



UACH  
UNIVERSIDAD AUTÓNOMA DE  
CHIHUAHUA

---

FACULTAD DE INGENIERÍA

3er Parcial

Módulo de Kernel

Sistemas Operativos I

Docente: M.I. Iván Miguel Chavero Jurado

Grupo: 5HW1

Alumno:

Diego Hilario Mendoza Ramirez 338956



## Índice

Transcripción de Código “kernel_3er_parcial.c” .....	1
Transcripción de Código “Makefile” .....	4
Funcionamiento .....	4
Como usarlo .....	5
Referencias .....	10

## Transcripción de Código "kernel\_3er\_parcial.c"

```
#include <linux/init.h>
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/fs.h>
MODULE_LICENSE("GPL");
MODULE_AUTHOR("a338956");
MODULE_DESCRIPTION("EXAMEN FINAL SO ");
MODULE_VERSION("0.01");

#define DEVICE_NAME "Kernel_3er_parcial"

static int major;

static int dev_open(struct inode*, struct file*);
static int dev_release(struct inode*, struct file*);
static ssize_t dev_read(struct file*, char*, size_t,
loff_t*);
static ssize_t dev_write(struct file*, const char*, size_t,
loff_t*);

static struct file_operations fops = {
    .open = dev_open,
    .read = dev_read,
    .write = dev_write,
    .release = dev_release,
};

static int __init super_duper_Inicial(void) {
    major = register_chrdev(0, DEVICE_NAME, &fops);

    if (major < 0) {
```

```
    printk(KERN_ALERT "Super Duper Fallido\n");
    return major;
}

    printk(KERN_INFO "Inicializando Módulo 3er
Parcial!!!\n");

    printk(KERN_INFO "Asignado al mayor %d. para hablar
con\n", major);

    printk(KERN_INFO "el driver, creado el archivo de
dispositivo\n");

    printk(KERN_INFO "'copy/paste --> sudo mknod /dev/%s c %d
0'.\n", DEVICE_NAME, major);

    printk(KERN_INFO "Try various minor numbers. Try to cat
and echo to\n");

    printk(KERN_INFO "the device file.\n");

    printk(KERN_INFO "Remove the device file and module when
done.\n");

    return 0;
}

static int dev_open(struct inode *inodep, struct file *filep)
{
    printk(KERN_INFO " --> Abierto\n");
    return 0;
}

static ssize_t dev_write(struct file *filep, const char
*buffer,

                        size_t len, loff_t *offset) {
```

```
    printk(KERN_INFO "Sorry, Super Duper --> Es de Solo
Lectura\n");

    return -EFAULT;
}

static int dev_release(struct inode *inodep, struct file
*filep) {

    printk(KERN_INFO " --> Finalizado\n");

    return 0;
}

static ssize_t dev_read(struct file *filep, char *buffer,
size_t len, loff_t *offset) {

    int errors = 0;

    char *message = "I want to break free, I want to break
free...";

    int message_len = strlen(message);

    errors = copy_to_user(buffer, message, message_len);

    return errors == 0 ? message_len : -EFAULT;

    return 0;
}

static void __exit super_duper_Final(void) {

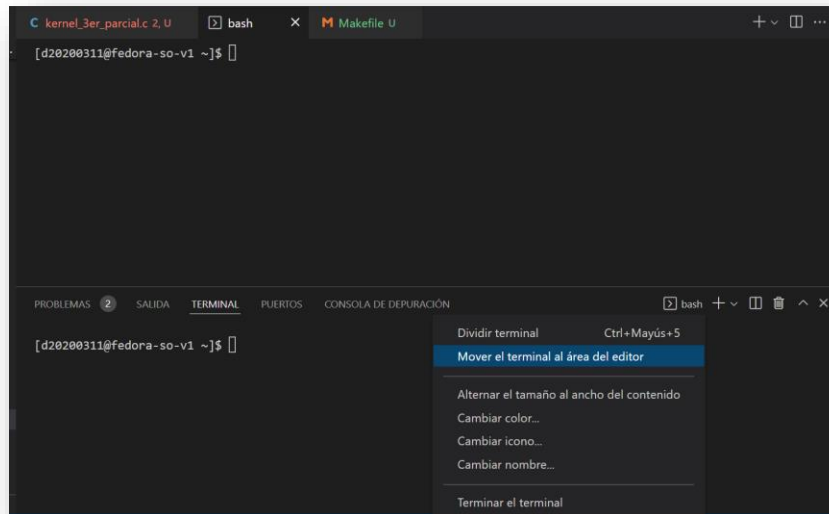
    printk(KERN_INFO ".:Retirando Modulo:.\n");
}

module_init(super_duper_Inicial);
module_exit(super_duper_Final);
```



## Como usarlo

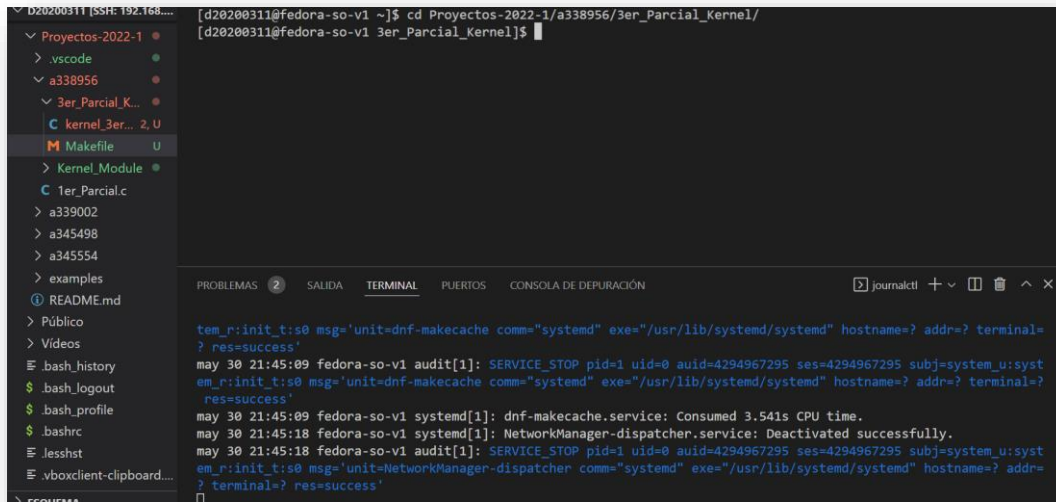
Se empieza con abrir 2 Ventanas de Comando (una como pestaña y otra normal):



Después se escriben los siguientes comandos en las terminales el de abajo se usa para ver los estados de la máquina de virtual que estamos corriendo y el de arriba tenemos que ubicarnos a la carpeta en donde tengamos no guardado nuestro archivo en C y nuestro archivo Makefile



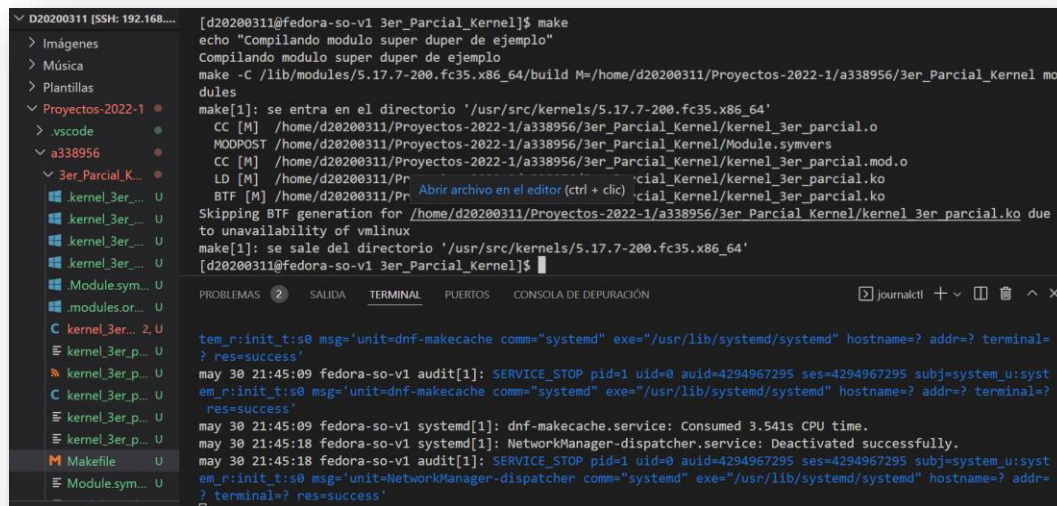
Quedando las terminales de este modo:



```
[d20200311@fedora-so-v1 ~]$ cd Proyectos-2022-1/a338956/3er_Parcial_Kernel/
[d20200311@fedora-so-v1 3er_Parcial_Kernel]$
```

```
tem_r:~$ init_t:s0 msg='unit=dnf-makecache comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=? res=success'
may 30 21:45:09 fedora-so-v1 audit[1]: SERVICE_STOP pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:syst
em_r:~$ init_t:s0 msg='unit=dnf-makecache comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=?
res=success'
may 30 21:45:09 fedora-so-v1 systemd[1]: dnf-makecache.service: Consumed 3.541s CPU time.
may 30 21:45:18 fedora-so-v1 systemd[1]: NetworkManager-dispatcher.service: Deactivated successfully.
may 30 21:45:18 fedora-so-v1 audit[1]: SERVICE_STOP pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:syst
em_r:~$ init_t:s0 msg='unit=NetworkManager-dispatcher comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=?
terminal=? res=success'
```

Luego se escribe en la Terminal de arriba (todo se realiza en la Terminal de Arriba) “make” y compilara el .c con el Makefile que tenemos:



```
[d20200311@fedora-so-v1 3er_Parcial_Kernel]$ make
echo "Compilando modulo super duper de ejemplo"
Compilando modulo super duper de ejemplo
make -C /lib/modules/5.17.7-200.fc35.x86_64/build M=/home/d20200311/Proyectos-2022-1/a338956/3er_Parcial_Kernel mo
dules
make[1]: se entra en el directorio '/usr/src/kernels/5.17.7-200.fc35.x86_64'
CC [M] /home/d20200311/Proyectos-2022-1/a338956/3er_Parcial_Kernel/kernel_3er_parcial.o
MODPOST /home/d20200311/Proyectos-2022-1/a338956/3er_Parcial_Kernel/Module.symvers
CC [M] /home/d20200311/Proyectos-2022-1/a338956/3er_Parcial_Kernel/kernel_3er_parcial.mod.o
LD [M] /home/d20200311/Proyectos-2022-1/a338956/3er_Parcial_Kernel/kernel_3er_parcial.ko
BTF [M] /home/d20200311/Proyectos-2022-1/a338956/3er_Parcial_Kernel/kernel_3er_parcial.ko
Skipping BTF generation for /home/d20200311/Proyectos-2022-1/a338956/3er_Parcial_Kernel/kernel_3er_parcial.ko due
to unavailability of vmlinux
make[1]: se sale del directorio '/usr/src/kernels/5.17.7-200.fc35.x86_64'
[d20200311@fedora-so-v1 3er_Parcial_Kernel]$
```

```
tem_r:~$ init_t:s0 msg='unit=dnf-makecache comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=? res=success'
may 30 21:45:09 fedora-so-v1 audit[1]: SERVICE_STOP pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:syst
em_r:~$ init_t:s0 msg='unit=dnf-makecache comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=?
res=success'
may 30 21:45:09 fedora-so-v1 systemd[1]: dnf-makecache.service: Consumed 3.541s CPU time.
may 30 21:45:18 fedora-so-v1 systemd[1]: NetworkManager-dispatcher.service: Deactivated successfully.
may 30 21:45:18 fedora-so-v1 audit[1]: SERVICE_STOP pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:syst
em_r:~$ init_t:s0 msg='unit=NetworkManager-dispatcher comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=?
terminal=? res=success'
```

Podemos observar que se agrega varios archivos nuevos que fueron necesarias para poder confirmar el programa.



Luego se utiliza el comando “sudo insmod kernel\_3er\_parcial.ko”

sudo --> Otorgar permisos de administrador

insmod --> Instalar módulo

kernel\_3er\_parcial.ko --> Nombre del .c con la extensión .ko que se creó anteriormente al darle al make

Dando como resultado en la terminal de abajo:

```
PROBLEMAS 2 SALIDA TERMINAL PUERTOS CONSOLA DE DEPURACIÓN journalctl + v [ ] [ ] ^ x

may 30 22:24:03 fedora-so-v1 audit[8024]: CRED_DISP pid=8024 uid=1000 auid=1000 ses=3 subj=unconfined_u:unconfine
d_r:unconfined_t:s0-s0:c0.c1023 msg='op=PAM:setcred grantors=pam_env,pam_fprintd acct="root" exe="/usr/bin/sudo"
hostname=? addr=? terminal=/dev/pts/0 res=success'
may 30 22:24:03 fedora-so-v1 kernel: Inicializando Módulo 3er Parcial!!!
may 30 22:24:03 fedora-so-v1 kernel: Asignado al mayor 235. para hablar con
may 30 22:24:03 fedora-so-v1 kernel: el driver, creado el archivo de dispositivo
may 30 22:24:03 fedora-so-v1 kernel: 'copy/paste --> sudo mknod /dev/Kernel_3er_parcial c 235 0'.
may 30 22:24:03 fedora-so-v1 kernel: Try various minor numbers. Try to cat and echo to
may 30 22:24:03 fedora-so-v1 kernel: the device file.
may 30 22:24:03 fedora-so-v1 kernel: Remove the device file and module when done.
[ ]
```

Para comprobar que se haya insertado se escribe el sig comando

“lsmod | grep kernel\_3er\_parcial”

lsmod --> Listar módulos

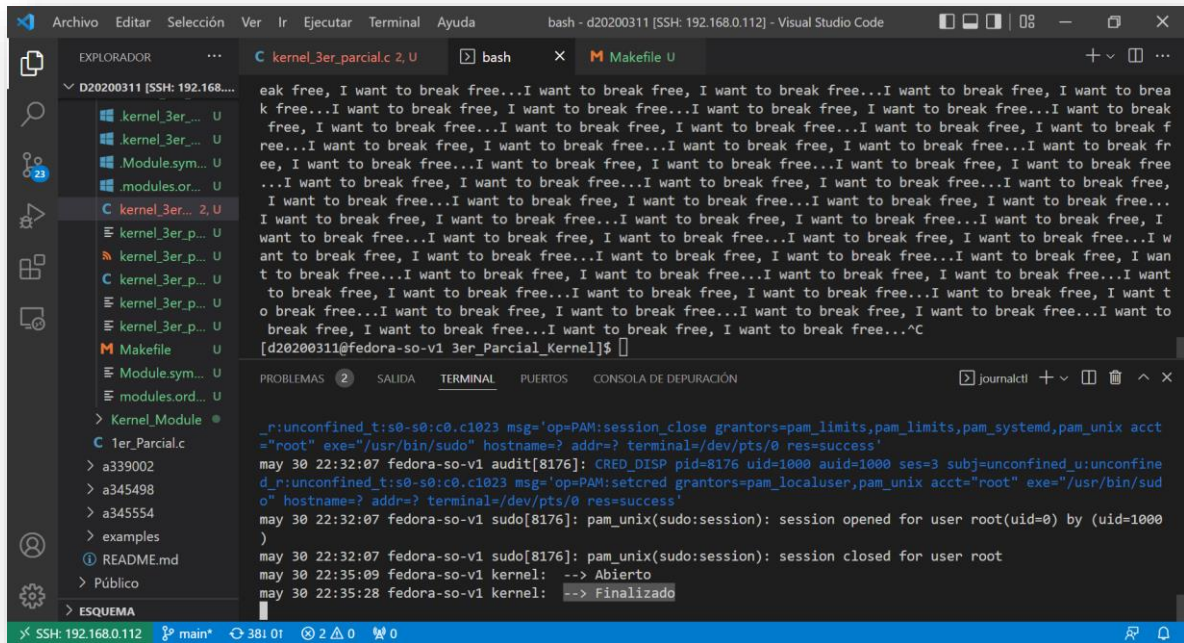
| grep --> Filtrar

kernel\_3er\_parcial --> Nombre del .c sin ninguna extensión

```
[d20200311@fedora-so-v1 3er_Parcial_Kernel]$ lsmod | grep kernel_3er_parcial
kernel_3er_parcial      16384  0
[d20200311@fedora-so-v1 3er_Parcial_Kernel]$ [ ]
```



Para Cerrar comunicación se teclea "Ctrl" + "c" dando el mensaje de "--> Finalizado":



## Referencias

- [https://github.com/Sistemas-Operativos-I-UACH/Proyectos-2022-1/blob/main/a338956/Parcial\\_3/Makefile](https://github.com/Sistemas-Operativos-I-UACH/Proyectos-2022-1/blob/main/a338956/Parcial_3/Makefile)
- [https://github.com/Sistemas-Operativos-I-UACH/Proyectos-2022-1/blob/main/a338956/Parcial\\_3/kernel\\_3er\\_parcial.c](https://github.com/Sistemas-Operativos-I-UACH/Proyectos-2022-1/blob/main/a338956/Parcial_3/kernel_3er_parcial.c)
- [https://github.com/Sistemas-Operativos-I-UACH/Proyectos-2022-1/tree/main/examples/example\\_kernel\\_module](https://github.com/Sistemas-Operativos-I-UACH/Proyectos-2022-1/tree/main/examples/example_kernel_module)
- <https://stackoverflow.com/questions/8697300/how-to-create-a-device-in-dev-automatically-upon-loading-of-the-kernel-module-f#41996319>