    
```

:Linux

در لینوکس، می‌توانیم بدون ارتباط مستقیم با محیط گرافیکی و تنها از طریق ترمینال به دالود، نصب و راه‌اندازی کامپایلرهای زبان C و اجرای کدهای این زبان بپردازیم. ابتدا برای اطمینان از نصب نبودن کامپایلر GCC (مطابق ویندوز)، دستور gcc را در ترمینال لینوکس تایپ می‌کنیم. در صورتی که کامپایلر مورد نظر نصب نشده باشد یا مراحل نصب آن به درستی انجام نشده باشند، با خطای زیر مواجه خواهیم شد.

و برای نصب کامپایلر مذکور، دستور `sudo apt install gcc` استفاده می‌کنیم.

```
zahra@zahra-vm:~/Desktop$ gcc
Command 'gcc' not found, but can be installed with:
sudo apt install gcc
zahra@zahra-vm:~/Desktop$ sudo apt install gcc
```

پس از وارد کردن این دستور، با ارور زیر مواجه می‌شویم که با اضافه کردن repositoryهای زیر و وارد کردن مجدد دستور نصب gcc حل می‌شوند.

```
Reading state information... Done
E: Unable to locate package GCC
zahra@zahra-vm: $ sudo add-apt-repository main
Adding component(s) 'main' to all repositories.
Press [ENTER] to continue or Ctrl-c to cancel.
Hit:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:2 http://ir.archive.ubuntu.com/ubuntu jammy InRelease
Hit:3 http://ir.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:4 http://ir.archive.ubuntu.com/ubuntu jammy-backports InRelease
Reading package lists... Done
zahra@zahra-vm: $ sudo add-apt-repository universe
Adding component(s) 'universe' to all repositories.
Press [ENTER] to continue or Ctrl-c to cancel.
Hit:1 http://ir.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://ir.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://ir.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
zahra@zahra-vm: $ sudo add-apt-repository restricted
Adding component(s) 'restricted' to all repositories.
Press [ENTER] to continue or Ctrl-c to cancel.
Hit:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:2 http://ir.archive.ubuntu.com/ubuntu jammy InRelease
Hit:3 http://ir.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:4 http://ir.archive.ubuntu.com/ubuntu jammy-backports InRelease
Reading package lists... Done
zahra@zahra-vm: $ sudo add-apt-repository multiverse
Adding component(s) 'multiverse' to all repositories.
Press [ENTER] to continue or Ctrl-c to cancel.
Hit:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:2 http://ir.archive.ubuntu.com/ubuntu jammy InRelease
Hit:3 http://ir.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:4 http://ir.archive.ubuntu.com/ubuntu jammy-backports InRelease
Reading package lists... Done
zahra@zahra-vm: $ sudo apt install gcc
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

مطابق عملیاتی که برای چک کردن صحت نصب GCC در ویندوز انجام دادیم را در ترمینال لینوکس نیز انجام می‌دهیم و خروجی زیر را در صورت انجام درست عملیات نصب شاهد خواهیم بود.

```

zahra@zahra-vm: ~/Desktop$ gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/lib/gcc/x86_64-linux-gnu/11/lto-wrapper
OFFLOAD_TARGET_NAMES=nvptx-none:amdgc- amdhsa
OFFLOAD_TARGET_DEFAULT=1
Target: x86_64-linux-gnu
Configured with: ./src/configure -v --with-pkgversion='Ubuntu 11.3.0-1ubuntu1-22.04' --with-bugurl=file:///usr/share/doc/gcc-11/README.Bugs
--enable-languages=c,ada,c++,go,brig,d,fortran,objc,obj-c++,m2 --prefix=/usr --with-gcc-major-version-only --program-suffix=-11 --program-pre
fix=x86_64-linux-gnu- --enable-shared --enable-linker-build-id --libexecdir=/usr/lib --without-included-gettext --enable-threads=posix --libd
ir=/usr/lib --enable-nls --enable-bootstrap --enable-clocale=gnu --enable-libstdc++-debug --enable-libstdc++-time=yes --with-default-libstdc++
x-abi=new --enable-gnu-unique-object --disable-vtable-verify --enable-plugin --enable-default-pie --with-system-zlib --enable-libphobos-check
ing=release --with-target-system-zlib=auto --enable-objc-gc=auto --enable-multilarch --disable-werror --enable-cet --with-arch=32-l686 --with-abi=m64
--with-multilib-list=m32,m64,mx32 --enable-multilib --with-tune=generic --enable-offload-targets=nvptx-none=/build/gcc-11-xKlWf/gcc-11-xKlWf/gcc-
11-11.3.0/debian/tmp-nvptx/usr,amdgc- amdhsa=/build/gcc-11-xKlWf/gcc-11-11.3.0/debian/tmp-gcn/usr --without-cuda-driver --enable-checking=re
lease --build=x86_64-linux-gnu --host=x86_64-linux-gnu --target=x86_64-linux-gnu --with-build-config=bootstrap-lto-lean --enable-link-seria
lization=2
Thread model: posix
Supported LTO compression algorithms: zlib zstd
gcc version 11.3.0 (Ubuntu 11.3.0-1ubuntu1-22.04)
zahra@zahra-vm: ~/Desktop$

```

مشابه کدی که در مثال ویندوز در یک Text Editor نوشت و با پسوند c. ذخیره کردیم را نوشته و با همان فرمت در یک پوشه (مثلا دسکتاپ) ذخیره می‌کنیم:

```

1 #include <stdio.h>
2
3 int main() {
4     printf("Hello World!");
5     return 0;
6 }

```

حال به کمک دستور زیر در ترمینال، کد نوشته شده را کامپایل می‌کنیم.

```

zahra@zahra-vm: ~/Desktop$ gcc Hello.c
zahra@zahra-vm: ~/Desktop$

```

در این مرحله، اگر به محل ذخیره فایل c. نوشته شده برویم فایل کامپایل شده را نیز خواهیم دید.



برای اجرای فایل ایجاد شده توسط کامپایلر نیز مطابق زیر در ترمینال عمل می‌کنیم:

```

zahra@zahra-vm: ~/Desktop$ ./a.out
Hello World!
zahra@zahra-vm: ~/Desktop$

```

۲. در حال حاضر فایل‌های موجود در دسکتاپ به صورت زیر می‌باشند:

```

zahra@zahra-vm: ~/Desktop$ ls -l
total 44
-rw-rw-r-- 1 zahra zahra 13 2023 5      می a
-rwxrwxr-x 1 zahra zahra 15960 17:29 5    می a.out
-rw-rw-r-- 1 zahra zahra 34 2023 5      می b
-rw-rw-r-- 1 zahra zahra 22 2023 5      می c.py
drwxrwxr-x 2 zahra zahra 4096 2023 5    می f1
drwxrwxr-x 2 zahra zahra 4096 2023 5    می f2
drwxrwxr-x 2 zahra zahra 4096 2023 5    می f3
-rw-rw-r-- 1 zahra zahra 75 17:28 5     می Hello.c
zahra@zahra-vm: ~/Desktop$

```


دستور زیر را برای مشخص شدن نوع فایل ایجاد شده در سوال قبل اجرا می‌کنیم:

```
zahra@zahra-vm:~/Desktop$ file Hello.c
Hello.c: C source, ASCII text
zahra@zahra-vm:~/Desktop$
```

پس از اجرای دستور بالا، خواهیم دید که نوع فایل را از نوع C و تکست به فرمت ASCII اعلام می‌کند.

سپس با اجرای دستور زیر نام فایل را طبق خواسته سوال از Hello.c به Hello تغییر می‌دهیم:

```
zahra@zahra-vm:~/Desktop$ mv Hello.c Hello
```

و مجدداً به کمک دستور file درخواست اعلام نوع فایل را صادر می‌کنیم. همینطور که در تصویر زیر قابل مشاهده است، نوع فایل از نظر سیستم عامل به دلیل محتوایش تغییری نکرده. در نتیجه حذف پسوند فایل تغییری روی نوع فایل ایجاد نمی‌کند.

```
zahra@zahra-vm:~/Desktop$ file Hello
Hello: C source, ASCII text
zahra@zahra-vm:~/Desktop$
```

۳. برای ایجاد و اجرای یک فایل در لینوکس و طریق ترمینال آن، ابتدا باید از نصب بودن مفسر پایتون بر روی سیستم اطمینان حاصل کنیم:

```
zahra@zahra-vm:~/Desktop$ cd
zahra@zahra-vm:~$ sudo apt install python3
[sudo] password for zahra:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3 is already the newest version (3.10.6-1~22.04).
python3 set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 162 not upgraded.
```

در این جا خواهیم دید که مفسر آن از قبل نصب شده است و نیازی به آپدیت هم ندارد.

در قدم بعدی، به دو روش برای نوشتن و اجرای کد پایتون به کمک ترمینال بر می‌خوریم. روش اول به این صورت است که فایلی را به کمک ترمینال ایجاد کرده و محتوای کد پایتون را داخل فایل نوشته و به کمک دستور `python3 file_name` کد نوشته شده را اجرا کرده و خروجی را در ترمینال مشاهده کنیم. روش اول به صورت زیر خواهد بود:

```
zahra@zahra-vm:~/Desktop$ gedit test.py
```

برای فایل ایجاد شده توسط gedit، نامی با پسوند فایل‌های پایتون می‌گذاریم و محتوای کد را در gedit به صورت زیر ذخیره می‌کنیم:

```
Open  [F1]
1 print('This is a test')
```

برای مشاهده خروجی این کد، دستور زیر را (با توجه به نام فایل) در ترمینال وارد می‌کنیم و خروجی کد را مشاهده خواهیم کرد:

```
zahra@zahra-vm:~/Desktop$ gedit test.py
zahra@zahra-vm:~/Desktop$ python3 test.py
This is a test
```

روش دوم بدین صورت است که خط به خط کدمان را در همان ترمینال نوشته و هر خط را در همان لحظه اجرا کنیم. این عمل با دستور python3 به صورت زیر ممکن است:

```
zahra@zahra-vm:~/Desktop$ python3
Python 3.10.6 (main, Nov 14 2022, 16:10:14) [GCC 11.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print('Terminal python test')
Terminal python test
>>>
```

همان طور که در تصویر بالا مشخص است، ترمینال پس از دریافت و اجرای دستور اول همچنان پذیرای خط دیگری از کد پایتون است.

۴. اولین کاری لازم است برای اتصال بی سیم در سیستم های لینوکسی انجام دهیم، نوشتن دستور زیر در ترمینال است:

```
zahra@zahra-vm:~$ iwconfig
lo          no wireless extensions.

ens33      no wireless extensions.
```

در این جا لیست اتصالات بی سیم ممکن نمایش داده می شود. بدیهی است که در این سیستم هیچ اتصالی از نوع بی سیم به اینترنت وجود ندارد.

حال برای استفاده از دستورات بیشتر نیاز است پکیجی به نام net-tools را نصب کنیم:

```
zahra@zahra-vm:~$ sudo apt install net-tools
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  net-tools
0 upgraded, 1 newly installed, 0 to remove and 162 not upgraded.
Need to get 204 kB of archives.
After this operation, 819 kB of additional disk space will be used.
Get:1 http://ir.archive.ubuntu.com/ubuntu jammy/main amd64 net-tools amd64 1.60+git20181103.0eebece-1ubuntu5 [204 kB]
Ign:1 http://ir.archive.ubuntu.com/ubuntu jammy/main amd64 net-tools amd64 1.60+git20181103.0eebece-1ubuntu5
Get:1 http://ir.archive.ubuntu.com/ubuntu jammy/main amd64 net-tools amd64 1.60+git20181103.0eebece-1ubuntu5 [204 kB]
Fetched 163 kB in 56s (2,933 B/s)
Selecting previously unselected package net-tools.
(Reading database ... 165695 files and directories currently installed.)
Preparing to unpack .../net-tools_1.60+git20181103.0eebece-1ubuntu5_amd64.deb ...
Unpacking net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Setting up net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Processing triggers for man-db (2.10.2-1) ...
zahra@zahra-vm:~$
```

سپس لازم است از روشن بودن و صحت عملکرد واسط شبکه سیستم اطمینان حاصل کنیم. این کار را به کمک دستور زیر به این صورت انجام می دهیم:

```
zahra@zahra-vm:~$ sudo ifconfig lo up
zahra@zahra-vm:~$ sudo ifconfig ens33 up
zahra@zahra-vm:~$
```

حال لازم است در محیط اطراف به دنبال نقاط اتصال اینترنت فعال باشیم. این کار را با دستور و به صورت زیر انجام می دهیم:

```
zahra@zahra-vm:~$ sudo iwlist scan | more
lo          Interface doesn't support scanning.

ens33      Interface doesn't support scanning.
```

برای حل مشکل بالا از تمام دستورات زیر استفاده شد:

```
zahra@zahra-vm:~$ sudo ifconfig lo
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 4248 bytes 415725 (415.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4248 bytes 415725 (415.7 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

zahra@zahra-vm:~$ sudo ifconfig lo up
zahra@zahra-vm:~$ sudo ifconfig ens33 up
zahra@zahra-vm:~$ iwconfig
lo          no wireless extensions.

ens33       no wireless extensions.

zahra@zahra-vm:~$ sudo iwlist scan | more
lo          Interface doesn't support scanning.

ens33       Interface doesn't support scanning.

zahra@zahra-vm:~$ nmcli dev wifi
zahra@zahra-vm:~$ man nmcli
zahra@zahra-vm:~$ nmcli device status
DEVICE  TYPE      STATE      CONNECTION
ens33   ethernet  connected  Wired connection 1
lo       loopback  unmanaged  --
zahra@zahra-vm:~$ nmcli dev list
Error: argument 'list' not understood. Try passing --help instead.

zahra@zahra-vm:~$ nmcli dev show
GENERAL.DEVICE: ens33
GENERAL.TYPE: ethernet
GENERAL.HWADDR: 00:0C:29:78:64:29
GENERAL.MTU: 1500
GENERAL.STATE: 100 (connected)
GENERAL.CONNECTION: Wired connection 1
GENERAL.CON-PATH: /org/freedesktop/NetworkManager/ActiveConnection/3
WIRED-PROPERTIES.CARRIER: on
IP4.ADDRESS[1]: 192.168.192.129/24
IP4.GATEWAY: 192.168.192.2
IP4.ROUTE[1]: dst = 169.254.0.0/16, nh = 0.0.0.0, mt = 1000
IP4.ROUTE[2]: dst = 192.168.192.0/24, nh = 0.0.0.0, mt = 100
IP4.ROUTE[3]: dst = 0.0.0.0/0, nh = 192.168.192.2, mt = 100
IP4.DNS[1]: 192.168.192.2
IP4.DOMAIN[1]: localdomain
IP6.ADDRESS[1]: fe80::f6a5:4d:2410:9f51/64
IP6.GATEWAY: --
IP6.ROUTE[1]: dst = fe80::/64, nh = ::, mt = 1024

GENERAL.DEVICE: lo
GENERAL.TYPE: loopback
GENERAL.HWADDR: 00:00:00:00:00:00
GENERAL.MTU: 65536
GENERAL.STATE: 10 (unmanaged)
GENERAL.CONNECTION: --
GENERAL.CON-PATH: --
IP4.ADDRESS[1]: 127.0.0.1/8
IP4.GATEWAY: --
IP6.ADDRESS[1]: ::1/128
IP6.GATEWAY: --
IP6.ROUTE[1]: dst = ::1/128, nh = ::, mt = 256
zahra@zahra-vm:~$ nmcli con list
Error: argument 'list' not understood. Try passing --help instead.
```



```
zahra@zahra-vm:~$ nmcli con show
NAME                UUID                                  TYPE      DEVICE
Wired connection 1  db02c6f1-fdb7-32ed-b5f1-3a66af1b1040 ethernet  ens33
zahra@zahra-vm:~$ sudo iwlist eth1 scan
eth1                Interface doesn't support scanning.

zahra@zahra-vm:~$ sudo ifconfig eth1 && sudo iwlist eth1 scan
eth1: error fetching interface information: Device not found
zahra@zahra-vm:~$ iw dev wlan0 scan ap-force
Command 'iw' not found, but can be installed with:
sudo apt install iw
zahra@zahra-vm:~$ iw dev lo scan ap-force
Command 'iw' not found, but can be installed with:
sudo apt install iw
zahra@zahra-vm:~$ sudo apt install iw
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  iw
0 upgraded, 1 newly installed, 0 to remove and 162 not upgraded.
Need to get 108 kB of archives.
After this operation, 305 kB of additional disk space will be used.
Get:1 http://ir.archive.ubuntu.com/ubuntu jammy/main amd64 iw amd64 5.16-1build1 [108 kB]
Fetched 108 kB in 26s (4,244 B/s)
Selecting previously unselected package iw.
(Reading database ... 165744 files and directories currently installed.)
Preparing to unpack .../iw_5.16-1build1_amd64.deb ...
Unpacking iw (5.16-1build1) ...
Setting up iw (5.16-1build1) ...
Processing triggers for man-db (2.10.2-1) ...
zahra@zahra-vm:~$ iw dev lo scan ap-force
command failed: Operation not permitted (-1)
zahra@zahra-vm:~$ iw dev ens33 scan ap-force
command failed: Operation not permitted (-1)
zahra@zahra-vm:~$
```

از منابع زیر استفاده شد:

<https://www.geeksforgeeks.org/connecting-to-the-internet-using-command-line-in-linux>
<https://www.makeuseof.com/fix-ifconfig-command-not-found-error-linux/#:~:text=To%20be%20able%20to%20use,the%20net%2Dtools%20package%20first.&text=T.his%20should%20install%20the%20net,it%20from%20the%20command%20line>
<https://copyprogramming.com/howto/iwlist-interface-doesn-t-support-scanning-ubuntu-lts-10-04>
<https://askubuntu.com/questions/624572/terminal-says-interface-doesnt-support-scanning>