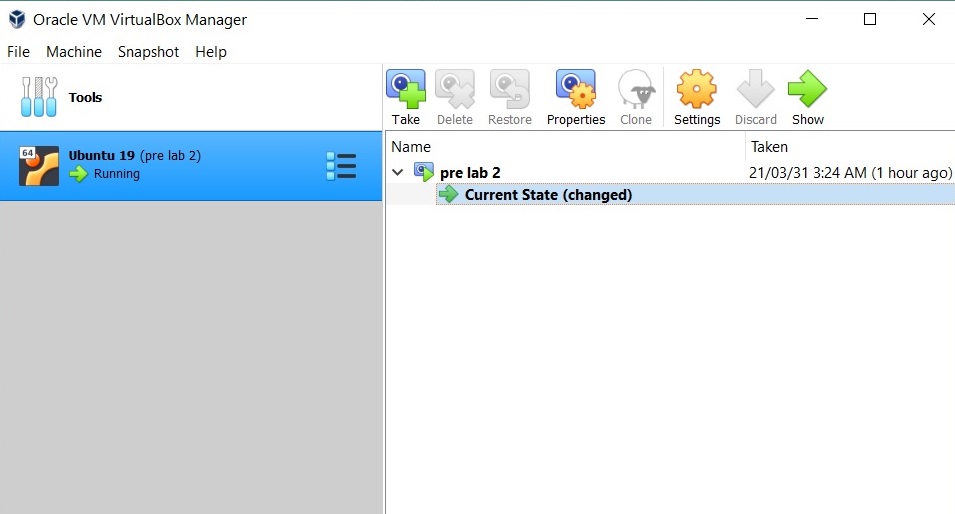
در این بخش از آزمایش قصد داریم تا یک ماژول هسته را بارگذاری و حذف کنیم و با بررسی بافر سابقه‌ی هسته مطمئن شویم که این فرآیند را به درستی طی کرده‌ایم.

در ابتدا بهتر است اشاره کنیم که ماژول‌های هسته چه هستند. Kernel Module یا ماژول‌های هسته عبارتند از تکه‌ کدهایی که در هنگام run-time می‌توانند در کرنل بارگذاری یا از آن حذف شوند و برای اجرای آن‌ها نیازی نیست که سیستم مجددا راه‌اندازی شود. این مازول‌ها، عملکرد هسته را در حین اجرا توسعه می‌دهند. ماژول‌ها می‌توانند یک سرویس، فایل سیستم یا درایور سخت‌افزاری باشند.

حال به انجام این بخش از آزمایش می‌پردازیم.

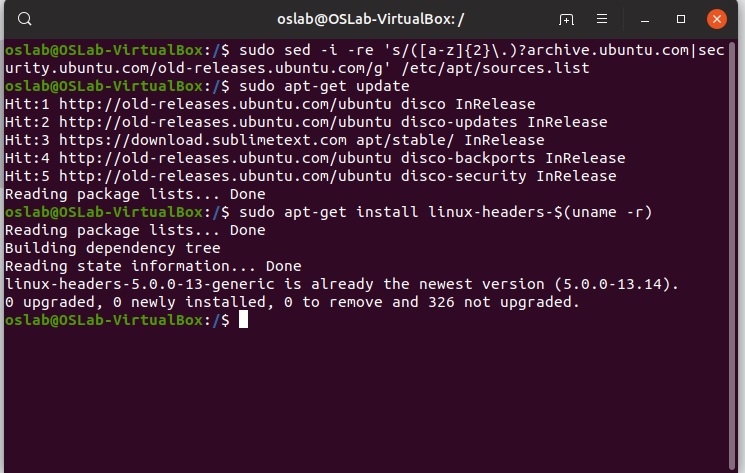
از آن‌جایی که ماژول‌های هسته به صورت مستقیم با کرنل در ارتباط هستند و هرگونه خطا و اشکال در آن‌ها می‌تواند باعث بروز اشکال در کل سیستم گردد، قبل از شروع کار از وضعیت فعلی ماشین مجازی خود snapshot می‌گیریم تا در صورت نیاز بتوانیم به شرایطی که قبل از شروع آزمایش داشتیم بازگردیم.

screen01



در مرحله‌ی بعدی برای اطمینان از اینکه هدرهای موردنیاز ما نصب شده و آپدیت هستند، مراحل زیر را طی میکنیم:

ابتدا دستور sudo sed -i -re 's/([a-z]{2}\.)?archive.ubuntu.com|security.ubuntu.com/old-releases.ubuntu.com/g' /etc/apt/sources.list و سپس به ترتیب دستورات sudo apt-get update و sudo apt-get install linux-headers-$(uname -r) را در ترمینال وارد می‌کنیم.

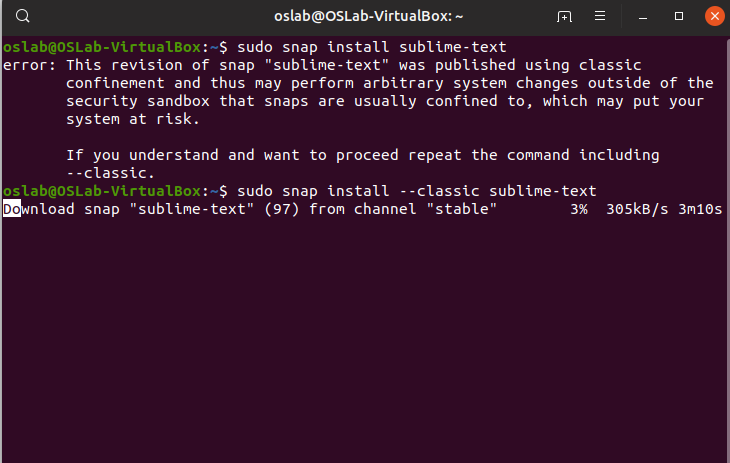


دستور sed یا stream editor دستوری است که این قابلیت را به کاربر می‌دهد تا بدون باز کردن یک فایل در ادیتور و فقط با اجرای این دستور، فایل را ادیت کند و فرآیندهایی همچون search, find, replacement, deletion, insert را به روی آن فایل اجرا کند.

از دستور apt-get برای مدیریت پکیج‌ها به واسطه‌ی کار با بخش APT (Advanced Packaging Tool) در لینوکس، استفاده می‌شود. از این دستور برای حذف، نصب و به روزرسانی پکیج‌ها و نرم‌افزارها استفاده می‌شود.

بعد از اینکه مطمئن شدیم هدرهای موردنظر ما نصب هستند، ادیتور sublime-text را با استفاده از دستور sudo snap install sublime-text نصب میکنیم تا بتوانیم ادیت فایلهای خود را در محیطی کارآمدتر انجام دهیم.

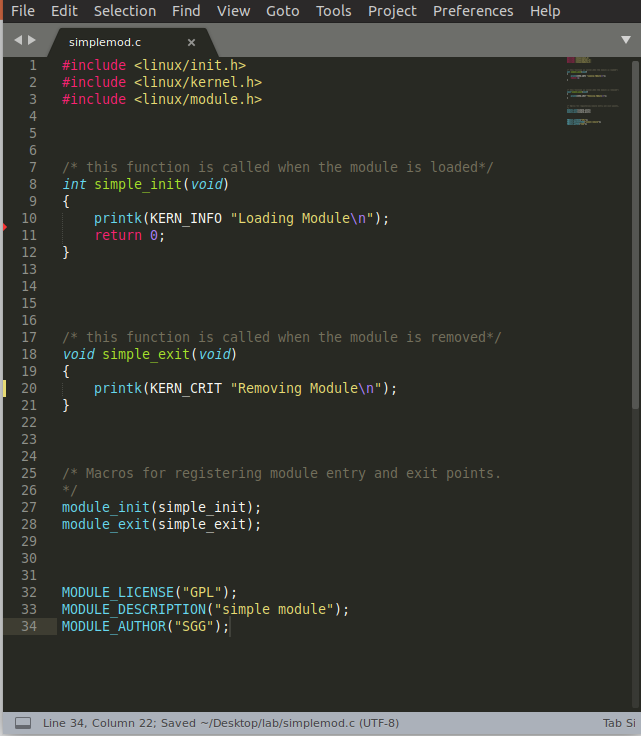
screen03



snap یک سیستم مدیریت پکیج در لینوکس است که به واسطه‌ی آن می‌توان به راحتی بسیاری از برنامه‌هایی را که در این سیستم پکیج شده‌اند را دانلود و نصب کرد. این سیستم از Ubuntu 16.04 LTS به بعد روی تمام ورژن‌های اوبونتو نصب است و تمام برنامه‌‌هایی که با استفاده از snap نصب شوند به صورت اتوماتیک آپدیت خواهند شد.

پس از نصب، sublime-text را باز میکنیم تا کد مربوط به ماژول خود را در آن وارد و ادیت کنیم. (از آنجایی که کد این ماژول را آماده داریم آن را کپی میکنیم و در فرمت یک فایل .c ذخیره می‌کنیم)

screen04



**این ماکروها نیز حاوی اطلاعات و مشخصات مربوط به ماژول می‌باشند**

این دو ماکرو محل ورود و خروج به ماژول Simplemod را ثبت میکنند و در محل ورود و خروج، توابع simple\_init و simple\_exit را فراخوانی می‌کنند.

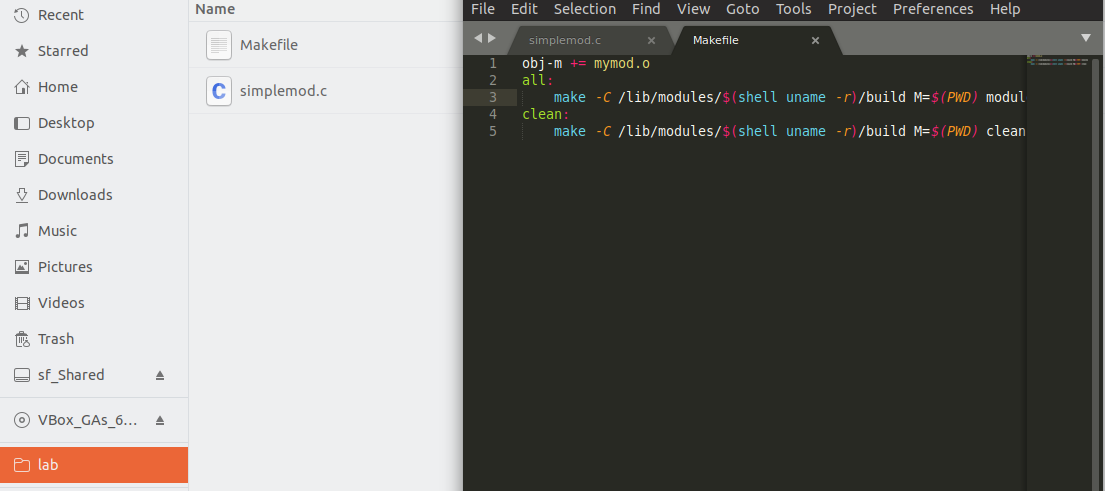
**این عبارت، نشان‌دهنده‌ی حالت و اهمیت پیام چاپ شده است. طبق جدول اولویت‌ها kern\_info نشان دهنده‌ی یک پیام عادی است اما kern\_crit یک پیام critical است که اولویت بالاتری دارد و در صورت مشاهده باید فورا به آن رسیدگی شود. در اینجا این عبارت را تغییر دادیم تا ماژول ما چند حالت متفاوت را در بگیرد. به همین دلیل است که پیام removing module به رنگ قرمز چاپ خواهد شد.**

**دستور printk معادل دستور printf است با این تفاوت که پیام خود را در بافر سابقه هسته چاپ می‌کند.**

فایل بدون پسوند دیگری شامل قوانین مربوط به کامپایل simplemod.c را در همین دایرکتوری ایجاد می‌کنیم و نام آن را Makefile می‌گذاریم. (به دلیل تغییر دادن نام ماژول نیاز است تا در کد آماده‌ای که برای makefile در اختیار داریم نیز تغییر ایحاد کنیم)

هرگاه که میخواهیم از دستور make استفاده کنیم به یک makefile نیز نیاز است که ارتباط فایلهای درون برنامه‌ی ما را توضیح میدهد.

screen05

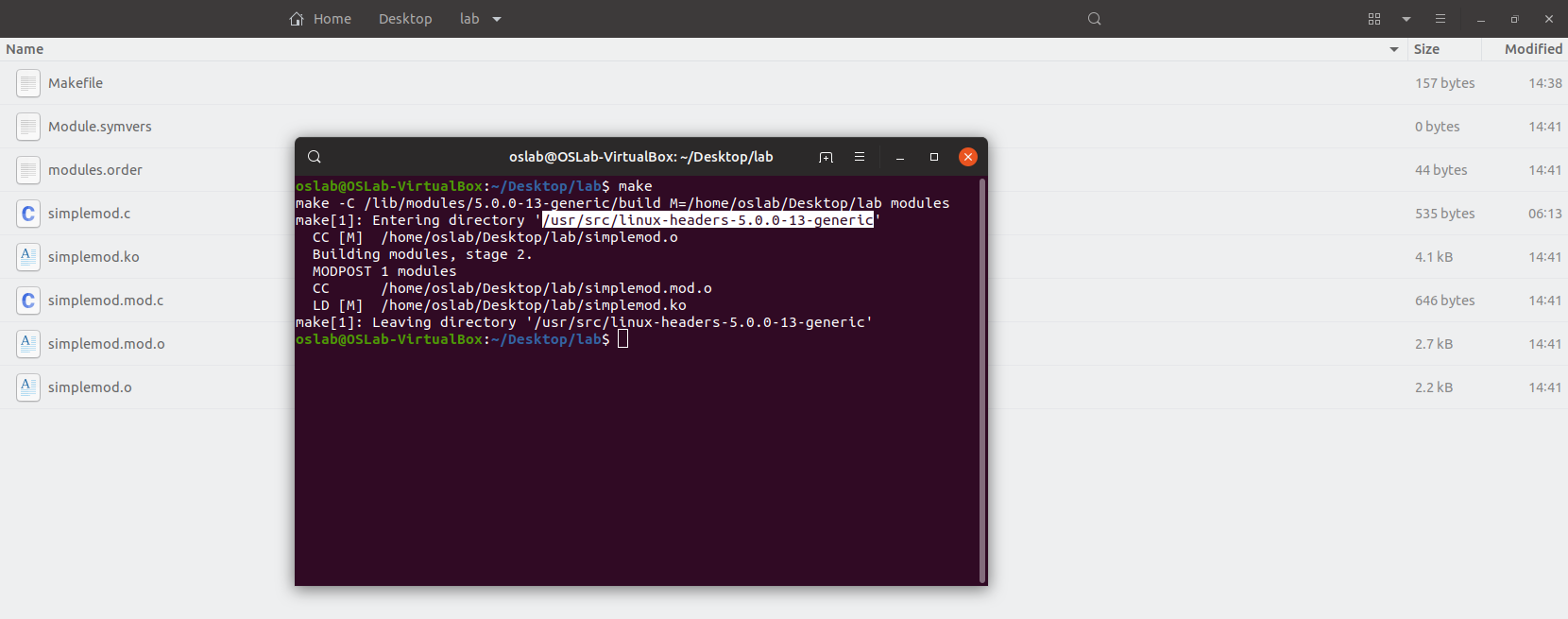


simplemod.o

حالا در همین دایرکتوری فعلی (Desktop/lab) که فایلهای ما درون آن قرار دارند، ترمینال را باز میکنیم و دستور make را وارد می کنیم. از این دستور برای کامپایل کردن ماژول‌ها و پروژه‌های بزرگ استفاده می‌شود. پس از کامپایل، فایل simplemod.ko در همین دایرکتوری ایجاد می‌شود. پسوند .ko برای ماژول‌های هسته در نظر گرفته می‌شود.

در همین مرحله، آدرس entering directory را در جایی سیو میکنیم چون برای مراحل بعدی به آن نیاز داریم.

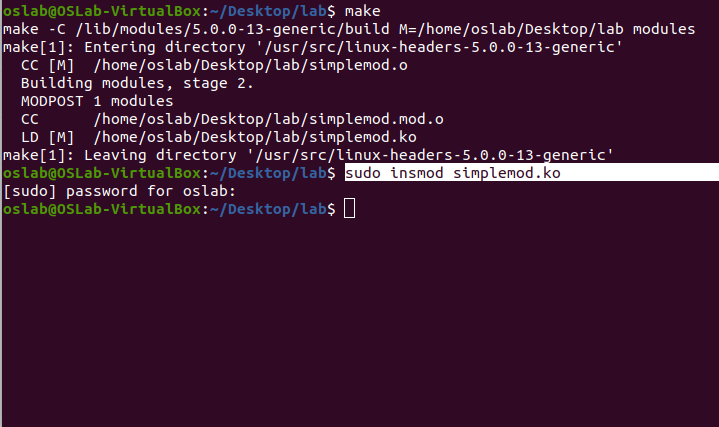
screen06



حالا که ماژول هسته‌ی ما ساخته شده، می‌توانیم آن را به کرنل اضافه و از آن حذف کنیم. برای اینکه به این فرآیند نظارت داشته باشیم، تب جدیدی در ترمینال باز میکنیم و دستور tail –f /var/log/syslog را وارد می کنیم تا لاگ‌های سیستم را زیرنظر بگیریم و ببینیم که آیا خروجی دلخواه خود را می گیریم یا خیر (برای همین منظور می‌توان از دستور dmesg استفاده کرد که محتویات بافر سایقه هسته را چاپ می‌کند) طبق کد simplemod.c، انتظار داریم که با لود کردن ماژول پیام Loading Module و با حذف آن پیام Removing Module چاپ شوند.

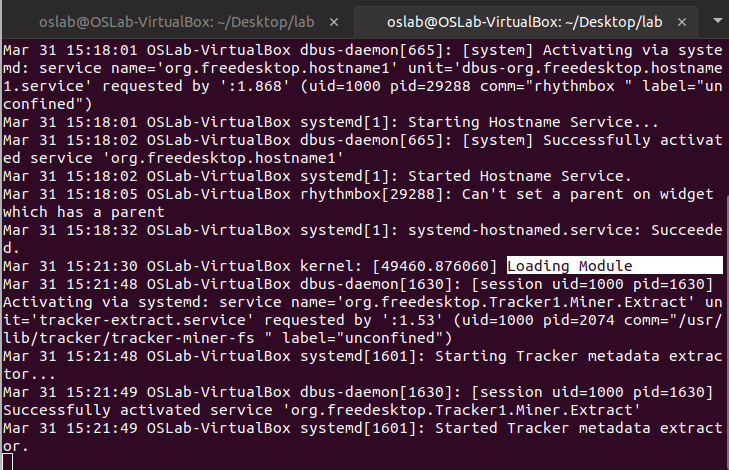
حالا با استفاده از دستور sudo insmod simplemod.ko این ماژول را در کرنل لود میکنیم.

screen07



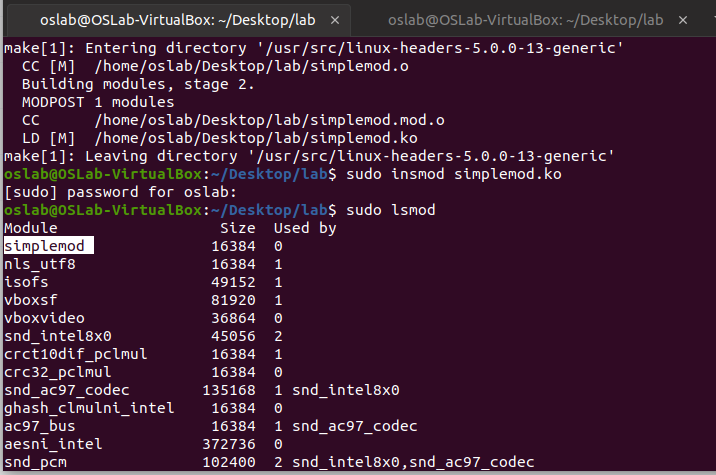
حالا اگر به تب دوم که دستور tail را در آن اجرا کرده بودیم نگاه کنیم، می‌بینیم که عبارت “loading module” طبق انتظار ما چاپ شده است

screen08



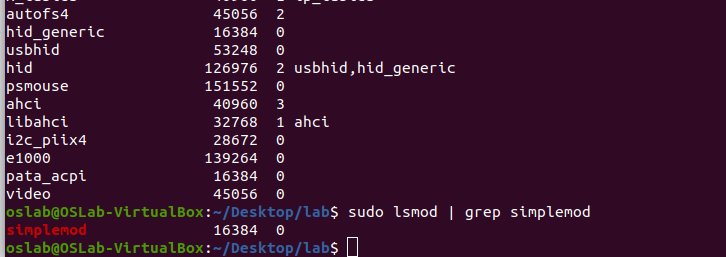
در این حالت اگر دستور sudo lsmod را وارد کنیم که ماژول‌های فعلی هسته را لیست می‌کند، می‌توانیم ماژول خود را میان آن‌ها ببینیم.

screen09



با دستور sudo lsmode | grep [module name] میتوانیم به دنبال یک ماژول خاص که در حال حاضر در هسته لود شده است بگردیم، اگر این دستور را برای simplemod اجرا کنیم، آن را می‌یابیم.

screen10



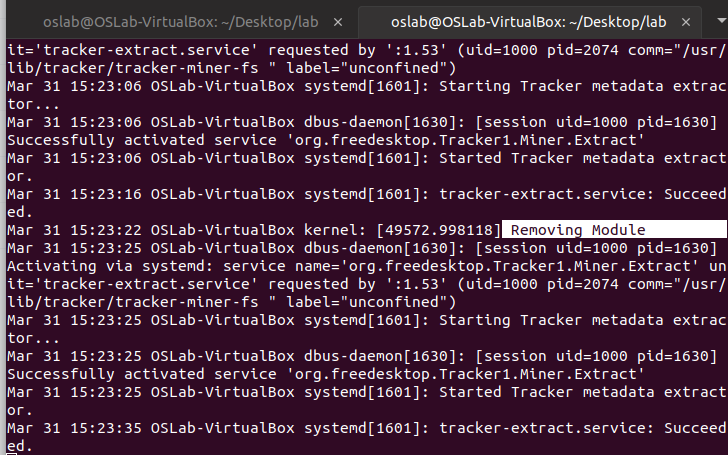
در این مرحله قصد داریم ماژولی که بارگذاری کردیم را از کرنل حذف کنیم. برای اینکار از دستور sudo rmmod [module name] استفاده میکنیم.

screen11



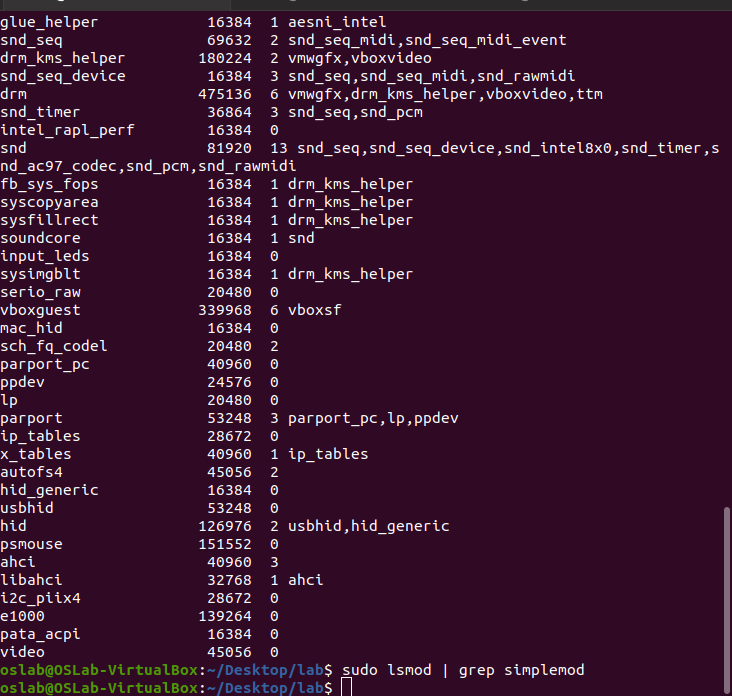
پس از اجرای این دستور در تب دوم می‌بینیم که عبارت removing module چاپ شده است

screen12



بعد از حذف کردن simplemod اگر از ماژول‌های هسته لیست بگیریم یا مجددا به دنبال simplemod بگردیم، آن را نخواهیم یافت.

screen13



همانطور که گفتیم با دستور dmesg نیز می‌توانیم پیام‌های بافر سابقه‌ی هسته را رویت کنیم. دلیل قرمز بودن عبارت remove را در بخش توضیحات کد شرح دادیم.

screen14

