- Kernel Module တွေဟာ Linux Kernel System တစ်ခုလုံးကို reboot လုပ်စရာမလိုအပ်ဘဲ Hardware နှင့် Software လိုအပ်ချက် အပေါ်လိုက်ပြီး ပြောင်းလွယ်/ ပြင်လွယ် ရှိအောင် နှင့် လိုက်လျောညီထွေဖြစ်အောင် လုပ် ဆောင်ပေးပါတယ်။

- Kernel Module ကို အသုံးများတဲ့ နေရာတွေ ကတော့ hardware peripheral တွေကို support ပေးတဲ့ Driver တွေ / အမျိုးမျိုး သော File System Type တွေကို Kernel က Support ဖြစ်အောင် / Security Feacture တွေ အသစ် ထပ်ထည့် တဲ့ အခါ မျိုး တွေမှာ သုံးပါတယ်။

- ရိုးရှင်းတဲ့ Kernel Module တစ်ခု ကို တည် ဆောက်ကြည့်ကြ ရ အောင်။
Kernel Module တစ်ခု စတင်တည်ဆောက်ရန် အတွက် Linux kernel တွင် kernel development environment (kernel header နဲ့ development package) များရှိထားရန်လိုအပ်ပါတယ်။

အောက်မှာ ဖော်ပြထားတာကတော့ " Hello World " message ကို Kernel ရဲ့ System log file ထဲတွင် သွားထည့် တဲ့ kernel module program ဖြစ်ပါတယ်။

ယခု program ကို Ubuntu 20.4 မှာ စမ်းပြထားတာဖြစ်ပါတယ်။ အခြား Linux Kernel များအလားတူဘဲ ဖြစ်ပါတယ်။

ပထမဆုံး Linux Kernel ရှိ Package များ update ဖြစ်စေရန် Terminal တွင် အောက်ပါ command ကို ရိုက်ပါ။

```
sudo apt update && sudo apt upgrade
```

လိုအပ်သော development tools နှင့် kernel header များ ကို အောက် ပါ command အတိုင်း install လုပ်ပါ။

```
$ sudo apt-get install gcc do build-essential libncurses-dev
exuberant-ctags build-essential linux-headers-`uname -r`
```

Install လုပ်ပြီးသွားရင်တော့ Kernel Module တစ်ခု အတွက် code စရေးလို့ရပါပြီ။

အောက်မှာ ဖော်ပြထားတဲ့ code example ကတော့ kernel module မှ load နှင့် unload လုပ်တဲ့အခါ System log မှာ Print message အနေနဲ့ ထည့်ပေးတဲ့ program ဖြစ်ပါတယ်။

```
#include <linux/init.h>
#include <linux/kernel.h>
#include <linux/module.h>
MODULE_LICENSE("GPL");
static int __init module_start(void)
{
    printk(KERN_INFO "Hello, World!\n");
```

```
    return 0;
}
static void __exit module_end(void)
{
    printk(KERN_INFO "Goodbye, World!\n");
}
module_init(module_start);
module_exit(module_end);
```

“Hello World” Kernel Module Program ကို အနည်းငယ် ရှင်းလင်းရအောင်။

- #include<linux/init.h> : Module တစ်ခု တည်ဆောက်ရန် လိုအပ်တဲ့ kernel initialization functions အတွက် header file ဖြစ်ပါတယ်။
- #include<linux/kernel.h>: kernel log file ထဲကို message print ထုတ်ရန်အတွက် လိုအပ်တဲ့ kernel လုပ်ဆောင်ချက်တွေ အတွက် header file ဖြစ်ပါတယ်။
- #include<linux/module.h> : kernel ထဲကို ထည့်သွင်းရန် kernel module အတွက် header ဖြစ်ပါတယ်။
- MODULE_LICENSE(“GPL”); Module အတွက် : GNU (General Public License) အထွေ ထွေ အများပြည်သူသုံး လိုင်စင် ဖြစ်ပါတယ်။ Kernel Module များအတွက် မဖြစ်မနေလိုအပ်ပါတယ်။
- Static int__init module_start(void) : Module အတွက် initialization function ဖြစ်ပါတယ်။ kernel မှ module ကို load တဲ့အခါ ယခု function ကို ခေါ်၍ အသုံးပြုမှာဖြစ်ပါတယ်။ “__init” keyword ကတော့ ယခု function ကို call ခေါ် ၍ အသုံးပြုပြီးပါက Memory အသုံးပြုမှု သက်သာစေရန် kernel မှ function ကို discard လုပ်ရန် ဖြစပါတယ်။
- printk(KERN_INFO “Hello, World!\n”); : printk() function ကို အသုံးပြု၍ “ Hello, World” message ကို kernel log ထဲကို print ထုတ်ရန်ဖြစ်ပါတယ်။ KERN_CONT ဆိုတာက kernel မှ လုပ်ဆောင်သည့် လုပ်ဆောင်ချက်များနှင့် ပတ်သတ်၍ log file message များအတွက် အသုံးပြုသည့် log level ဦးစားပေးအဆင့်(log level 6)ဖြစ်ပါတယ်။
- return 0 : Module initialization function အောင်မြင်စွာ ပြီးဆုံးပါက int 0 return ပေးရန်အတွက်ဖြစ်ပါတယ်။
- static void__exit module_end(void) : Module အတွက် cleanup function ဖြစ်ပါတယ်။ kernel မှ module ကို unload တဲ့အခါ ယခု function ကို ခေါ်၍ အသုံးပြုမှာဖြစ်ပါတယ်။ “__exit” keyword ကတော့ ယခု function ကို call ခေါ် ၍ အသုံးပြုပြီးပါက Memory အသုံးပြုမှု သက်သာစေရန် kernel မှ function ကို discard လုပ်ရန် ဖြစပါတယ်။
- printk(KERN_INFO “Goodbye, World!\n”); : printk() function ကို အသုံးပြု၍ “ Goodbye, World” message ကို kernel log ထဲကို print ထုတ်ရန်ဖြစ်ပါတယ်။ KERN_CONT ဆိုတာက kernel မှ လုပ်ဆောင်သည့် လုပ်ဆောင်ချက်များနှင့် ပတ်သတ်၍ log file message များအတွက် အသုံးပြုသည့် log level ဦးစားပေးအဆင့်(log level 6)ဖြစ်ပါတယ်။
- module_init(module_start);: Module load လုပ်သည့်အခါတွင် kernel မှ line register initilization လုပ်ပေးသည့် function ဖြစ်ပါသည်။
- module_exit(module_end);: Module unload လုပ်သည့်အခါတွင် kernel မှ line register initilization လုပ်ပေးသည့် function ဖြစ်ပါတယ်။

Module ကို Compile လုပ်ရန် Make file တစ်ခု တည်ဆောက်ရန် လိုအပ်ပါတယ်။

```
obj-m += hello.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
```

ယခု Make file သည် hello.c source file မှ hello.o အမည်ရှိ kernel module တစ်ခု တည်ဆောက်ပေးပါလိမ့်မယ်။

Module တည်ဆောက်ရန် အတွက် Make file ပါဝင်သော directory ကို run ပေးရန် လိုအပ်ပါတယ်။

Run တာ အဆင်ပြေပါက hello.ko အမည် ရှိ .ko နှင့် ဆုံးသော file ကို generate ထုတ်ပေးပါမည်။

.ko file ဟာ kernel module ကို compile လုပ်ထားသော file ဘဲဖြစ်ပါတယ်။

Make clean ကို အသုံးပြု၍လည်း directory နှင့် generate file ကို ရှင်းထုတ်နိုင်ပါတယ်။

all : ဒီ keyword ဟာ target မသက်မှတ်ထားပါက Default Target အနေဖြင့် Kernel Module ကို တည်ဆောက်ပေးမှာ ဖြစ်ပါတယ်။

clean : ဒီ keyword ဟာ kernel module module build process လုပ်နေချိန် create လုပ်ခဲ့ သော file များကို ရှင်းထုတ်ပေးမှာ ဖြစ်ပါတယ်။

Module load လုပ်ရန် အတွက် insmod command ကို အသုံးပြုရန် လိုအပ်ပါတယ်။

```
$ sudo insmod hello.ko
```

Module unload လုပ်ရန် အတွက် rmmod command ကို အသုံးပြုရန် လိုအပ်ပါတယ်။ system

```
$ sudo rmmod hello.ko
```

System log file တွင် kernel module မှ ထုတ်ပေး သော print message ကို ကြည့်ရှုရန် “dsmg” command ကို အသုံးပြု၍ ကြည့် ရှု နိုင်ပါသည်။ ဒါမှမဟုတ် /var/log/syslog log file ကို ဖွင့်၍လည်းကြည့်ရှု နိုင်ပါသည်။

ယခု ဆိုရင် တော့ ရိုးရှင်းလွယ်ကူတဲ့ kernel module တစ်ခု တည်ဆောက်လို့ ပြီးပါပြီ။

```
$ dmesg | grep World
[566458.370280] Hello, World!
[566490.188611] Goodbye, World!
```

kernel module တစ်ခု တည်ဆောက်ရန် အတွက် Linux Operating System နှင့် Command Line များ ကျွမ်းကျင်ရန် လိုအပ်သလို C programming language ကို အခြေခံ နားလည်ပြီး ရေးသားနိုင်ရန် လိုအပ်ပါတယ်။

Hello world kernel module တစ်ခု တည်ဆောက်ရန် အတွက် လိုအပ်သော အဆင့် ဆင့် ကို ဖော်ပြခဲ့ပြီ ဖြစ်ပါတယ်။

Command အနှစ်ချုပ် အနေဖြင့် “insmod” command ကို အသုံး ပြု၍ module ကို kernel ထဲကို compile and load လုပ်ပြီး “ rmmod” command ကို သုံး၍ module ကို unload လုပ်မှာ ဖြစ်ပါတယ်။