



دانشگاه جهرم

# آشنایی با سیستم عامل Ubuntu

نگارش:

امیررضا ارجمند

استاد:

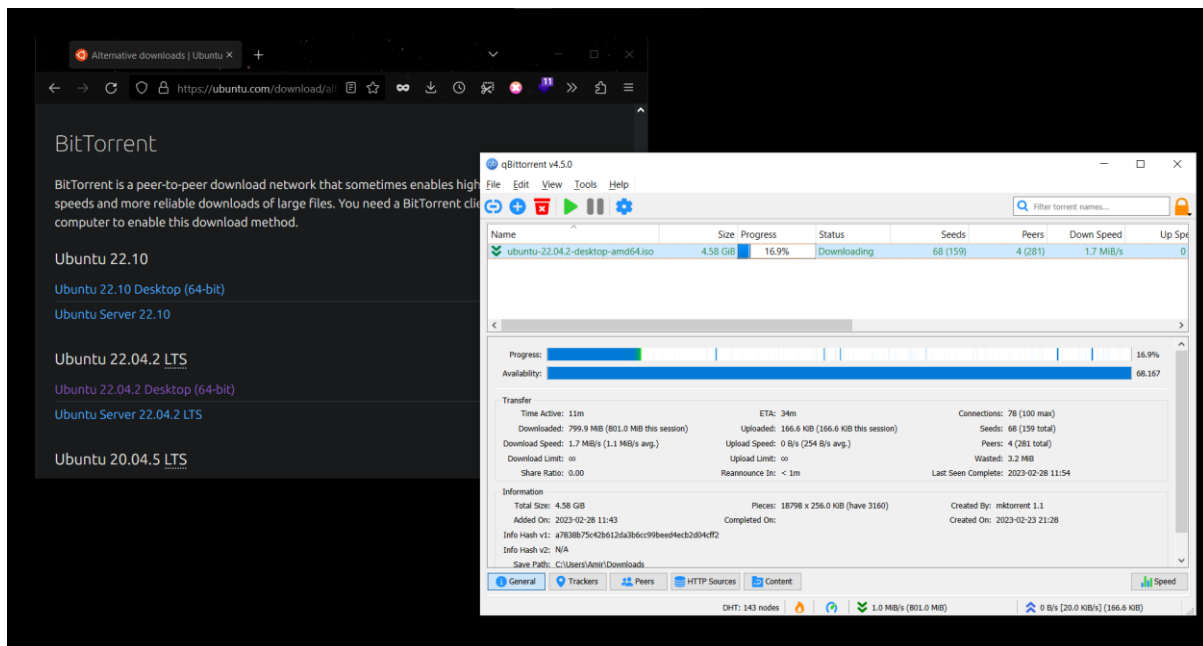
دکتر محمدجواد پارسه

شماره دانشجویی:

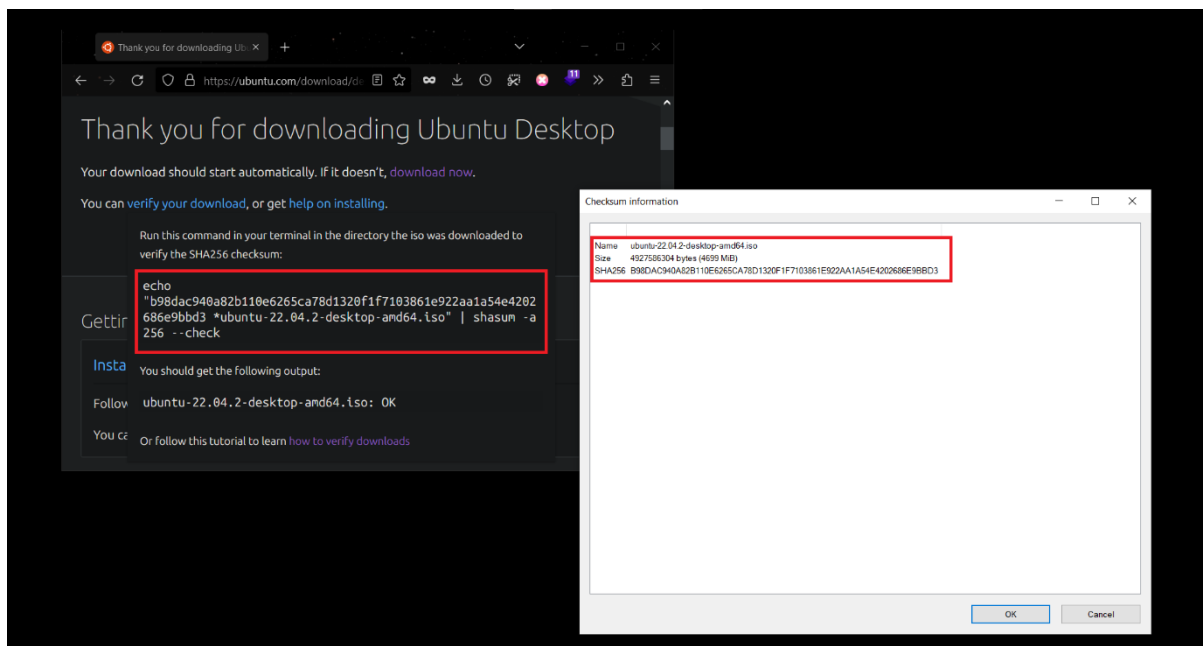
۹۹۳۱۲۲۰۱

## ۱. چگونگی نصب سیستم عامل Ubuntu

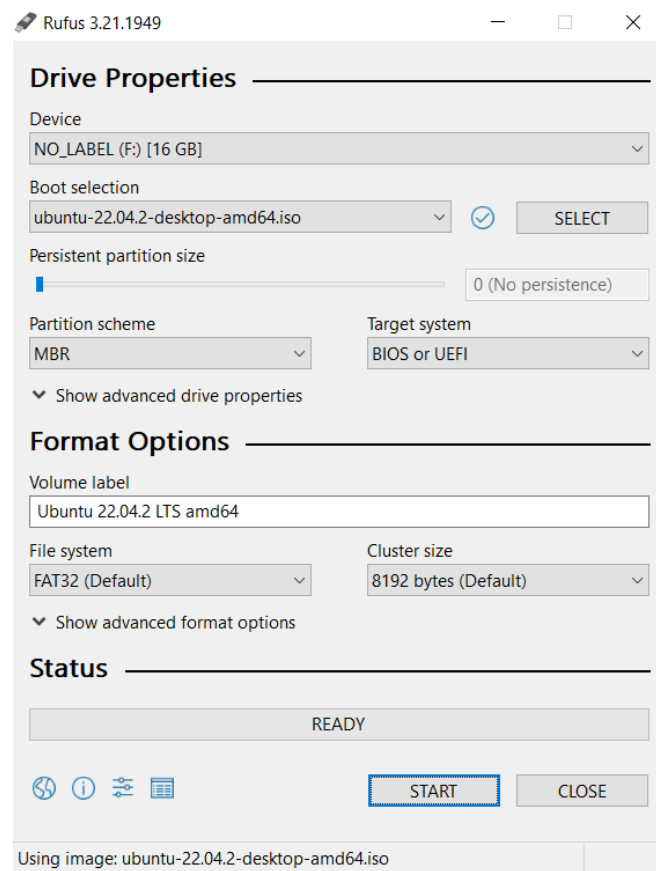
۱/۱. سیستم عامل Ubuntu را از وبسایت <https://ubuntu.com> دانلود یا torrent می‌کنیم.



۱/۲. برای سنجش صحت فایل دانلود شده، می‌توانیم sha256 checksum آن را محاسبه کنیم و با مقدار قرار داده شده در وبسایت Ubuntu مقایسه کنیم. مقادیر باید یکسان باشند.



۱/۳. با استفاده از نرم افزار Rufus فایل دانلود شده را در حالت bootable به یک فلش مموری انتقال می دهیم.

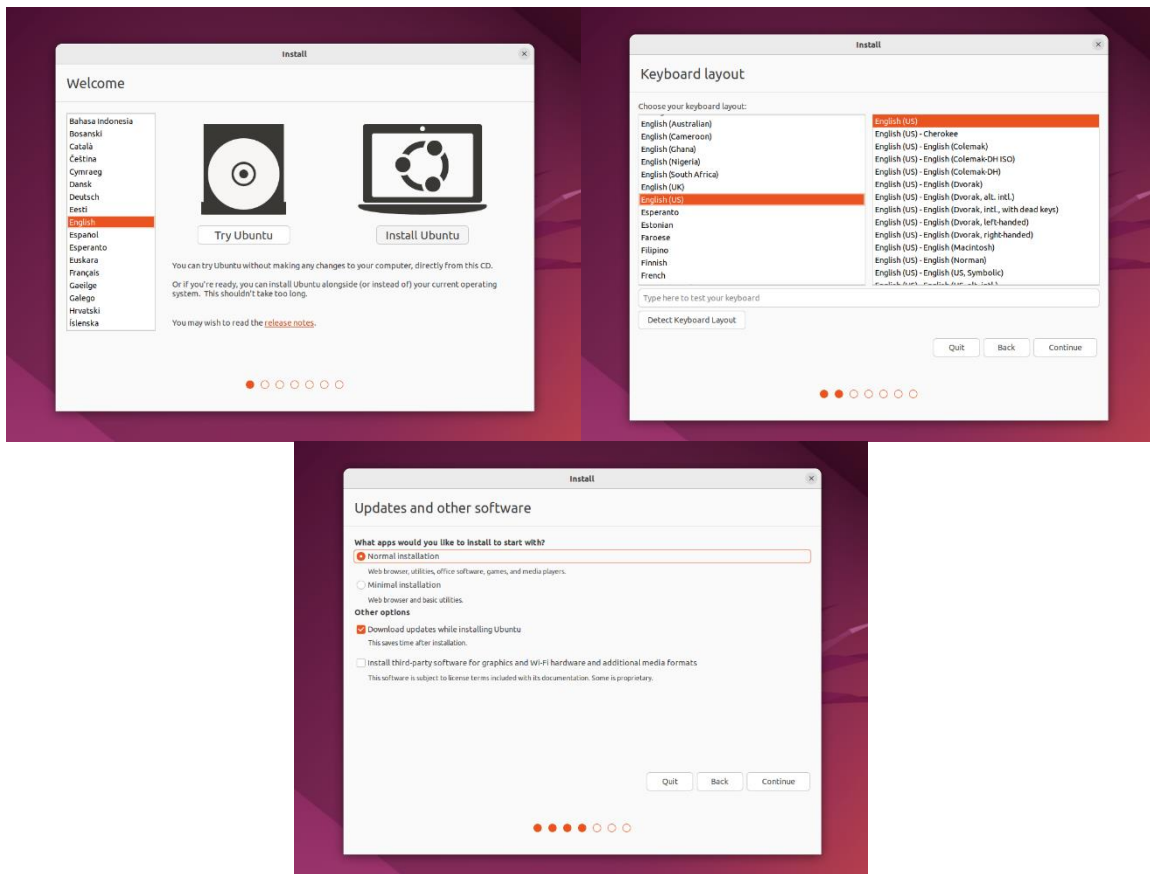


۱/۴. کامپیوتر خود را ریستارت کرده و آنرا از روی فلش مموری boot می کنیم (مراحل انجام این کار در پیکربندی های مختلف متفاوت است).

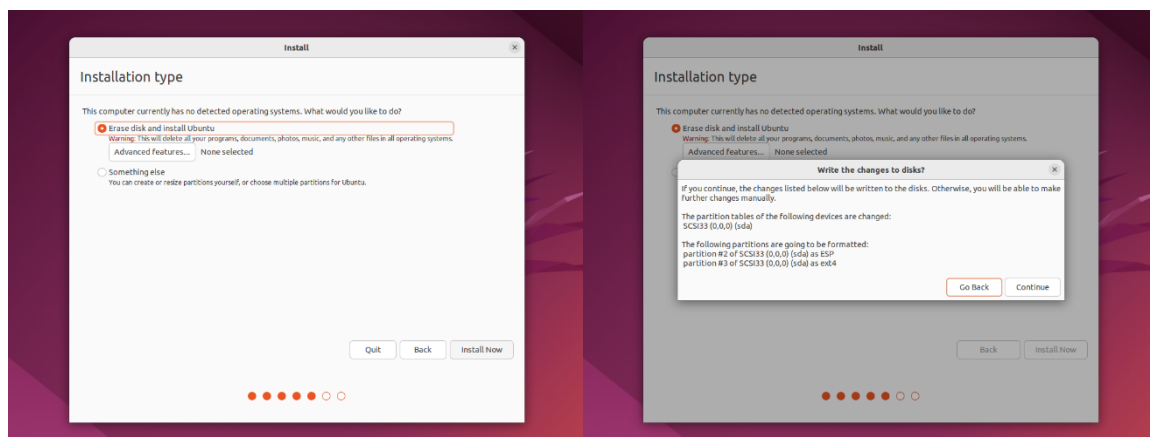
۱/۵. در صفحه ی جدید گزینه Try or Install Ubuntu را انتخاب می کنیم.



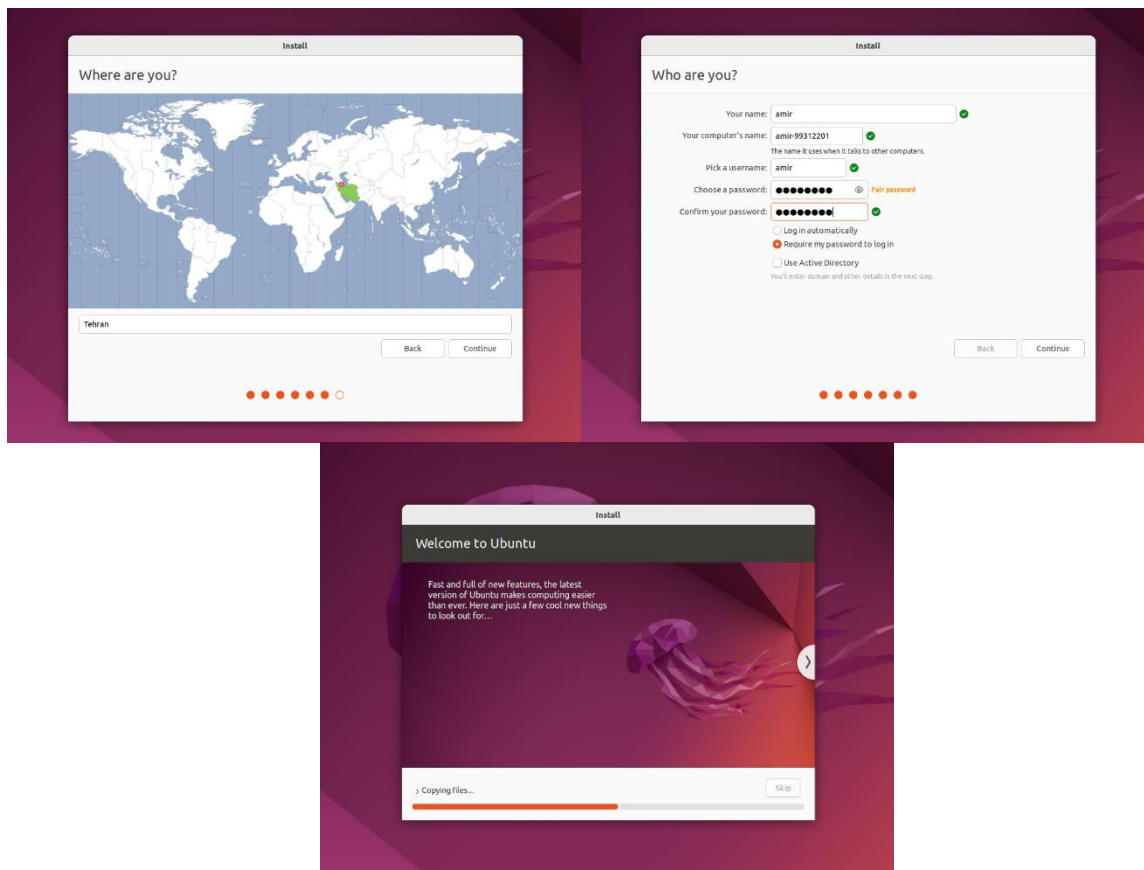
۱/۶. روی گزینه Install Ubuntu کلیک کرده و تنظیمات را طبق نیاز خود انتخاب می‌کنیم.



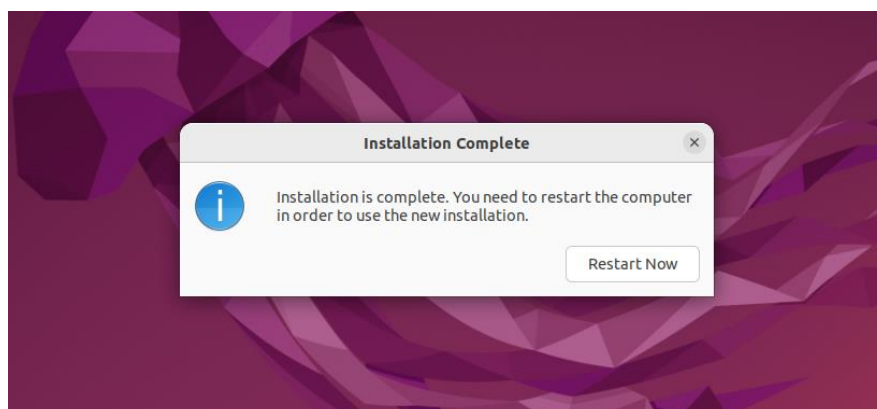
۱/۷. در صورت نیاز می‌توانیم در قسمت Installation type حافظه دیسک کامپیوتر خود را پارتیشن بندی کنیم و قسمت خاصی را برای نصب Ubuntu انتخاب کنیم. در اینجا نصب را در حالت پیش فرض انجام می‌دهیم.



۱/۸. به ادامه پیکربندی Ubuntu می‌پردازیم و اطلاعات مورد نظر خودمان را وارد می‌کنیم. پس از اتمام ورود اطلاعات با صفحه‌ای مواجه می‌شویم که مقدار پیشرفت نصب را نشان می‌دهد.



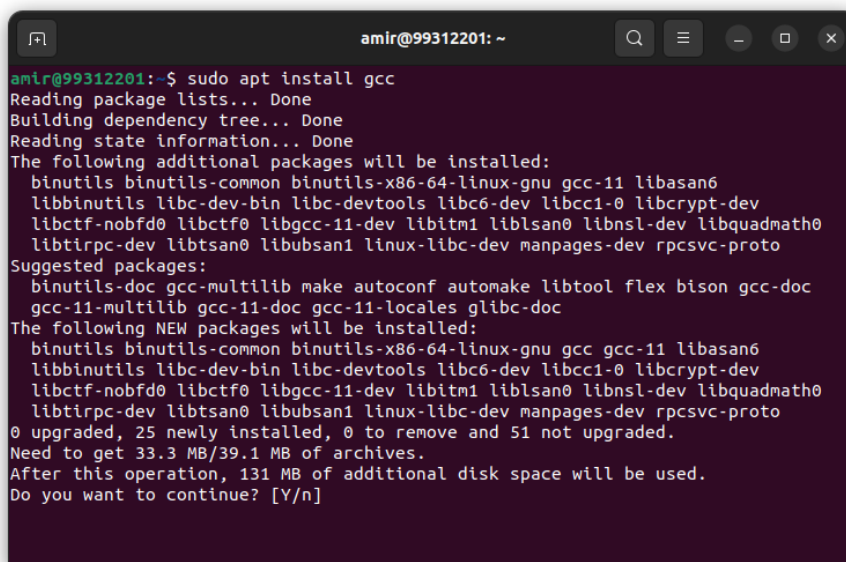
۱/۹. در پایان با پیام Installation Complete مواجه می‌شویم که نمایانگر اتمام نصب سیستم عامل می‌باشد و پس از ریستارت می‌توانیم از آن استفاده کنیم.



## ۲. نصب آخرین نسخه کامپایلر GCC

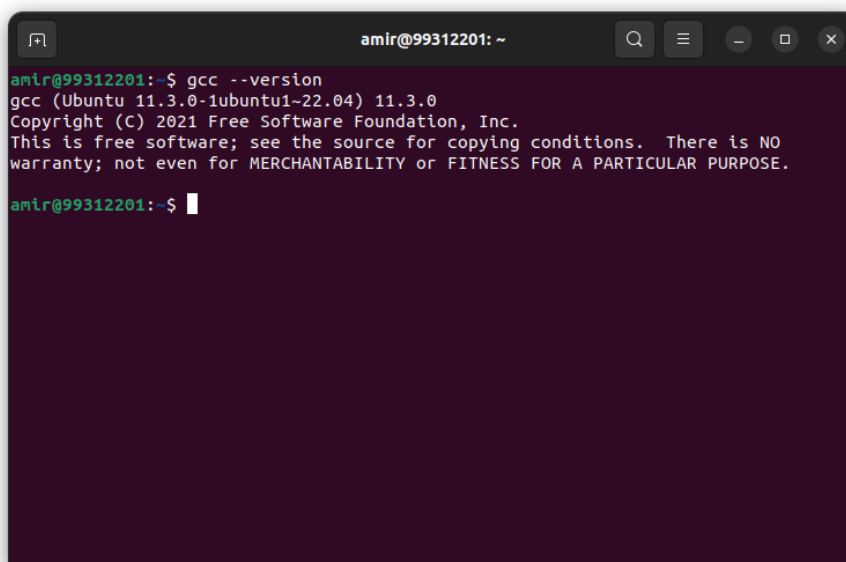
۲/۱. ابتدا دستورات `sudo apt update` و `sudo apt upgrade` را اجرا می‌کنیم تا پکیج‌های نصب شده روی سیستم به‌روز شوند.

۲/۲. دستور `sudo apt install gcc` را اجرا می‌کنیم تا کامپایلر GCC نصب شود.



```
amir@99312201: ~  
amir@99312201:~$ sudo apt install gcc  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following additional packages will be installed:  
  binutils binutils-common binutils-x86-64-linux-gnu gcc-11 libasan6  
  libbinutils libc-dev-bin libc-devtools libc6-dev libcc1-0 libcrypt-dev  
  libctf-nobfd0 libctf0 libgcc-11-dev libitm1 liblsan0 libnsl-dev libquadmath0  
  libtirpc-dev libtsan0 libubsan1 linux-libc-dev manpages-dev rpcsvc-proto  
Suggested packages:  
  binutils-doc gcc-multilib make autoconf automake libtool flex bison gcc-doc  
  gcc-11-multilib gcc-11-doc gcc-11-locales glibc-doc  
The following NEW packages will be installed:  
  binutils binutils-common binutils-x86-64-linux-gnu gcc gcc-11 libasan6  
  libbinutils libc-dev-bin libc-devtools libc6-dev libcc1-0 libcrypt-dev  
  libctf-nobfd0 libctf0 libgcc-11-dev libitm1 liblsan0 libnsl-dev libquadmath0  
  libtirpc-dev libtsan0 libubsan1 linux-libc-dev manpages-dev rpcsvc-proto  
0 upgraded, 25 newly installed, 0 to remove and 51 not upgraded.  
Need to get 33.3 MB/39.1 MB of archives.  
After this operation, 131 MB of additional disk space will be used.  
Do you want to continue? [Y/n]
```

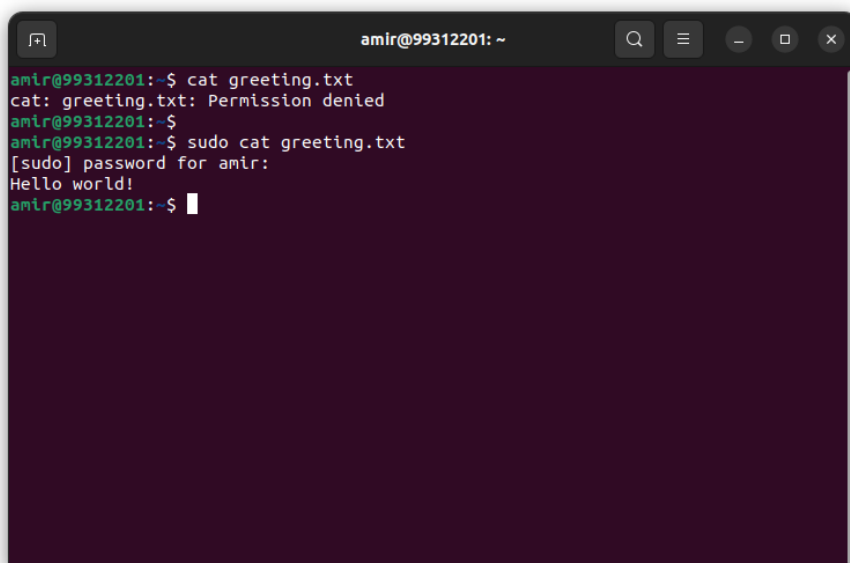
۲/۳. برای بررسی صحت نصب GCC می‌توانیم از دستور `gcc --version` استفاده کنیم.



```
amir@99312201: ~  
amir@99312201:~$ gcc --version  
gcc (Ubuntu 11.3.0-1ubuntu1~22.04) 11.3.0  
Copyright (C) 2021 Free Software Foundation, Inc.  
This is free software; see the source for copying conditions. There is NO  
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  
amir@99312201:~$
```

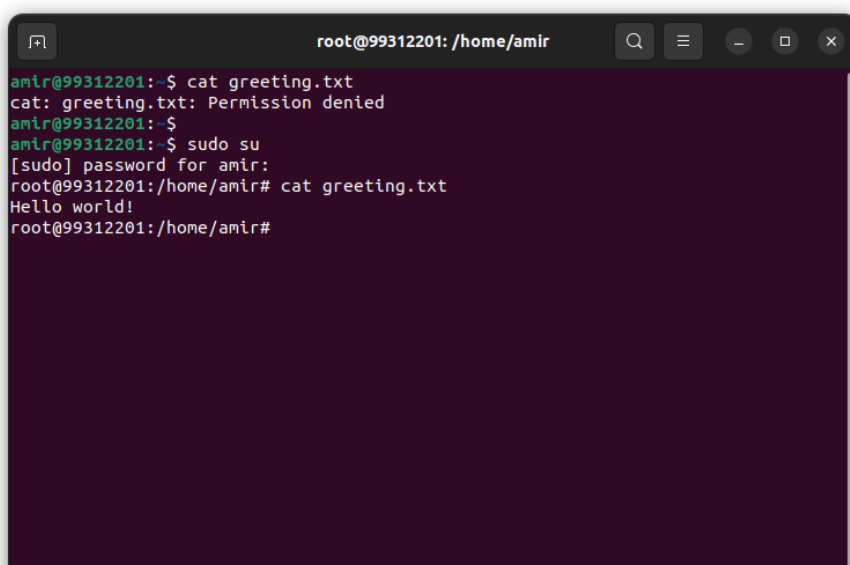
### ۳. استفاده از کاربر root

۳/۱. برای این کار می‌توانیم ابتدای هر دستور در terminal از کلمه‌ی کلیدی sudo استفاده کرده و پس از وارد کردن رمز خود دستور را به صورت ابرکاربر اجرا کنیم.



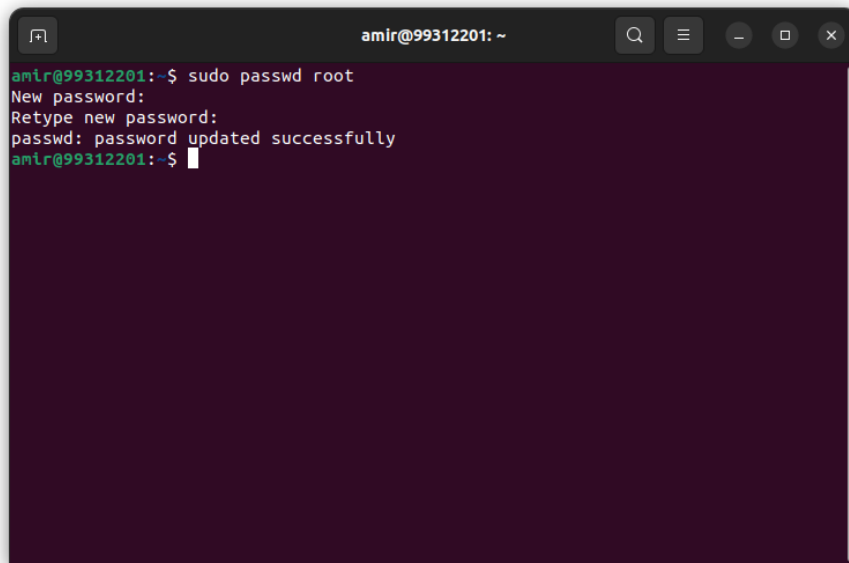
```
amir@99312201: ~  
amir@99312201:~$ cat greeting.txt  
cat: greeting.txt: Permission denied  
amir@99312201:~$  
amir@99312201:~$ sudo cat greeting.txt  
[sudo] password for amir:  
Hello world!  
amir@99312201:~$
```

۳/۲. همچنین می‌توانیم با اجرای دستور su sudo و وارد کردن رمز خود، تمامی دستورات بعد از آن را در shell به عنوان root اجرا کنیم.



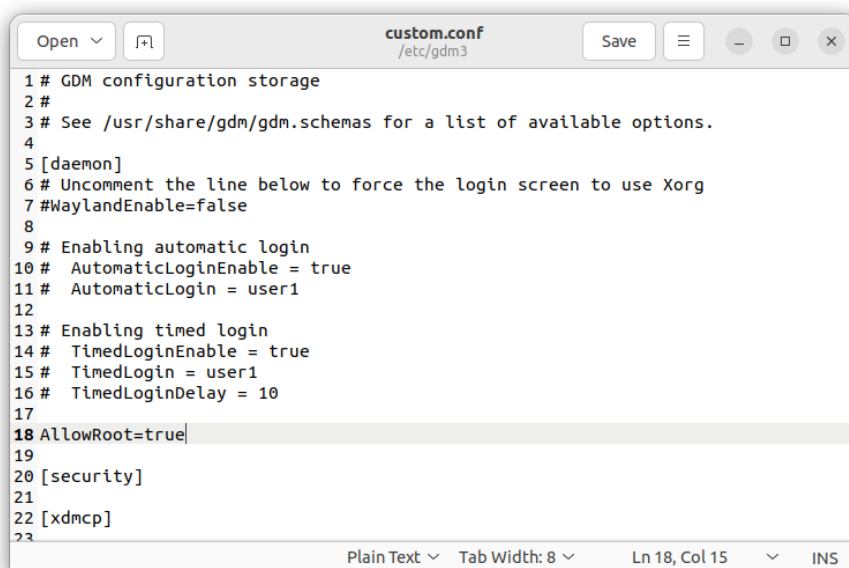
```
root@99312201: /home/amir  
amir@99312201:~$ cat greeting.txt  
cat: greeting.txt: Permission denied  
amir@99312201:~$  
amir@99312201:~$ sudo su  
[sudo] password for amir:  
root@99312201: /home/amir# cat greeting.txt  
Hello world!  
root@99312201: /home/amir#
```

۳/۳. مراحل فعال‌سازی کاربر root در محیط گرافیکی کمی پیچیده‌تر است. ابتدا با استفاده از دستور `sudo passwd root` برای کاربر root رمز عبوری تعیین می‌کنیم.



```
amir@99312201: ~  
amir@99312201:~$ sudo passwd root  
New password:  
Retype new password:  
passwd: password updated successfully  
amir@99312201:~$
```

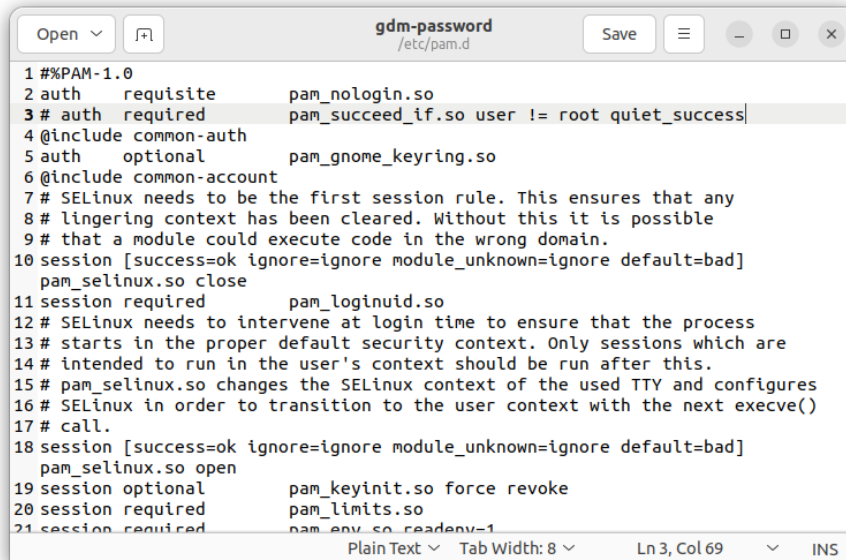
۳/۴. به فایل `/etc/gdm3/custom.conf` خط `AllowRoot=true` را اضافه می‌کنیم. این فایل مربوط به پیکربندی GDM می‌باشد که وظیفه مدیریت ورود کاربران در محیط گرافیکی به حساب خود را به عهده دارد.



```
custom.conf  
/etc/gdm3  
Open Save  
1 # GDM configuration storage  
2 #  
3 # See /usr/share/gdm/gdm.schemas for a list of available options.  
4  
5 [daemon]  
6 # Uncomment the line below to force the login screen to use Xorg  
7 #WaylandEnable=false  
8  
9 # Enabling automatic login  
10 # AutomaticLoginEnable = true  
11 # AutomaticLogin = user1  
12  
13 # Enabling timed login  
14 # TimedLoginEnable = true  
15 # TimedLogin = user1  
16 # TimedLoginDelay = 10  
17  
18 AllowRoot=true  
19  
20 [security]  
21  
22 [xdmcp]  
23  
Plain Text Tab Width: 8 Ln 18, Col 15 INS
```



۳/۵. خط `auth required pam_succeed_if.so user != root quiet_success` را در فایل `/etc/pam.d/gdm-password` کامنت می‌کنیم. این فایل مربوط به پیکربندی PAM daemon می‌باشد که وظیفه احراز هویت کاربران را به عهده دارد و این خط به طور پیش‌فرض جلوی ورود کاربر `root` را می‌گیرد.



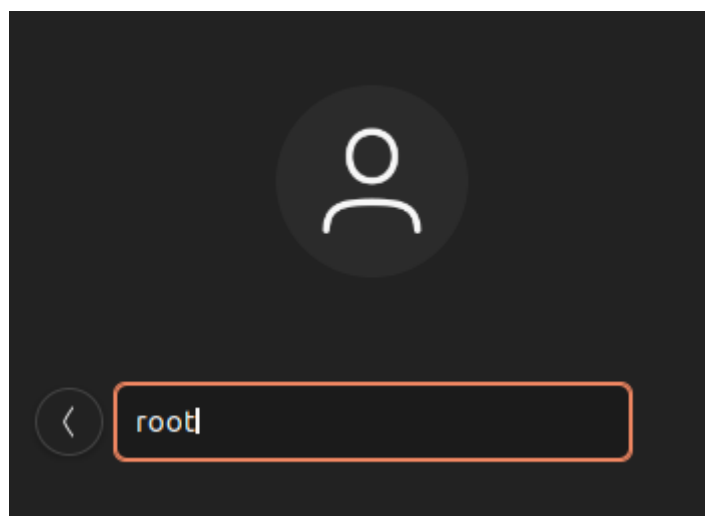
```
1 #%PAM-1.0
2 auth requisite pam_nologin.so
3 # auth required pam_succeed_if.so user != root quiet_success
4 @include common-auth
5 auth optional pam_gnome_keyring.so
6 @include common-account
7 # SELinux needs to be the first session rule. This ensures that any
8 # lingering context has been cleared. Without this it is possible
9 # that a module could execute code in the wrong domain.
10 session [success=ok ignore=ignore module_unknown=ignore default=bad]
11   pam_selinux.so close
12 session required pam_loginuid.so
13 # SELinux needs to intervene at login time to ensure that the process
14 # starts in the proper default security context. Only sessions which are
15 # intended to run in the user's context should be run after this.
16 # pam_selinux.so changes the SELinux context of the used TTY and configures
17 # SELinux in order to transition to the user context with the next execve()
18 # call.
19 session [success=ok ignore=ignore module_unknown=ignore default=bad]
20   pam_selinux.so open
21 session optional pam_keyinit.so force revoke
22 session required pam_limits.so
23 session required pam_env.so readenv=1
```

۳/۶. از حساب خود `logout` می‌کنیم.

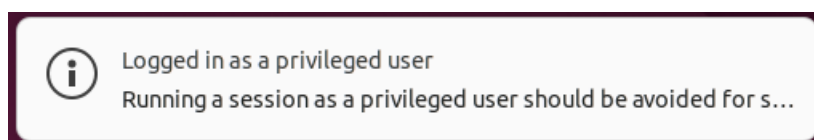
۳/۷. در صفحه ورود روی گزینه `Not listed?` کلیک می‌کنیم.



۳/۸. نام کاربری root و رمز عبوری که انتخاب کردیم را وارد می‌کنیم.

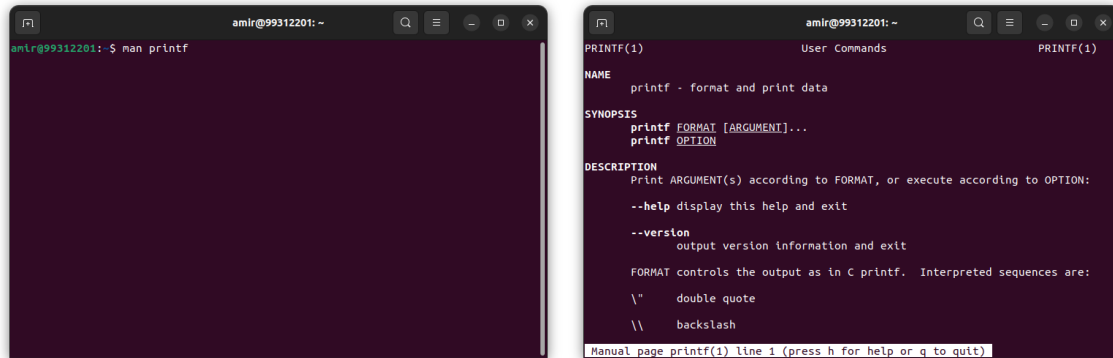


۳/۹. پس از وارد شدن با این پیام در بالای صفحه‌ی خود مواجه می‌شویم که نشان دهنده موفقیت آمیز بودن ورود به کاربر root می‌باشد.

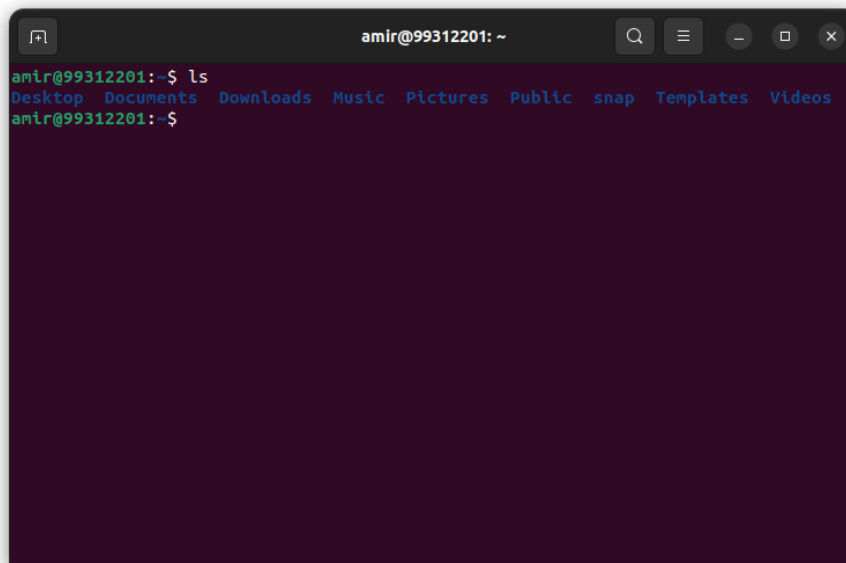


#### ٤. دستورات terminal

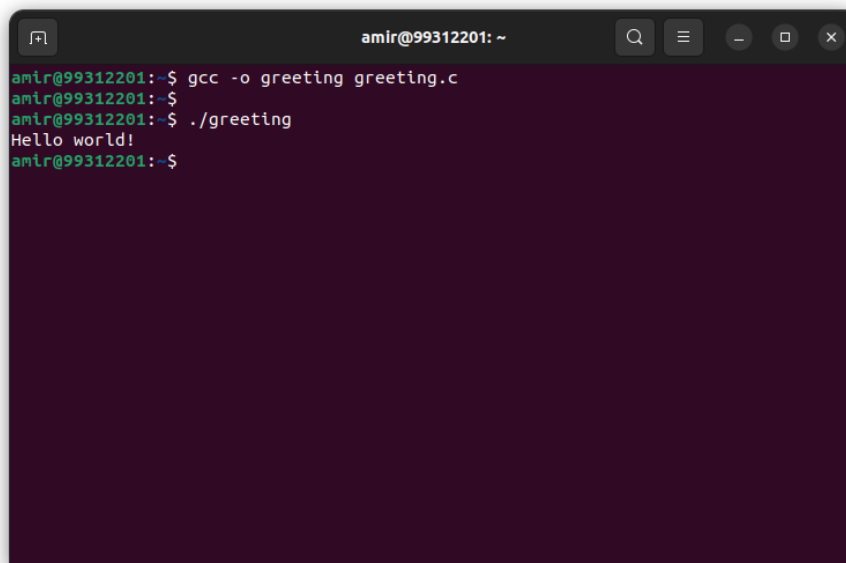
۴/۱. man: این دستور برای نمایش man page (مخفف manual page)های مختلف در سیستم عامل‌های شبیه Unix استفاده می‌شود و ابزار خوبی برای خواندن مستندات بخش‌های مختلف سیستم عامل مانند System Callها، دستورات خط فرمان و توابع استاندارد مختلف است.



۴/۲. ls: از این دستور می‌توانیم برای نمایش فایل‌های موجود در directory فعلی استفاده کرد. همچنین می‌توانیم از دستورات مشابه مانند la و ll نیز استفاده کنیم.

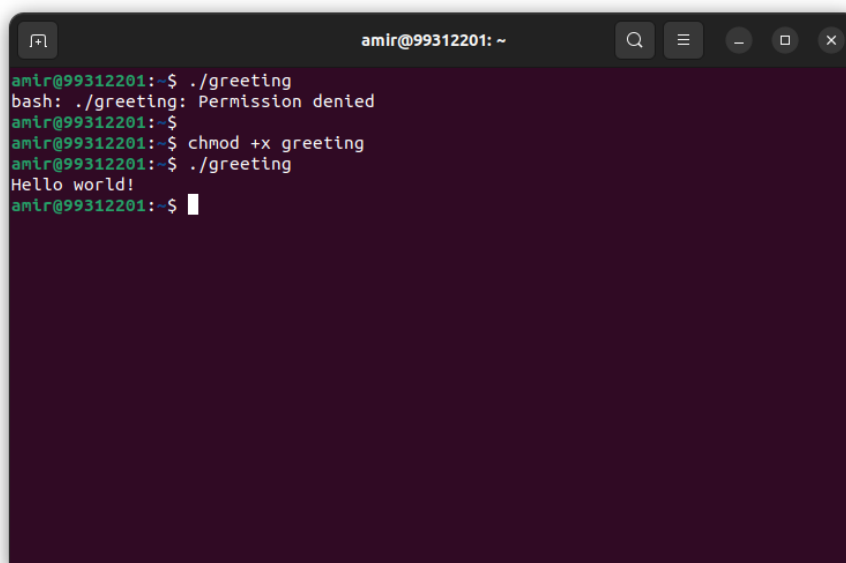


۴/۳. gcc: این دستور برای اجرای کامپایلر GNU استفاده می‌شود و می‌توان کد منبع C، C++، Fortran، Ada و غیره را با آن کامپایل کرد.



```
amir@99312201: ~  
amir@99312201:~$ gcc -o greeting greeting.c  
amir@99312201:~$  
amir@99312201:~$ ./greeting  
Hello world!  
amir@99312201:~$
```

۴/۴. chmod: (مخفف **change mode**) از این دستور می‌توانیم برای تغییر حالت دسترسی به فایل‌ها استفاده کرد.

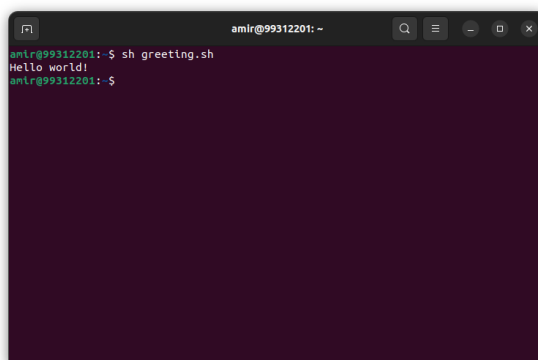


```
amir@99312201: ~  
amir@99312201:~$ ./greeting  
bash: ./greeting: Permission denied  
amir@99312201:~$  
amir@99312201:~$ chmod +x greeting  
amir@99312201:~$ ./greeting  
Hello world!  
amir@99312201:~$
```

۴/۵. sh: از این دستور می‌توان برای اجرای پوسته‌ی Bourne shell استفاده کرد و در صورت نیاز با استفاده از آن اسکریپت اجرا کرد.

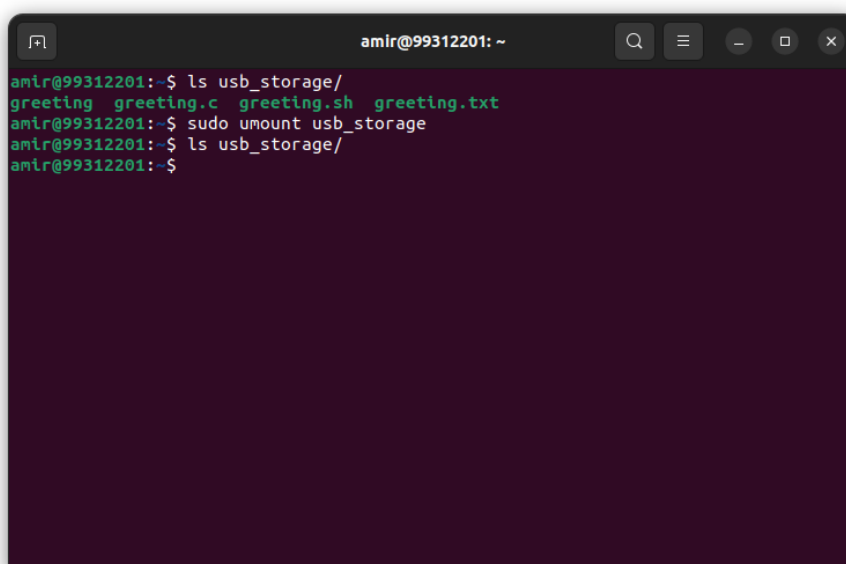


```
greeting.sh
1 echo "Hello world!"
```



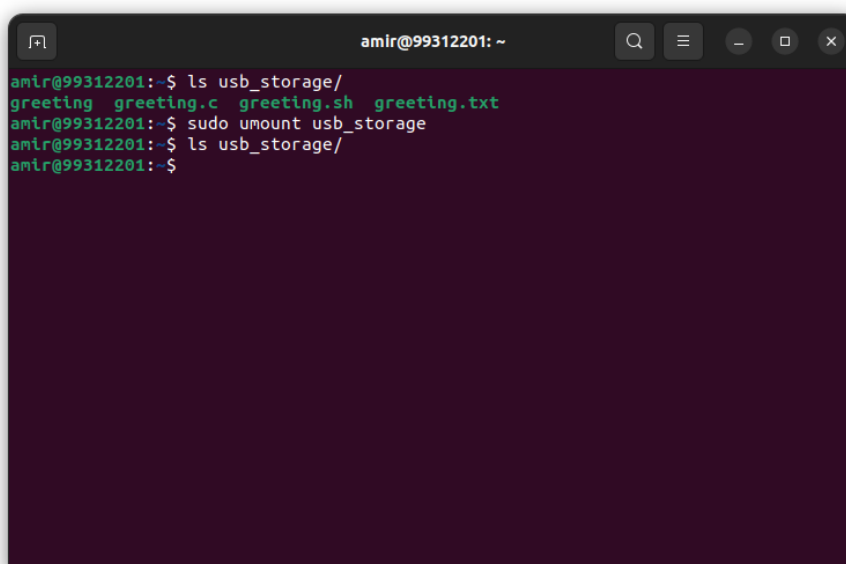
```
amir@99312201: ~
$ sh greeting.sh
Hello world!
amir@99312201: ~
```

۴/۶. mount: تمام فایل‌های قابل دسترسی در یک سیستم شبیه Unix به شکل یک مدل انتزاعی درخت با ریشه / نمایان می‌شوند. دستور mount برای اضافه کردن دستگاه‌های ذخیره سازی مختلف به این درخت و دسترسی به آنها استفاده می‌شود. برای استفاده از این دستور باید دسترسی به کاربر root داشته باشیم.



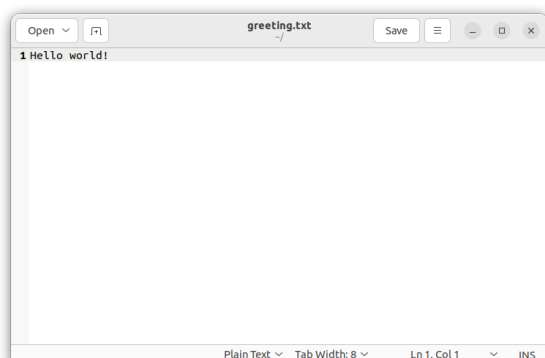
```
amir@99312201: ~
$ ls usb_storage/
greeting  greeting.c  greeting.sh  greeting.txt
$ sudo umount usb_storage
$ ls usb_storage/
$
```

۴/۷. `umount`: از این دستور می‌توان به عنوان همتای دستور `mount` استفاده کرد.

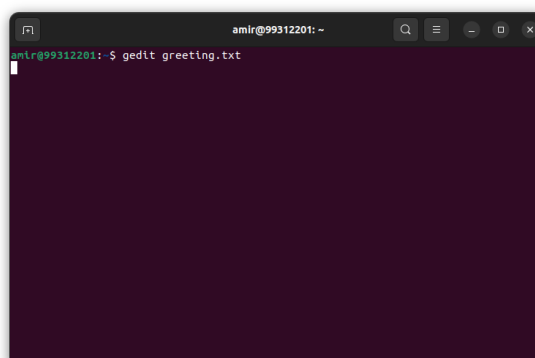


```
amir@99312201: ~  
amir@99312201:~$ ls usb_storage/  
greeting greeting.c greeting.sh greeting.txt  
amir@99312201:~$ sudo umount usb_storage  
amir@99312201:~$ ls usb_storage/  
amir@99312201:~$
```

۴/۸. `gedit`: از این دستور می‌توان برای باز کردن ویرایشگر متن GNOME استفاده کرد.



```
greeting.txt  
1 Hello world!
```



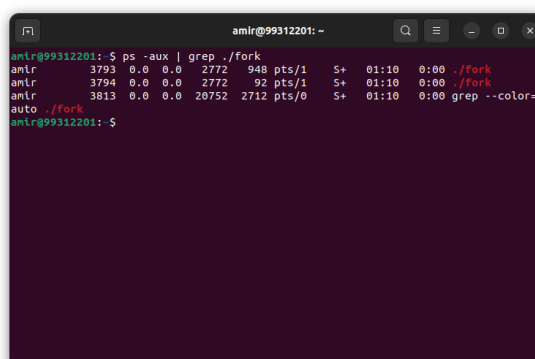
```
amir@99312201: ~  
amir@99312201:~$ gedit greeting.txt
```

## ۵. System call ها

۵/۱. `fork`: این دستور برای گرفتن یک کپی از فرایند فعلی و اجرای هر دو به صورت همزمان استفاده می‌شود.



```
fork.c  
1 #include <stdio.h>  
2 #include <unistd.h>  
3  
4 int main(void)  
5 {  
6     fork();  
7  
8     getchar();  
9 }  
10
```



```
amir@99312201:~$ ps -aux | grep ./fork  
amir 3793 0.0 0.0 2772 948 pts/1 S+ 01:10 0:00 ./fork  
amir 3794 0.0 0.0 2772 92 pts/1 S+ 01:10 0:00 ./fork  
amir 3813 0.0 0.0 20752 2712 pts/0 S+ 01:10 0:00 grep --color=  
auto ./fork  
amir@99312201:~$
```

۵/۲. `getpid` با استفاده از این دستور می‌توان شناسه (id) فرایند را در سیستم بدست آورد.

```
getpid.c
1 #include <stdio.h>
2 #include <unistd.h>
3
4 int main(void)
5 {
6     int id = getpid();
7     printf("Process ID: %d\n", id);
8 }
9
```

```
amir@99312201: ~
amir@99312201:~$ ./getpid
Process ID: 4230
amir@99312201:~$
```

۵/۳. `getppid` این دستور مشابه `getpid` است، با این تفاوت که بجای شناسه فرایند درحال اجرا شناسه والد آنرا به ما می‌دهد.

```
getppid.c
1 #include <stdio.h>
2 #include <unistd.h>
3
4 int main(void)
5 {
6     int parent_id = getppid();
7     printf("Parent ID: %d\n", parent_id);
8 }
9
```

```
amir@99312201: ~
amir@99312201:~$ ./getppid
Parent ID: 3590
amir@99312201:~$
```

## ۶. سوالات تشریحی

۶/۱. چيست GNU-Linux؟

سیستم‌عامل GNU-Linux یک سیستم‌عامل متن‌باز است که ابتدا توسط Linus Torvalds طراحی شد و نام آن از کلمات Unix + Linux برگرفته شده. این سیستم‌عامل مبتنی بر Unix می‌باشد که به دلیل پایداری و امنیت بالا اغلب در سرورها استفاده می‌شود. همچنین انعطاف‌پذیری این سیستم‌عامل استفاده از آنرا در دستگاه‌های IoT و تلفن‌های همراه که منابع سخت‌افزاری محدودی دارند را مقدور کرده.

۶/۲. تفاوت Unix و GNU-Linux:

سیستم عامل Unix سیستم عاملی چند کاربره با قابلیت multitasking است که در دهه ۱۹۶۰ در Bell Labs توسط Ken Thompson، Dennis Ritchie و دیگران توسعه یافت. بسیاری از استانداردها و فلسفه‌های استفاده شده در GNU-Linux و دیگر سیستم‌عامل‌ها (بجز ویندوز) از Unix برگرفته شده.

۶/۳. GNOME:

GNOME یک محیط گرافیکی است که توسط بسیاری از توزیع‌های لینوکس مانند Debian، Ubuntu، Fedora، و غیره استفاده می‌شود و امکان کار کردن با سیستم‌عامل بدون نیاز به نوشتن

دستورات را فراهم می‌کند. هدف از این محیط این است که افراد تازه وارد و ناآشنا با محیط لینوکس راحت‌تر بتوانند با آن ارتباط برقرار کنند و از آن استفاده کنند.