

بسمه تعالی

گزارش کار آزمایشگاه سیستم عامل

استاد مربوطه:

مهندس اوا انوری

آزمایش دوم:

Kernel module programming

فروغ افخمی 9831703

پاییز 1401

تمرین 1: مراحل بالا را دنبال کنید تا یک ماژول هسته را ایجاد، بارگذاری و بردارید. ضمن بررسی محتوی بافر سابقه هسته مطمئن شوید مراحل کار را به درستی انجام داده اید.

ابتدا فایل mymod5.c را مینویسیم که به صورت زیر است. کد سی هست در نتیجه با پسوند C. داریم.

```
#include<linux/init.h>
#include <linux/kernel.h>
#include <linux/module.h>
/* this function is called when the module is loaded*/
int simple_init(void) {

    printk(KERN_INFO "Loading Module\n" );
    return 0; }
/* this function is called when the module is removed*/
void simple_exit(void){
    printk(KERN_INFO "Removing Module\n"); } /* Macros for registering module entry and exit points.*/
module_init(simple_init);
module_exit(simple_exit);
MODULE_LICENSE("GPL");
MODULE_DESCRIPTION("simple module");
MODULE_AUTHOR("SGG");
```

در این کد ما یک تابع simple_init داریم که در آن یک printk داریم که همان printf هست که در مود کرنل استفاده میشود و الویت پیامی که قرار است نشان دهد را میگیرد که از جنس Information است و الویت بالایی ندارد و محتوایی که قرار است نمایش بدهد در آن قرار گرفته است.

یک تابع دیگر به اسم simple_exit داریم که آن هم یک printk و KERN_INFO دارد. در ادامه دو macro داریم که module_init و module_exit می باشد. در module_init هنگامی که ماژول دارد مقدار دهی اولیه میشود و در هسته بارگزاری میشود تابع simple_init را فراخوانی میکند. در module_exit وقتی ماژول قرار است از کرنل حذف شود هم تابع simple_exit فراخوانی میشود. دستورات بعدی هم یک سری اطلاعات در رابطه با ماژول است که با دستور modinfo میتوان این اطلاعات مثل ادرس بارگزاری ماژول، مجوز و ... را مشاهده کرد.

در رابطه با اینکه الویت KERN_INFO کم هست در جدول زیر الویت ها از پایین به بالا بیشتر میشود.

Name	String	Alias function
KERN_EMERG	"0"	pr_emerg()
KERN_ALERT	"1"	pr_alert()
KERN_CRIT	"2"	pr_crit()
KERN_ERR	"3"	pr_err()
KERN_WARNING	"4"	pr_warn()
KERN_NOTICE	"5"	pr_notice()
KERN_INFO	"6"	pr_info()
KERN_DEBUG	"7"	pr_debug() and pr_devel() if DEBUG is defined
KERN_DEFAULT	""	
KERN_CONT	"c"	pr_cont()

برای کامپایل کردن فایل c لازم است یک Makefile ایجاد کنیم که به صورت زیر است. دقت شود در خط اول اسم فایل o. باید با اسم فایل c. ما یکی باشد. در همان دایرکتوری که فایل c. را ذخیره کردیم این فایل را هم با نام Makefile ذخیره میکنیم.

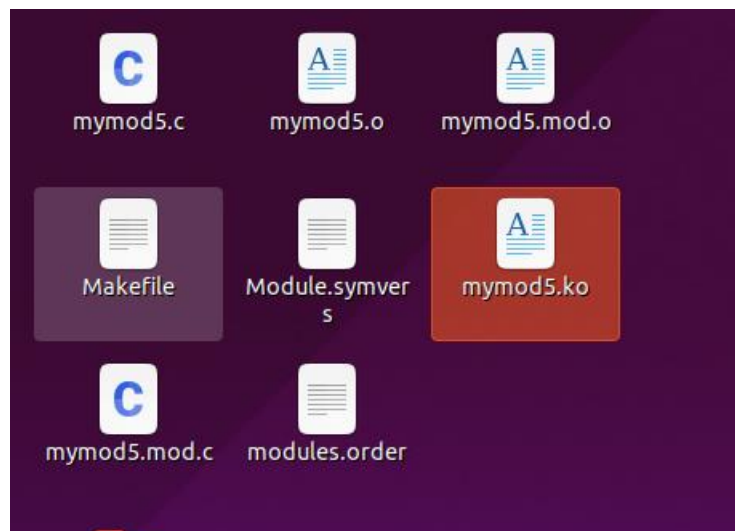
```

                                *mymod5.c
obj-m += mymod5.o
all:
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules
clean:
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean

```

حال با داشتن این دو فایل میتوان فایل را کامپایل کرد. سپس در ترمینال ابتدا وارد دایرکتوری میشویم که فایل ها وجود دارند و با دستور make در terminal یک سری فایل برای ما ایجاد میشود. در بین این فایل ها یک فایل mymod5.ko با نام همان فایل خود داریم که در واقع فایل ماژول ما با پسوند ko. می باشد.

```
zk@zk-PC: ~/Desktop
zk@zk-PC:~$ cd Desktop
zk@zk-PC:~/Desktop$ make
make -C /lib/modules/5.0.0-13-generic/build M=/home/zk/Desktop modules
make[1]: Entering directory '/usr/src/linux-headers-5.0.0-13-generic'
CC [M] /home/zk/Desktop/mymod5.o
Building modules, stage 2.
MODPOST 1 modules
CC /home/zk/Desktop/mymod5.mod.o
LD [M] /home/zk/Desktop/mymod5.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.0.0-13-generic'
zk@zk-PC:~/Desktop$
```



حال این ماژول را با دستور `sudo insmod mymod5.ko` در کرنل بارگزاری میکنیم. برای تشخیص اینکه آیا کار ما درست انجام شده است با دستور `lsmod` میتوان دید که آیا ماژول ما در لیست ماژول ها وجود دارد یا خیر.

همان طور که در تصویر زیر مشاهده میکنید این ماژول در لیست ماژول های هسته وجود دارد.

```

zk@zk-PC: ~/Desktop
zk@zk-PC:~/Desktop$ make
make -C /lib/modules/5.0.0-13-generic/build M=/home/zk/Desktop modules
make[1]: Entering directory '/usr/src/linux-headers-5.0.0-13-generic'
CC [M] /home/zk/Desktop/mymod5.o
Building modules, stage 2.
MODPOST 1 modules
CC /home/zk/Desktop/mymod5.mod.o
LD [M] /home/zk/Desktop/mymod5.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.0.0-13-generic'
zk@zk-PC:~/Desktop$ sudo insmod mymod5.ko
[sudo] password for zk:
zk@zk-PC:~/Desktop$ lsmod
Module                  Size  Used by
mymod5                  16384  0
nls_utf8                16384  1
isofs                   49152  1
vboxvideo               36864  0
snd_intel8x0            45056  2
snd_ac97_codec          135168  1 snd_intel8x0
ac97_bus                16384  1 snd_ac97_codec
snd_pcm                 102400  2 snd_intel8x0,snd_ac97_codec
snd_seq_midi            20480  0
snd_seq_midi_event      16384  1 snd_seq_midi
snd_rawmidi             36864  1 snd_seq_midi

```

میتوان برای اینکار از یک روش دیگر استفاده کرد. ما گفته بودیم در صورت بارگزاری ماژول “loading module” چاپ شود. که این چاپ شدن در بافر سابقه هسته اتفاق می افتد. برای دسترسی به بافر سابقه هسته از دستور dmesg استفاده میکنیم که مبینیم loading module اتفاق افتاده است.

```

zk@zk-PC:~/Desktop$ dmesg
[    0.000000] Linux version 5.0.0-13-generic (build@lgcy01-andd4-020) (gcc version 8.3.0 (Ubuntu 8.3.0-6ubuntu1)) #14-Ubuntu SMP Mon Apr 15 14:59:14 UTC 2019 (Ubuntu 5.0.0-13.14-generic 5.0.6)
[    0.000000] Command line: BOOT_IMAGE=/boot/vmlinuz-5.0.0-13-generic root=UUID=896f5d99-8925-4f65-a74d-6031a8d38886 ro quiet splash
[    0.000000] KERNEL supported cpus:
[    0.000000] Intel GenuineIntel
[    0.000000] AMD AuthenticAMD
[    0.000000] Hygon HygonGenuine
[    0.000000] Centaur CentaurHauls
[    0.000000] x86/fpu: x87 FPU will use FXSAVE
[    0.000000] BIOS-provided physical RAM map:
[    0.000000] BIOS-e820: [mem 0x0000000000000000-0x000000000009fbff] usable
[    0.000000] BIOS-e820: [mem 0x000000000009fc00-0x000000000009ffff] reserved
[    0.000000] BIOS-e820: [mem 0x00000000000a0000-0x00000000000fffff] reserved
[    0.000000] BIOS-e820: [mem 0x0000000000100000-0x0000000000dfffff] usable
[    0.000000] BIOS-e820: [mem 0x0000000000df0000-0x0000000000dfffff] ACPI data
[    0.000000] BIOS-e820: [mem 0x0000000000fec00000-0x0000000000fec0ffff] reserved
[    0.000000] BIOS-e820: [mem 0x0000000000fec00000-0x0000000000fec0ffff] reserved
[    0.000000] BIOS-e820: [mem 0x0000000000fee00000-0x0000000000fee0ffff] reserved
[    0.000000] BIOS-e820: [mem 0x0000000000fffc0000-0x0000000000ffffff] reserved
[    0.000000] BIOS-e820: [mem 0x000000001000000000-0x0000000015ffffff] usable
[    0.000000] NX (Execute Disable) protection: active
[    0.000000] SMBIOS 2.5 present.
[    0.000000] DMI: Innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006
[    0.000000] Hypervisor detected: KVM
[    0.000000] kvm-clock: Using msrc 4b564d01 and 4b564d00
[    0.000001] kvm-clock: cpu 0, msrc 20010001, primary cpu clock
[    0.000001] kvm-clock: using sched offset of 9274948422 cycles
[    0.000002] clocksource: kvm-clock: mask: 0xffffffffffffffff max_cycles: 0x1cd42e4dffb, max_idle_ns: 881590591483 ns
[    0.000004] tsc: Detected 1991.998 MHz processor
[    0.004782] e820: update [mem 0x00000000-0x000000fff] usable ==> reserved
[    0.004783] e820: remove [mem 0x00000000-0x000000fff] usable ==> reserved
[    0.004787] last_pfn = 0x100000 max_arch_pfn = 0x400000000
[    0.004830] MTRR default type: uncachable
[    0.004839] MTRR fixed ranges disabled:
[    0.004840] 00000-fffff uncachable
[    0.004841] MTRR variable ranges disabled:
[    0.004841] 0 disabled
[    0.004842] 1 disabled
[    0.004842] 2 disabled
[    0.004842] 3 disabled
[    0.004842] 4 disabled
[    0.004843] 5 disabled
[    0.004843] 6 disabled
[    0.004843] 7 disabled
[    0.004846] Disabled
[    0.004848] x86/PAT: MTRRs disabled, skipping PAT initialization too.
[    0.004852] CPU MTRRs all blank - virtualized system.
[    0.004853] x86/PAT: Configuration [0-7]: WB WT UC- UC- WB WT UC- UC-

```

```

zk@zk-PC: ~/Desktop
[ 10.300390] [drm] GMR.
[ 10.300390] [drm] Traces.
[ 10.300391] [drm] GMR2.
[ 10.300391] [drm] Screen Object 2.
[ 10.300392] [drm] Max GMR lds is 8192
[ 10.300393] [drm] Max number of GMR pages is 1048576
[ 10.300393] [drm] Max dedicated hypervisor surface memory is 393216 KiB
[ 10.300394] [drm] Maximum display memory size is 131072 KiB
[ 10.300394] [drm] VRAM at 0xe0000000 size is 131072 KiB
[ 10.300395] [drm] MMIO at 0xf0000000 size is 2048 KiB
[ 10.303182] [drm] Zone kernel: Available graphics memory: 2534504 KiB
[ 10.303182] [drm] Zone dma32: Available graphics memory: 2097152 KiB
[ 10.303183] [TTM] Initializing pool allocator
[ 10.303180] [TTM] Initializing DMA pool allocator
[ 10.303200] [drm] Supports vblank timestamp caching Rev 2 (21.10.2013).
[ 10.303200] [drm] No driver support for vblank timestamp query.
[ 10.300327] [drm] Screen Objects Display Unit initialized
[ 10.300420] [drm] width 720
[ 10.300477] [drm] height 400
[ 10.300432] [drm] bpp 32
[ 10.405906] [drm] Fifo max 0x00200000 min 0x00001000 cap 0x00000355
[ 10.406174] [drm] DX: no.
[ 10.406175] [drm] Atomic: yes.
[ 10.406175] [drm] SM: no.
[ 10.406216] [drm:vmw_host_log [vmwgfx]] *ERROR* Failed to send host log message.
[ 10.417523] [drm:vmw_host_log [vmwgfx]] *ERROR* Failed to send host log message.
[ 10.443490] fbcon: svgaDRMfb (fb0) is primary device
[ 10.507920] Console: switching to colour frame buffer device 100x37
[ 10.518092] [drm] Initialized vmwgfx 2.15.0 20180704 for 0000:00:02.0 on minor 0
[ 19.061811] snd_intel8x0 0000:00:05.0: white list rate for 1028:0177 is 40000
[ 25.200687] e1000: enp0s3 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: RX
[ 25.309506] IPV6: ADDRCONF(NETDEV_CHANGE): enp0s3: link becomes ready
[ 50.351172] vboxvideo: Loading version 6.1.26 r145957
[ 50.698544] 22:19:21.645653 main VBoxService 6.1.26 r145957 (verbosity: 0) linux.amd64 (Jul 28 2021 18:25:57) release log
[ 50.699403] 22:19:21.645658 main Log opened 2022-12-05T22:19:21.645648000Z
[ 50.699422] 22:19:21.646730 main OS Product: Linux
[ 50.699572] 22:19:21.646780 main OS Release: 5.0.0-13-generic
[ 50.699579] 22:19:21.646780 main OS Version: #14-Ubuntu SMP Mon Apr 15 14:59:14 UTC 2019
[ 50.699642] 22:19:21.646844 main Executable: /opt/VBoxGuestAdditions-6.1.26/sbin/VBoxService
[ 50.699642] 22:19:21.646845 main Process ID: 1108
[ 50.699642] 22:19:21.646846 main Package type: LINUX_64BITS_GENERIC
[ 50.709906] 22:19:21.657107 main 6.1.26 r145957 started. Verbose level = 0
[ 50.711353] 22:19:21.658551 main vboxR3GuestCtrlDetectPeekGetCancelSupport: Supported (#1)
[ 197.708530] rfkll: input handler disabled
[ 202.500000] 150 9000 Extensions: Microsoft Joliet Level 3
[ 202.600000] 150 9000 Extensions: RRIIP_1991A
[ 781.232406] Loading Module
zk@zk-PC: ~/Desktop$

```

همچنین میتوان محتوای dmesg را در یک فایل با دستور `dmesg > name_of_file.txt` ذخیره کرد. در اینجا `name_of_file` را همان `dmesg` گذاشته ایم.

```

[ 781.232406] Loading Module
zk@zk-PC: ~/Desktop$ dmesg > dmesg.txt
zk@zk-PC: ~/Desktop$ dmesg -c

```

برای جلوگیری از پر شدن بافر سابقه هسته بهتر است زود به زود بافر را خالی کنیم. اگر از دستور `dmesg -c` که در آن به معنی `clear` است استفاده کنیم قبل از اینکه محتوا را پاک کند یک بار دیگر آن را چاپ میکند ولی اگر از `-C` استفاده شود دیگر محتوا را چاپ نمیکند. البته باید دقت شود این کار باید در نقش `super user` انجام شود یعنی `sudo dmesg -c`. و حال اگر `dmesg` را بزنی مشاهده میکنیم که بافر سابقه هسته خالی است.

حال برای حذف کردن ماژول از دستور `sudo rmmod mymod5` استفاده میکنیم و اگر محتوای `dmesg` را ببینیم مشاهده میکنیم که پیغام `Removing Module` نمایش داده می شود.

```

onegit: /usr/kernel/border: rdtsc: operation not permitted
zk@zk-PC:~/Desktop$ sudo dmesg -c
0.000000 Linux version 5.0.0-13-generic (build@lcy01-and64-020) (gcc version 8.3.0 (Ubuntu 8.3.0-6ubuntu1)) #14-Ubuntu SMP Mon Apr 15 14:59:14 UTC 2019 (Ubuntu 5.0.0-13.14-generic 5.0.6)
0.000000 Command line: BOOT_IMAGE=/boot/vmlinuz-5.0.0-13-generic root=UUID=d96f5d99-8925-4f5d-a74d-6031a8d38086 ro quiet splash
0.000000 KERNEL: Linux 5.0.0-13-generic
0.000000 Intel GenuineIntel
0.000000 AMD AuthenticAMD
0.000000 Hygon HygonGenuine
0.000000 Centaur CentaurHauls
0.000000 x86/fpu: x87 FPU will use FXSAVE
0.000000 BIOS-provided physical RAM map:
0.000000 BIOS-e820: [mem 0x0000000000000000-0x00000000000009bfff] usable
0.000000 BIOS-e820: [mem 0x00000000000009c0-0x00000000000009ffff] reserved
0.000000 BIOS-e820: [mem 0x0000000000000a00-0x0000000000000fffff] reserved
0.000000 BIOS-e820: [mem 0x000000000000010000-0x0000000000000dffff] usable
0.000000 BIOS-e820: [mem 0x000000000dffff0000-0x000000000dffffffffff] ACPI data
0.000000 BIOS-e820: [mem 0x00000000fec00000-0x00000000fec00fff] reserved
0.000000 BIOS-e820: [mem 0x00000000fec00000-0x00000000fec00fff] reserved
0.000000 BIOS-e820: [mem 0x00000000fffc0000-0x00000000fffc0fff] reserved
0.000000 BIOS-e820: [mem 0x00000000100000000-0x0000000015ffffffffff] usable
0.000000 NX (Execute Disable) protection: active
0.000000 SMBIOS 2.5 present.
0.000000 DMI: Innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006
0.000000 Hypervisor detected: KVM
0.000000 kvm-clock: using msrc 4b564d01 and 4b564d00
0.000001 kvm-clock: cpu 0, msrc c2001001, primary cpu clock
0.000001 kvm-clock: using sched offset of 9274948422 cycles
0.000002 clocksource: kvm-clock: mask: 0xffffffffffffff max_cycles: 0x1cd42e4dffb, max_idle_ns: 881590591483 ns
0.000004 tsc: Detected 1991.998 MHz processor
0.004782 e820: update [mem 0x00000000-0x00000fff] usable ==> reserved
0.004783 e820: remove [mem 0x00000000-0x00000fff] usable
0.004787 last_pfn = 0x100000 max_arch_pfn = 0x400000000
0.004830 MTRR default type: uncachable
0.004830 MTRR fixed ranges disabled:
0.004840 00000-fffff uncachable
0.004841 MTRR variable ranges disabled:
0.004841 0 disabled
0.004842 1 disabled
0.004842 2 disabled
0.004842 3 disabled
0.004842 4 disabled
0.004843 5 disabled
0.004843 6 disabled
0.004843 7 disabled
0.004848 Disabled
0.004848 x86/PAT: MTRRs disabled skipping PAT initialization too.
0.004852 CPU MTRRs: all blank - virtualized system.
0.004853 x86/PAT: configuration [0-7]: WB WT UC- UC- WB WT UC- UC-

[ 10.300392] [drm] Max CMR lds is 8192
[ 10.300393] [drm] Max number of CMR pages is 1048576
[ 10.300393] [drm] Max dedicated hypervisor surface memory is 393216 KiB
[ 10.300394] [drm] Maximum display memory size is 131072 KiB
[ 10.300394] [drm] VRAM at 0xe0000000 size is 131072 KiB
[ 10.300395] [drm] WRIO at 0xf0000000 size is 2048 KiB
[ 10.303181] [TTM] Zone kernel: Available graphics memory: 2534504 KiB
[ 10.303182] [TTM] Zone dma32: Available graphics memory: 2097152 KiB
[ 10.303183] [TTM] Initializing pool allocator
[ 10.303186] [TTM] Initializing DMA pool allocator
[ 10.303200] [drm] Supports vblank timestamp caching Rev 2 (21.10.2013).
[ 10.303200] [drm] No driver support for vblank timestamp query.
[ 10.308327] [drm] Screen Objects Display Unit initialized
[ 10.308420] [drm] width 720
[ 10.308477] [drm] height 400
[ 10.308532] [drm] bpp 32
[ 10.405906] [drm] Fifo max 0x00200000 min 0x00001000 cap 0x00000355
[ 10.406174] [drm] OK: no.
[ 10.406175] [drm] Atonic: yes.
[ 10.406175] [drm] SMA: it: no.
[ 10.406216] [drm:vmw_host_log [vmwgfx]] *ERROR* Failed to send host log message.
[ 10.417523] [drm:vmw_host_log [vmwgfx]] *ERROR* Failed to send host log message.
[ 10.443499] fbcon: svgaDRMfb (fb0) is primary device
[ 10.507929] Console: switching to colour frame buffer device 100x37
[ 10.510092] [drm] Initialized vmwgfx 2.15.0 20180704 for 0000:00:02.0 on minor 0
[ 19.061011] snd_intel8x0 0000:00:05.0: white list rate for 1020:0177 is 48000
[ 25.280687] e1000: enp0s3 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: RX
[ 25.300960] [IPv6: ADDRCONF(NETDEV_CHANGE): enp0s3: link becomes ready
[ 50.351173] vboxvideo: loading version 6.1.26 r145957
[ 50.698544] 22:19:21.645653 main VBoxService 6.1.26 r145957 (verbosity: 0) Linux.and64 (Jul 28 2021 18:25:57) release log
[ 50.699463] 22:19:21.645658 main OS Product: Linux
[ 50.699522] 22:19:21.646730 main OS Releases: 5.0.0-13-generic
[ 50.699579] 22:19:21.646788 main OS Version: #14-Ubuntu SMP Mon Apr 15 14:59:14 UTC 2019
[ 50.699642] 22:19:21.646844 main Executable: /opt/VBoxGuestAdditions-6.1.26/sbin/VBoxService
[ 50.709906] 22:19:21.646846 main Process ID: 1108
[ 50.711355] 22:19:21.658551 main Package type: LINUX_64BITS_GENERIC
[ 50.711355] 22:19:21.658551 main 6.1.26 r145957 started. Verbose level = 0
[ 197.708339] rfkil: input handler disabled
[ 202.589884] ISO 9660 Extensions: Microsoft Joliet Level 3
[ 202.604873] ISO 9660 Extensions: RRIP_1991A
[ 781.232406] Loading Module
zk@zk-PC:~/Desktop$ dmesg
zk@zk-PC:~/Desktop$ sudo rmmod mynods
zk@zk-PC:~/Desktop$ dmesg
[ 1109.780835] Removing Module
zk@zk-PC:~/Desktop$

```

بخش دو: ساختمان داده های هسته

تمرین 2 : در نقطه ورود ماژول، یک لیست پیوندی شامل پنج عنصر **birthday struct** ایجاد کنید. لیست پیوندی را پیمایش کنید و محتوای آن را به بافر سابقه هسته انتقال دهید. فرمان **dmesg** را احضار کنید تا مطمئن شوید که به محض بار شدن ماژول هسته، لیست به درستی ایجاد می شود. در نقطه خروج ماژول، عناصر لیست را از لیست پیوندی حذف کرده و دوباره حافظه آزاد شده را به هسته برگردانید. باز هم فرمان **dmesg** را احضار کنید تا بررسی کنید به محض برداشتن ماژول هسته، لیست حذف می شود.

هسته ی لینوکس ساختمان داده های مختلفی را پوشش می دهد مثل لیست پیوندی دو طرفه، درخت مینا، ارایه بیتی و...

در این قسمت از آزمایش قصد داریم از لیست پیوندی دو طرفه استفاده کنیم که در کتابخانه **list.h** موجود است. که این کتابخانه را در فایل **mymod5.c** اضافه میکنیم. (`# include <linux/list.h>`)

ابتدا تابع **add_element** را ایجاد میکنیم و برای استفاده از لیست اول یک **struct** ایجاد میکنیم. همانند زیر:

```
struct BirthInfo {
    int day;
    int month;
    int year;
    char city[20];
    struct list_head list;
```

این **struct** عناصر مختلفی دارد که شامل روز و ماه و سال و ارایه ای از کارکتر ها به طول 20 و **list** که از رکورد **list_head** است. برای استفاده از ارایه ای از کارکتر ها باید کتابخانه **string.h** را نیز Include کرد و برای استفاده از رکورد **list_head** باید کتابخانه **types.h** را **include** کرد.

List_head در کتابخانه اش به صورت زیر تعریف شده است که به صورت یک رکورد می باشد که دو ارگومان ***prev** و ***next** دارد که اشاره گر به گره های بعدی و قبلی هستند.

```
struct list_head {
    struct list_head *next, *prev;
};
```


میتوان از macro هایی برای مدیریت list ای که از نوع list_head تعریف شده است استفاده کرد. برای مشخص کردن ابتدای لیست پیوندی از ماکروی LIST_HEAD استفاده میکنیم.

در ادامه ما یک نمونه از struct میسازیم به صورت زیر:

```
struct BirthInfo *student;
student = kmalloc(sizeof(student),GFP_KERNEL);
student -> day =17;
student -> month =2;
student -> year =1380;
strcpy(student -> city,"Tehran");
INIT_LIST_HEAD(&student ->list);
```

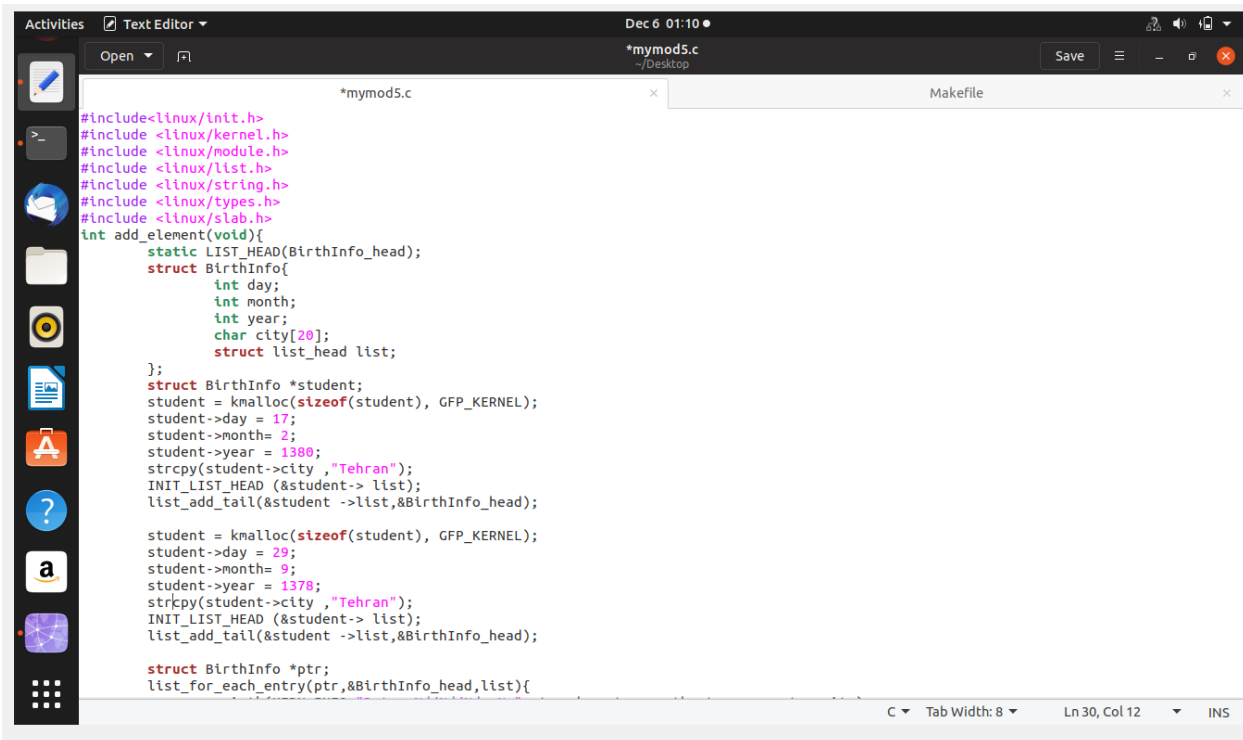
کد بالا نمونه ای از struct به نام student ساخته شده است که در آن kmalloc داریم مثل malloc برای تخصیص حافظه است که k آن معرف kernel است. GFP مخفف get free pages که چگونگی تخصیص حافظه هسته را تعیین میکند. سپس عناصر مختلف struct مقدار دهی میشود. برای city چون ارایه ای از کارکتر ها هست از strcpy استفاده میکنیم که در کتابخانه string.h تعریف شده است. برای آخرین عنصر یعنی list از ماکروی INIT_LIST_HEAD استفاده میکنیم که آن را به صورت خودکار مقدار دهی میکند. حال نمونه نوشته شده را به list اضافه میکنیم. یعنی به استفاده از دستور list_and_tail پارامتر لیست student را اضافه میکنیم به لیستی که ابتدای آن BirthInfo_HEAD میباشد.

این کار را برای نمونه های دیگر هم انجام میدهیم.

حال میخواهیم لیست را پیمایش کنیم و در خروجی چاپ کنیم. از اشاره گری از نوع BirthInfo استفاده میکنیم. برای هر entry داخل لیست محتوا را میگیرد و وارد متغیر ptr می کند. در list_for_each_entry ، Ptr خروجی است و ابتدای لیست و خود لیست ورودی هستند. همچنین printk نیز محتوا را در خروجی به ما نمایش میدهد.

در اخر برای تابع یک return0 میگذاریم.

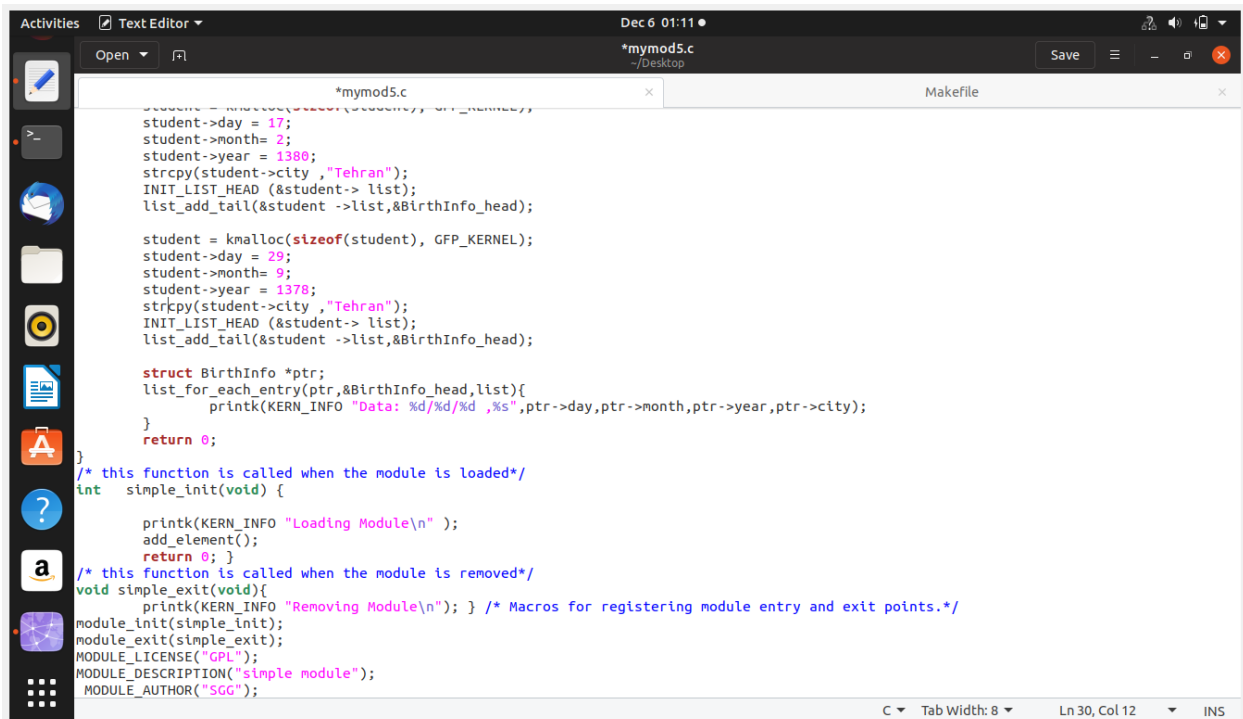
این تابع را وقتی ماژول در حال مقدار دهی اولیه است یعنی در simple_init تابع را فراخوانی میکنیم. برای kmalloc هم باید کتابخانه slab.h را include کرد.



```
#include<linux/init.h>
#include <linux/kernel.h>
#include <linux/module.h>
#include <linux/list.h>
#include <linux/string.h>
#include <linux/types.h>
#include <linux/slab.h>
int add_element(void){
    static LIST_HEAD(BirthInfo_head);
    struct BirthInfo{
        int day;
        int month;
        int year;
        char city[20];
        struct list_head list;
    };
    struct BirthInfo *student;
    student = kmalloc(sizeof(student), GFP_KERNEL);
    student->day = 17;
    student->month= 2;
    student->year = 1380;
    strcpy(student->city,"Tehran");
    INIT_LIST_HEAD (&student-> list);
    list_add_tail(&student ->list,&BirthInfo_head);

    student = kmalloc(sizeof(student), GFP_KERNEL);
    student->day = 29;
    student->month= 9;
    student->year = 1378;
    strcpy(student->city ,"Tehran");
    INIT_LIST_HEAD (&student-> list);
    list_add_tail(&student ->list,&BirthInfo_head);

    struct BirthInfo *ptr;
    list_for_each_entry(ptr,&BirthInfo_head,list){
```



```
        printk(KERN_INFO "Data: %d/%d/%d ,%s",ptr->day,ptr->month,ptr->year,ptr->city);
    }
    return 0;
}
/* this function is called when the module is loaded*/
int simple_init(void) {
    printk(KERN_INFO "Loading Module\n" );
    add_element();
    return 0; }
/* this function is called when the module is removed*/
void simple_exit(void){
    printk(KERN_INFO "Removing Module\n"); } /* Macros for registering module entry and exit points.*/
module_init(simple_init);
module_exit(simple_exit);
MODULE_LICENSE("GPL");
MODULE_DESCRIPTION("simple module");
MODULE_AUTHOR("SGG");
```

سپس در ترمینال برای این فایل make انجام میدهم.سپس ماژول را با دستور `sudo insmod mymod5.ko` را انجام میدهم سپس با دستور `sudo rmmod mymod5` حذف میکنیم.با `dmesg`محتوای بافر هسته را مشاهده میکنیم.

```
Activities Terminal Dec 6 01:13
zk@zk-PC: ~/Desktop

/home/zk/Desktop/mymod5.c:36:20: error: missing terminating " character
printk(KERN_INFO "Data: %d/%d/%d ,%s,ptr->day,ptr->month,ptr->year,ptr->city);
                                     ^
/home/zk/Desktop/mymod5.c:37:2: error: expected ')' before '}' token
}
^
/home/zk/Desktop/mymod5.c:37:2: error: expected ';' before '}' token
make[2]: *** [scripts/Makefile.build:293: /home/zk/Desktop/mymod5.o] Error 1
make[1]: *** [Makefile:1584: _module_/home/zk/Desktop] Error 2
make[1]: Leaving directory '/usr/src/linux-headers-5.0.0-13-generic'
make: *** [Makefile:3: all] Error 2
zk@zk-PC:~/Desktop$ make
make -C /lib/modules/5.0.0-13-generic/build M=/home/zk/Desktop modules
make[1]: Entering directory '/usr/src/linux-headers-5.0.0-13-generic'
CC [M] /home/zk/Desktop/mymod5.o
/home/zk/Desktop/mymod5.c: In function 'add_element':
/home/zk/Desktop/mymod5.c:34:2: warning: ISO C90 forbids mixed declarations and code [-Wdeclaration-after-statement]
    struct BirthInfo *ptr;
    ^
Building modules, stage 2.
MODPOST 1 modules
CC      /home/zk/Desktop/mymod5.mod.o
LD [M]   /home/zk/Desktop/mymod5.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.0.0-13-generic'
zk@zk-PC:~/Desktop$ sudo insmod mymod5.ko
[sudo] password for zk:
Sorry, try again.
[sudo] password for zk:
zk@zk-PC:~/Desktop$ sudo rmmod mymod5
zk@zk-PC:~/Desktop$ dmesg
[ 1302.829583] Removing Module
[ 3137.780747] Loading Module
[ 3137.780748] Data: 17/2/1380 ,Tehran
[ 3137.780749] Data: 29/9/1378 ,Tehran
[ 3154.890532] Removing Module
zk@zk-PC:~/Desktop$
```