به نام ایزد یکتا

گزارش دوم آزمایشگاه سیستم‌عامل

­­

استاد: مهندس قاسمی

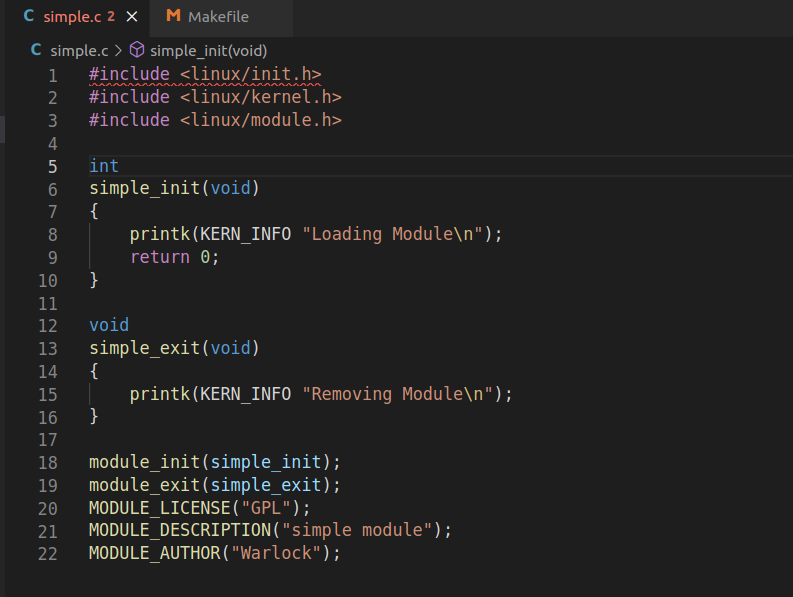
تهیه کننده: بردیا اردکانیان

۹۸۳۱۰۷۲

**بخش اول**

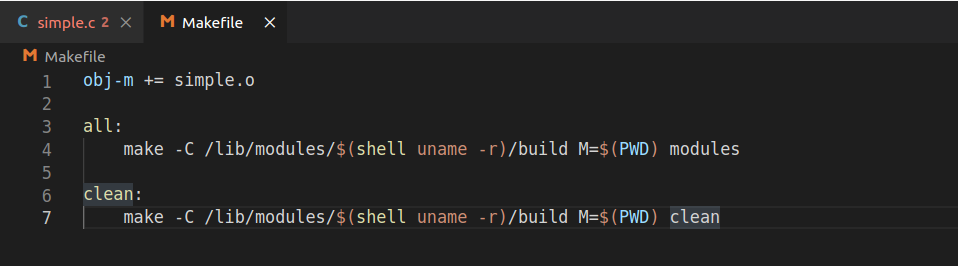
الف) ابتدا کد مثال زده شده در دستور کار – مثالی بسیار ساده از یک ماژول هسته – با نام simple.c می‌نویسیم.

کد ذکر شده در عکس 1-1 آمده است.



عکس 1-1

ب) یک فایل Makefile جهت کامپایل کردن ماژول می‌نویسیم. لطفا به عکس 1-2 مراجعه کنید.



عکس 1-2

**چرا به Makefile نیاز داریم؟**

کتابخانه‌های استفاده شده در ابتدا کد مانند linux/init.h، linux/kernel.h و linux/module.h در gcc موجود نمی‌باشند. به همین جهت می‌بایست ابتدا هدر‌های کرنل لینوکس را دانلود کرده و بعد با کمک Makefile از آنها در کامپایل شدن کد خود استفاده کنیم.

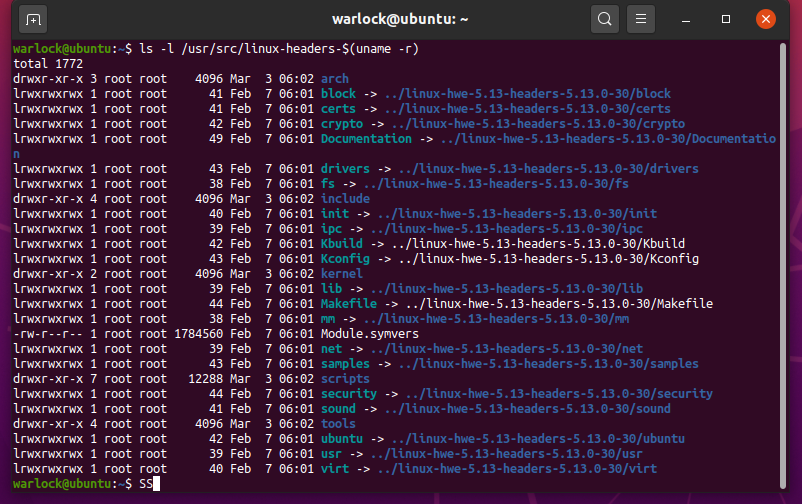
جهت دانلود کردن هدر‌های کرنل لینوکس می‌توان از کامند زیر استفاده کرد:

sudo apt install linux-headers-$(uname -r)

همچنین برای اینکه مشخص شود دانلود موفقیت آمیر بوده می‌توان از کامند زیر استفاده کرد:

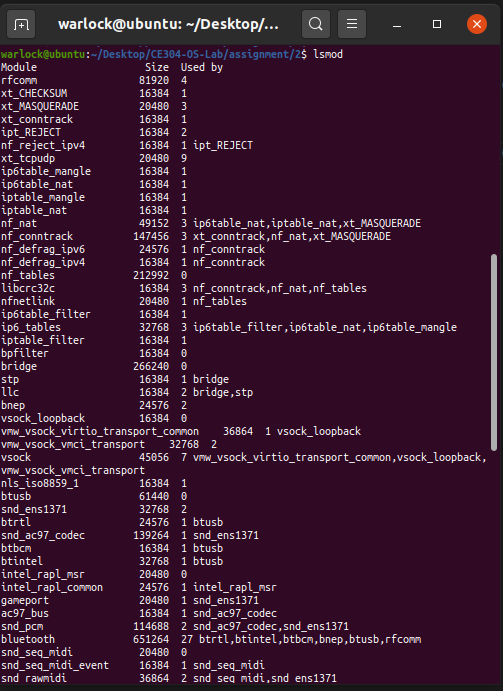
ls -l /usr/src/linux-headers-$(uname -r)

بعد از اجرای دستور بالا می‌بایست خروجی مشابه عکس 1-3 داشته باشید.



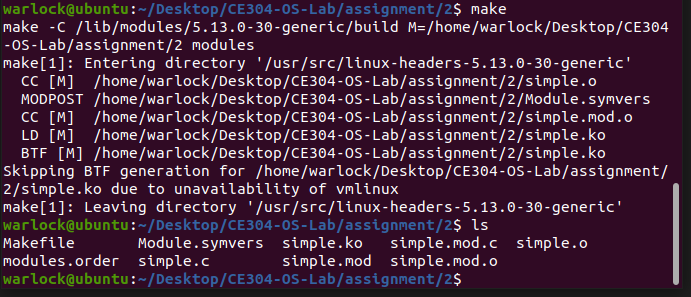
عکس 1-3

پ) با استفاده از دستور lsmod تمامی ماژول‌های موجود در هسته لینوکس را نمایش می‌دهیم. با توجه به اینکه هنوز ماژول simple را اضافه نکرده‌ایم در لیست (عکس 1-4) قابل مشاهده نمی‌باشد.



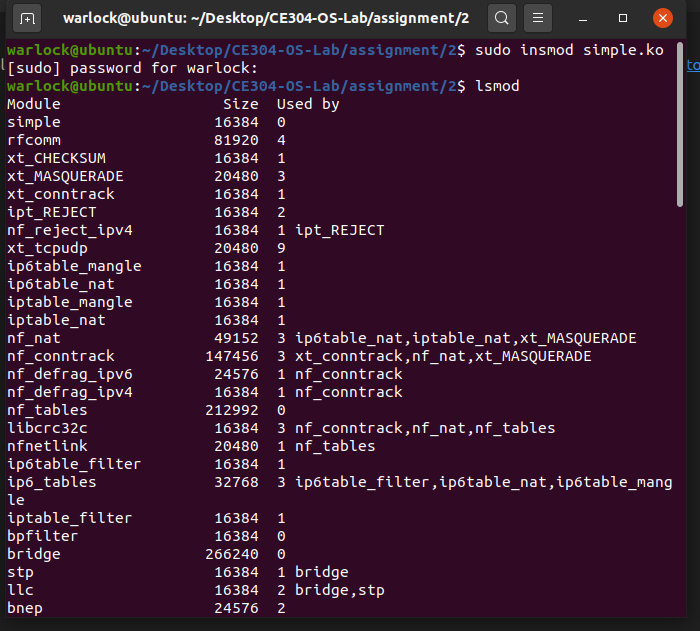
عکس 1-4

ت) با استفاده از دستور make ماژول نوشته شده را کامپایل می‌کنیم و یک فایل .ko (عکس 1-5) ساخته می‌شود.



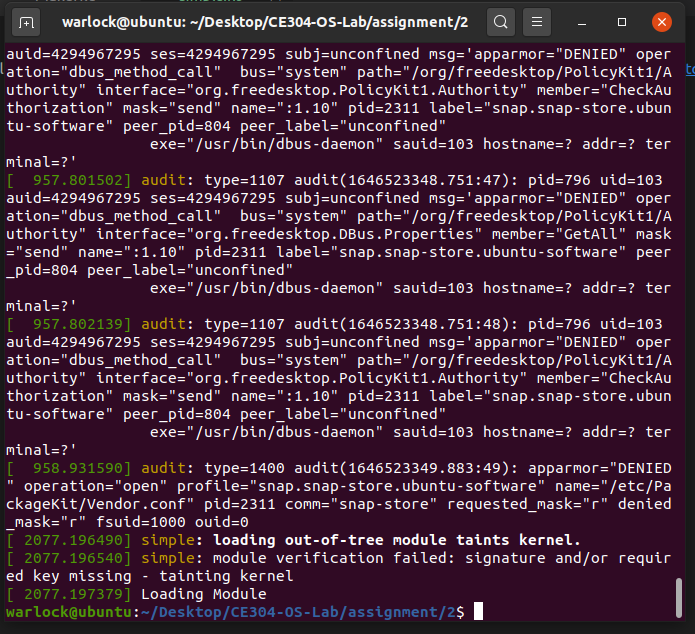
عکس 1-5

ج) در نهایت با کمک دستور insmod ماژول را در هسته لینوکس بارگذاری می‌کنیم. با تکرار دستور lsmod می‌توان ماژول simple را در میان ماژول‌های هسته مشاهده کرد (عکس 1-6).



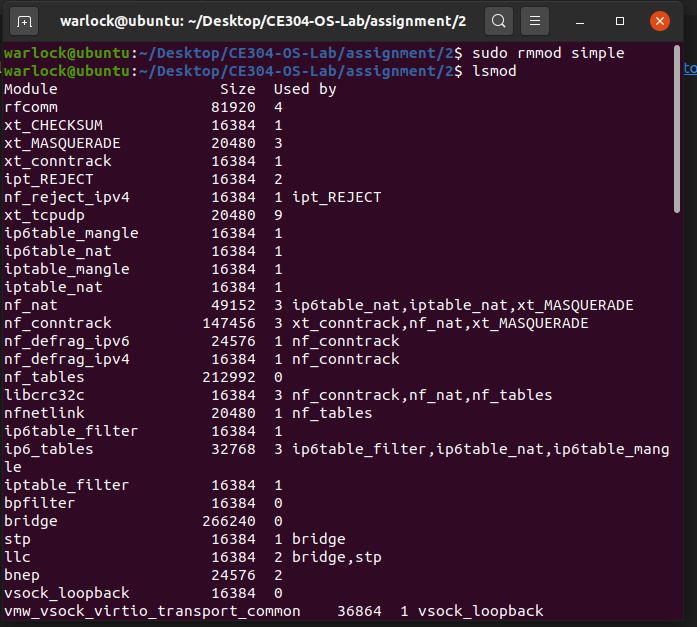
عکس 1-6

دستور dmesg محتوای بافر را برای ما به نمایش می‌گذارد و همانطور که انتظار می‌رفت در عکس 1-7 پیام Loading Module قابل مشاهده است.

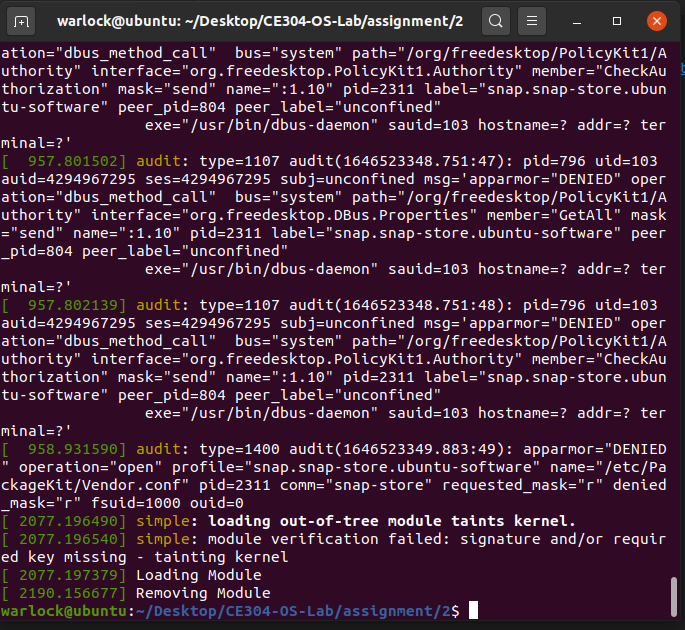


عکس 1-7

با استفاده ار دستور rmmod ماژول ساخته شده را حذف می‌کنیم (عکس 1-8-1) و سپس با dmesg محتویات بافر را به نمایش می‌گذاریم (عکس 1-8-2)

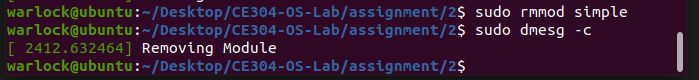


عکس 1-8-1



عکس 1-8-2

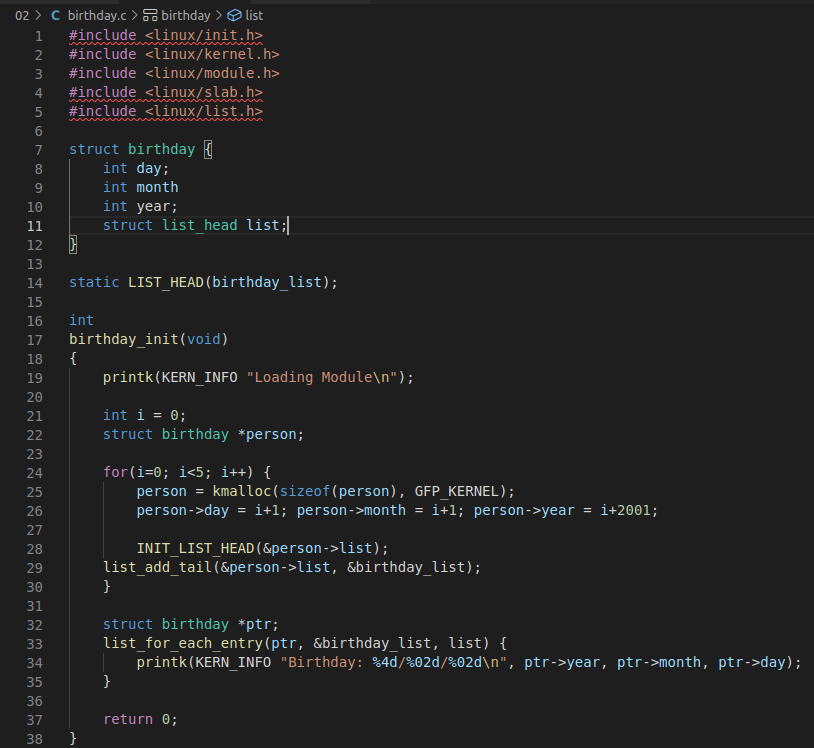
همچنین اگر قبل از rmmod از دستور sudo dmesg -c استفاده کنیم بافر ما را خالی (clear) می‌کند و تنها پیامی که هنگام حذف شدن ماژول به کار رفته است را نمایش می‌دهد. (عکس 1-9)

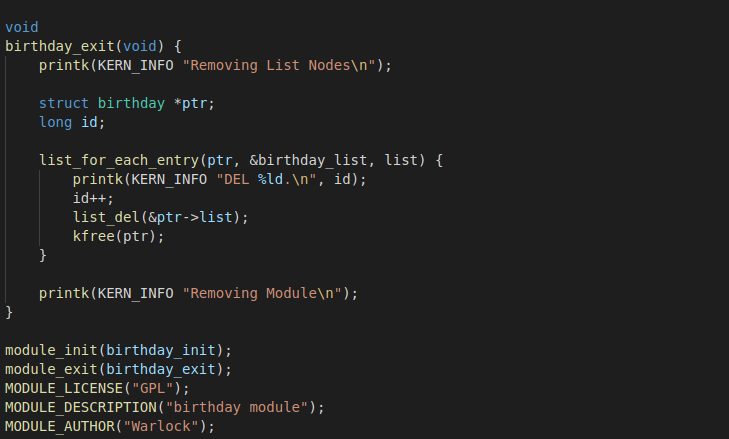


عکس 1-9

**بخش دوم**

ابتدا Struct birthday را مطابق دستور کار می‌سازیم. این استراکت در واقع ساختار هر عنصر از لیست ما می‌باشد. با استفاده ار ماکروی LIST\_HEAD، پوینتر به ابتدا لیست را مشخص کنیم. و در نهایت توابع init, exit را پیاده‌سازی می‌کنیم. (کد در عکس‌های زیر قابل مشاهده است)

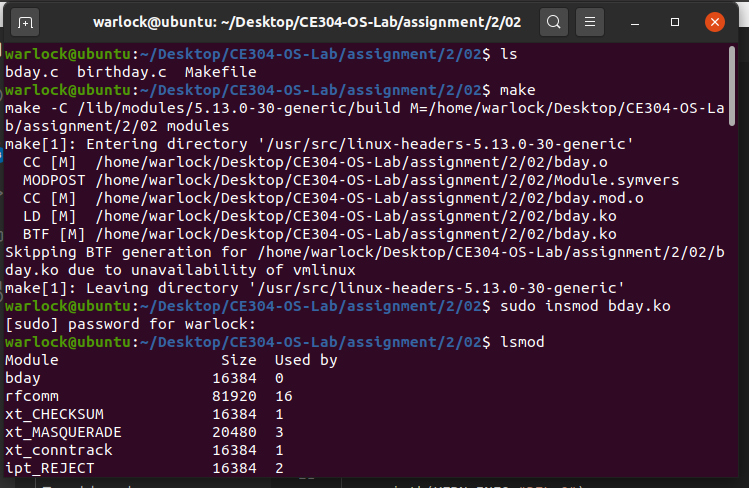




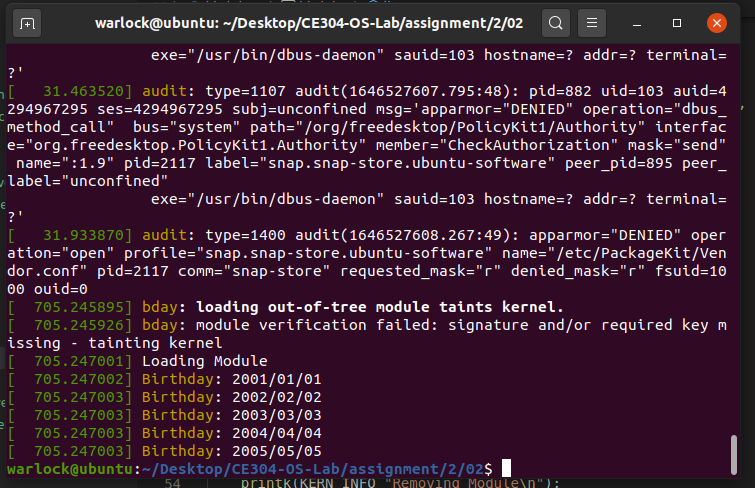
طبق خواسته سوال یک لیست متشکل از 5 عضو ساخته می‌شود (به انها با استفاده از kmalloc حافظه اختصاص داده می‌شود و با کمک list\_add\_tail به انتها لیست اضافه می‌شوند).

در مرحله بعد به کمک list\_for\_each\_entry بر روی لیست پیمایش می‌شود و اطلاعات اعضا آن به بافر اضافه می‌شود.

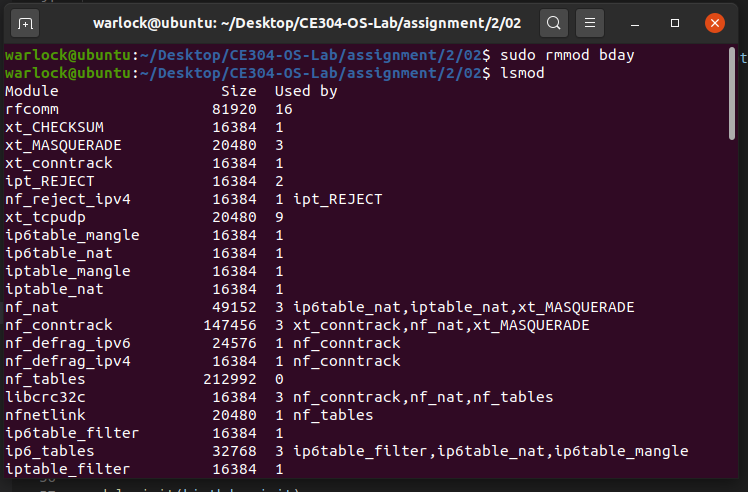
در نهایت با کمک تابع گفته شده بر روی لیست پیمایش می‌شود و حافظه اعضا آن ازاد می‌شود تا دوباره در جایی دیگر قابل استفاده باشند. پیامی مناسب نیست به بافر اضافه می‌شود. به عکس‌های 2-1 و 2-2 و 2-3 و 2-4 توجه کنید.



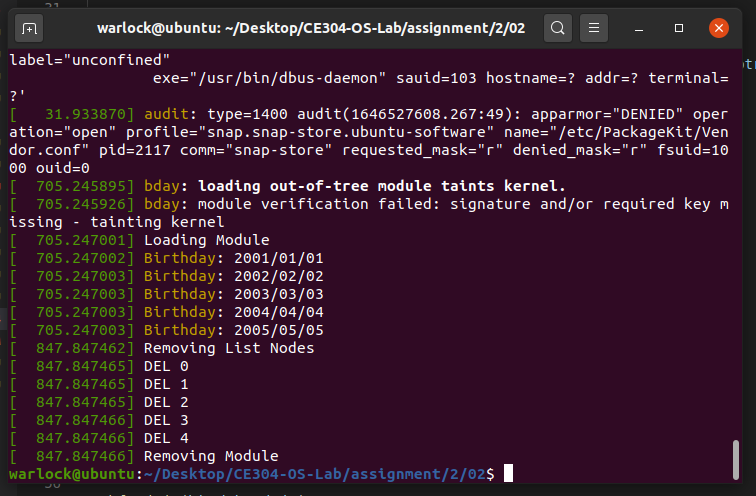
عکس 2-1



عکس 2-2



عکس 2-3



عکس 2-4