DHRUV GARG 16 BCE 1190

USER AND KERNEL THREAD

LAB 5

USER THREAD

STEP 1: Creating the "uthread_create.c" file to demonstrate user thread.

```
vmdhruv@ubuntu:~/OS_LAB$ gedit uthread_create.c
```

STEP 2: Code inside "uthread_create.c"

```c
#include<stdio.h>
#include<string.h>
#include<pthread.h>
#include<stdlib.h>
#include<unistd.h>

pthread_t tid[2];

void* doSomeThing(void *arg)
{
    unsigned long i = 0;
    pthread_t id = pthread_self();

    if(pthread_equal(id,tid[0]))
    {
        printf("\n First thread processing\n");
    }
    else
    {
        printf("\n Second thread processing\n");
    }

    for(i=0; i<(0xFFFFFFFF);i++);

    return NULL;
}

int main(void)
{
    int i = 0;
    int err;

    while(i < 2)
    {
        err = pthread_create(&(tid[i]), NULL, &doSomeThing, NULL);
        if (err != 0)
            printf("\ncan't create thread :[%s]", strerror(err));
        else
            printf("\n Thread created successfully\n");

        i++;
    }

    sleep(5);
    return 0;
}
```

STEP 3: Executing the code

```
vmdhruv@ubuntu:~/OS_LAB$ gcc -pthread -o uthread_create uthread_create.c
vmdhruv@ubuntu:~/OS_LAB$ ./uthread_create

 Thread created successfully

 Thread created successfully

 Second thread processing

 First thread processing _
```

KERNEL THREAD

STEP 1: Installing the linux headers using "apt-get"

```
vmdhruv@ubuntu:~$ sudo apt-get install build-essential linux-headers
-$(uname -r)
[sudo] password for vmdhruv:
Reading package lists... Done
Building dependency tree
Reading state information... Done
build-essential is already the newest version (12.1ubuntu2).
linux-headers-4.10.0-28-generic is already the newest version (4.10.
0-28.32~16.04.2).
linux-headers-4.10.0-28-generic set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 308 not upgraded.
vmdhruv@ubuntu:~$
```

STEP 2: Writing the source code for "hello.c" module

```c
#include <linux/module.h>    // included for all kernel modules
#include <linux/kernel.h>    // included for KERN_INFO
#include <linux/init.h>      // included for __init and __exit macros

MODULE_LICENSE("GPL");
MODULE_AUTHOR("Dhruv Garg");
MODULE_DESCRIPTION("A Simple Hello World module");

static int __init hello_init(void)
{
    printk(KERN_INFO "Hello world!\n");
    return 0;    // Non-zero return means that the module couldn't be loaded.
}

static void __exit hello_cleanup(void)
{
    printk(KERN_INFO "Cleaning up module.\n");
}

module_init(hello_init);
module_exit(hello_cleanup);
```

STEP 3: Create "Makefile" to compile kernel module

```
vmdhruv@ubuntu:~$ gedit Makefile &
[1] 2551
vmdhruv@ubuntu:~$
```

Code

```
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
```

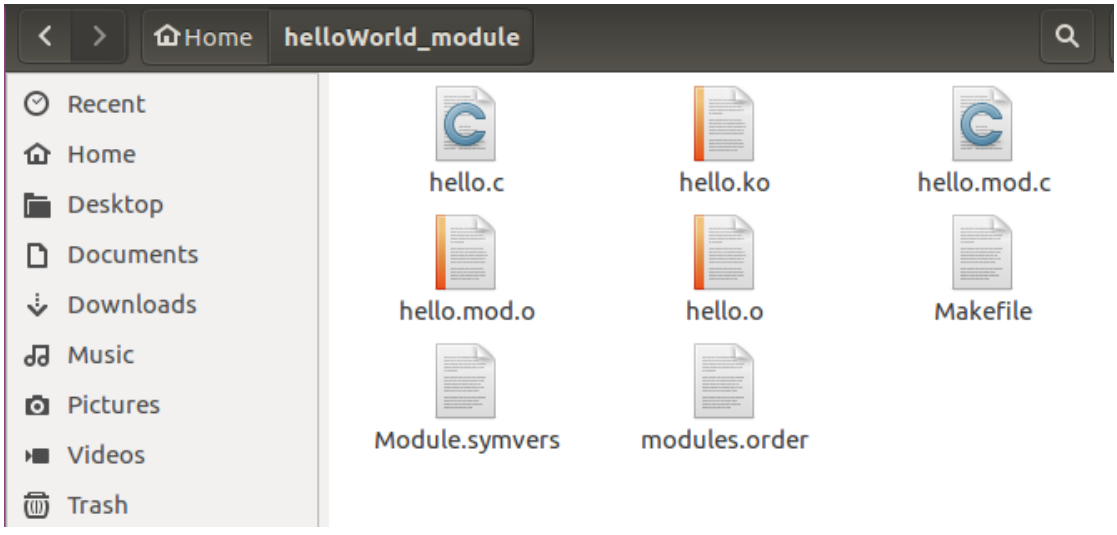
Displaying the path and contents of the module directory

```
vmdhruv@ubuntu:~$ ls
Desktop     Downloads         helloWorld_module  OS_LAB     Public     Videos
Documents   examples.desktop  Music              Pictures   Templates
vmdhruv@ubuntu:~$ cd helloWorld_module
vmdhruv@ubuntu:~/helloWorld_module$ ls
hello.c  Makefile
```

STEP 4: Use the make command to compile hello world kernel module.

```
vmdhruv@ubuntu:~/helloWorld_module$ make
make -C /lib/modules/4.10.0-28-generic/build M=/home/vmdhruv/helloWorld_module m
odules
make[1]: Entering directory '/usr/src/linux-headers-4.10.0-28-generic'
  CC [M]  /home/vmdhruv/helloWorld_module/hello.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /home/vmdhruv/helloWorld_module/hello.mod.o
  LD [M]  /home/vmdhruv/helloWorld_module/hello.ko
make[1]: Leaving directory '/usr/src/linux-headers-4.10.0-28-generic'
```

Files inside the directory after compilation:



Using the "modinfo" command we can view the properties of the module

```
vmdhruv@ubuntu:~/helloWorld_module$ modinfo hello.ko
filename:       /home/vmdhruv/helloWorld_module/hello.ko
description:    A Simple Hello World module
author:         Dhruv Garg
license:        GPL
srcversion:     453EC1ADBC6C59A397EEC0A
depends:
vermagic:       4.10.0-28-generic SMP mod_unload
```

STEP 5: Running the "insmod" command in super-user mode to insert the module into the kernel

```
vmdhruv@ubuntu:~/helloWorld_module$ sudo insmod ./hello.ko
[sudo] password for vmdhruv:
```

Using the "lsmod" command we can view all the modules in the kernel. The newly inserted "hello" module gets listed there as well.

```
vmdhruv@ubuntu:~$ lsmod
Module                  Size  Used by
hello                  16384  0
vmw_vsock_vmci_transport    28672  2
vsock                  36864  3 vmw_vsock_vmci_transport
vmw_balloon            20480  0
snd_ens1371            28672  2
snd_ac97_codec        131072  1 snd_ens1371
gameport               16384  1 snd_ens1371
crct10dif_pclmul       16384  0
crc32_pclmul           16384  0
ac97_bus               16384  1 snd_ac97_codec
snd_pcm               102400  2 snd_ac97_codec,snd_ens1371
ghash_clmulni_intel    16384  0
pcbc                   16384  0
aesni_intel           167936  0
aes_x86_64             20480  1 aesni_intel
crypto_simd            16384  1 aesni_intel
glue_helper            16384  1 aesni_intel
snd_seq_midi           16384  0
snd_seq_midi_event     16384  1 snd_seq_midi
cryptd                 24576  3 crypto_simd,ghash_clmulni_intel,aesni_intel
snd_rawmidi            32768  2 snd_seq_midi,snd_ens1371
snd_seq                65536  2 snd_seq_midi_event,snd_seq_midi
intel_rapl_perf        16384  0
snd_seq_device         16384  3 snd_seq,snd_rawmidi,snd_seq_midi
snd_timer              32768  2 snd_seq,snd_pcm
joydev                 20480  0
input_leds             16384  0
serio_raw              16384  0
snd                    77824  11 snd_seq,snd_ac97_codec,snd_timer,snd_rawmidi,snd_ens1371,snd_seq_device,snd_pcm
soundcore              16384  1 snd
i2c_piix4              24576  0
vmw_vmci               69632  2 vmw_balloon,vmw_vsock_vmci_transport
shpchp                 36864  0
nfit                   49152  0
mac_hid                16384  0
parport_pc             32768  0
```

STEP 6: Use “dmesg” command, to see the output from the kernel thread.

```
vmdhruv@ubuntu:~/helloWorld_module$ dmesg | tail -1
[  930.508524] Hello world!
```

```
vmdhruv@ubuntu:~/helloWorld_module$ sudo rmmod hello.ko
vmdhruv@ubuntu:~/helloWorld_module$ dmesg | tail -1
[ 1768.281222] Cleaning up module.
vmdhruv@ubuntu:~/helloWorld_module$
```

Explanation: *When a module is inserted into the kernel, the module_init macro will be invoked, which will call the function hello_init. Similarly, when the module is removed with rmmod, module_exit macro will be invoked, which will call the hello_exit. Using dmesg command, we can see the output from the sample Kernel module.*

STEP 7: Verify that the kernel module has been removed after using the “rmmod” command.

```
vmdhruv@ubuntu:~$ lsmod
Module                  Size  Used by
vmw_vsock_vmci_transport    28672  2
vsock                   36864  3 vmw_vsock_vmci_transport
vmw_balloon             20480  0
snd_ens1371             28672  2
snd_ac97_codec         131072  1 snd_ens1371
gameport                16384  1 snd_ens1371
crct10dif_pclmul        16384  0
crc32_pclmul            16384  0
ac97_bus                16384  1 snd_ac97_codec
snd_pcm                102400  2 snd_ac97_codec,snd_ens1371
ghash_clmulni_intel     16384  0
pcbc                    16384  0
aesni_intel            167936  0
aes_x86_64              20480  1 aesni_intel
crypto_simd             16384  1 aesni_intel
glue_helper             16384  1 aesni_intel
snd_seq_midi            16384  0
snd_seq_midi_event      16384  1 snd_seq_midi
cryptd                  24576  3 crypto_simd,ghash_clmulni_intel,aesni_intel
snd_rawmidi             32768  2 snd_seq_midi,snd_ens1371
snd_seq                 65536  2 snd_seq_midi_event,snd_seq_midi
intel_rapl_perf         16384  0
snd_seq_device          16384  3 snd_seq,snd_rawmidi,snd_seq_midi
snd_timer               32768  2 snd_seq,snd_pcm
joydev                  20480  0
input_leds              16384  0
serio_raw               16384  0
snd                     77824  11 snd_seq,snd_ac97_codec,snd_timer,snd_rawmidi,snd_ens1371,snd_seq_device,snd_pcm
soundcore               16384  1 snd
i2c_piix4               24576  0
vmw_vmci                69632  2 vmw_balloon,vmw_vsock_vmci_transport
shpchp                  36864  0
nfit                    49152  0
mac_hid                 16384  0
parport_pc              32768  0
ppdev                   20480  0
```