

# Relatório Módulo do Kernel

---

Nome do Aluno: Murilo Aldigueri Marino

## Parte I

---

Inicialmente foi necessário criar um makefile para compilar o código e facilitar os processos de teste. O arquivo foi criado com o seguinte conteúdo:

```
# Nome do módulo
MODULE_NAME = simple

# Nome do zip
ZIP_NAME = $(MODULE_NAME)

# Lista de arquivos fonte
SOURCES = $(MODULE_NAME).c

# Lista todos os arquivos envolvidos
FILES = $(SOURCES) $(HEADERS) Makefile

# Diretório do kernel
KDIR = /lib/modules/$(shell uname -r)/build

# Adiciona o arquivo .o ao obj-m para indicar que ele é um módulo
obj-m += $(MODULE_NAME).o

# Alvo padrão
all: $(MODULE_NAME).ko

# Regras de compilação
$(MODULE_NAME).ko: $(SOURCES)
    make -C $(KDIR) M=$(PWD) modules

# run: Carregar o módulo e verificar o log
run: $(MODULE_NAME).ko
    sudo insmod $(MODULE_NAME).ko
    dmesg | tail

# clean: Limpa os arquivos gerados pela compilação
clean:
    make -C $(KDIR) M=$(PWD) clean
    rm -f *.zip

# zip: Zippa os arquivos fonte e outros necessários
zip: $(ZIP_NAME).zip

$(ZIP_NAME).zip: $(FILES)
    zip -r $(ZIP_NAME).zip $(FILES)
```

```

unzip -q $(ZIP_NAME).zip -d $(ZIP_NAME)

# unload: Remove o módulo do kernel
unload:
    sudo rmmod $(MODULE_NAME)
    dmesg | tail

# finish: Compila, carrega, remove e zippa
finish: all run unload clean zip

.PHONY: all clean run zip unload finish

```

Em seguida foi criado o arquivo `simple.c` com o mesmo código do livro:

```

#include <linux/init.h>
#include <linux/kernel.h>
#include <linux/module.h>

/* Esta função é chamada ao carregar o módulo */
int simple_init(void) {
    printk(KERN_INFO "Loading Module\n");
    return 0;
}

/* Esta função é chamada ao remover o módulo */
void simple_exit(void) {
    printk(KERN_INFO "Removing Module\n");
}

module_init(simple_init);
module_exit(simple_exit);

MODULE_LICENSE("GPL");
MODULE_DESCRIPTION("Simple Module");
MODULE_AUTHOR("SGG");

```

Em seguida foi executado `make` para compilar o módulo e teve a seguinte saída:

```

~/Workspace/SistemasOperacionais/ModuloKernel$ make
make -C /lib/modules/6.8.0-49-generic/build
M=/home/lilo/Workspace/SistemasOperacionais/ModuloKernel modules
make[1]: Entrando no diretório '/usr/src/linux-headers-6.8.0-49-generic'
warning: the compiler differs from the one used to build the kernel
The kernel was built by: x86_64-linux-gnu-gcc (Ubuntu 13.2.0-23ubuntu4)
13.2.0
You are using: gcc-13 (Ubuntu 13.2.0-23ubuntu4) 13.2.0
CC [M] /home/lilo/Workspace/SistemasOperacionais/ModuloKernel/simple.o
/home/lilo/Workspace/SistemasOperacionais/ModuloKernel/simple.c:6:5:
warning: no previous prototype for 'simple_init' [-Wmissing-prototypes]

```

```

6 | int simple_init(void) {
  | ^~~~~~
/home/lilo/Workspace/SistemasOperacionais/ModuloKernel/simple.c:12:6:
warning: no previous prototype for 'simple_exit' [-Wmissing-prototypes]
12 | void simple_exit(void) {
    | ^~~~~~
MODPOST
/home/lilo/Workspace/SistemasOperacionais/ModuloKernel/Module.symvers
CC [M]
/home/lilo/Workspace/SistemasOperacionais/ModuloKernel/simple.mod.o
LD [M] /home/lilo/Workspace/SistemasOperacionais/ModuloKernel/simple.ko
make[1]: Saindo do diretório '/usr/src/linux-headers-6.8.0-49-generic'

```

Antes de iniciar o módulo foi executado um **lsmod** para verificar os módulos ativos:

Module	Size	Used by
snd_seq_dummy	12288	0
snd_hrtimer	12288	1
vboxsf	45056	0
vboxguest	57344	7 vboxsf
vboxvideo	36864	0
drm_vram_helper	24576	1 vboxvideo
qrtr	53248	2
ip6t_REJECT	12288	1
nf_reject_ipv6	24576	1 ip6t_REJECT
xt_hl	12288	22
ip6t_rt	16384	3
ipt_REJECT	12288	1
nf_reject_ipv4	12288	1 ipt_REJECT
xt_LOG	16384	10
nf_log_syslog	20480	10
nft_limit	16384	13
xt_limit	12288	0
xt_addrtype	12288	4
xt_tcpudp	16384	60
xt_conntrack	12288	16
nf_conntrack	196608	1 xt_conntrack
nf_defrag_ipv6	24576	1 nf_conntrack
nf_defrag_ipv4	12288	1 nf_conntrack
nft_compat	20480	117
nf_tables	372736	594 nft_compat,nft_limit
binfmt_misc	24576	1
zfs	6598656	6
spl	180224	1 zfs
nls_iso8859_1	12288	1
intel_rapl_msr	20480	0
intel_rapl_common	40960	1 intel_rapl_msr
snd_intel8x0	53248	1
intel_uncore_frequency_common	16384	0
snd_ac97_codec	196608	1 snd_intel8x0
ac97_bus	12288	1 snd_ac97_codec
intel_pmc_core	118784	0

intel_vsec	20480	1	intel_pmc_core
snd_pcm	192512	2	snd_intel8x0,snd_ac97_codec
pmt_telemetry	16384	1	intel_pmc_core
pmt_class	12288	1	pmt_telemetry
snd_seq_midi	24576	0	
snd_seq_midi_event	16384	1	snd_seq_midi
snd_rawmidi	57344	1	snd_seq_midi
snd_seq	114688	9	
snd_seq_midi,snd_seq_midi_event,snd_seq_dummy			
snd_seq_device	16384	3	snd_seq,snd_seq_midi,snd_rawmidi
snd_timer	49152	3	snd_seq,snd_hrtimer,snd_pcm
rapl	20480	0	
snd	143360	11	
snd_seq,snd_seq_device,snd_intel8x0,snd_timer,snd_ac97_codec,snd_pcm,snd_rawmidi			
joydev	32768	0	
i2c_piix4	32768	0	
soundcore	16384	1	snd
input_leds	12288	0	
serio_raw	20480	0	
mac_hid	12288	0	
sch_fq_codel	24576	2	
vmwgfx	438272	3	
drm_ttm_helper	12288	3	vmwgfx,drm_vram_helper,vboxvideo
ttm	110592	3	vmwgfx,drm_vram_helper,drm_ttm_helper
msr	12288	0	
parport_pc	53248	0	
ppdev	24576	0	
lp	28672	0	
parport	73728	3	parport_pc,lp,ppdev
efi_pstore	12288	0	
nfnetlink	20480	3	nft_compat,nf_tables
dmi_sysfs	24576	0	
ip_tables	32768	0	
x_tables	65536	11	
xt_conntrack,nft_compat,xt_LOG,xt_tcpudp,xt_addrtype,ip6t_rt,ipt_REJECT,ip_tables,xt_limit,xt_hl,ip6t_REJECT			
autofs4	57344	2	
btrfs	2019328	0	
blake2b_generic	24576	0	
raid10	73728	0	
raid456	192512	0	
async_raid6_recov	20480	1	raid456
async_memcpy	16384	2	raid456,async_raid6_recov
async_pq	20480	2	raid456,async_raid6_recov
async_xor	16384	3	async_pq,raid456,async_raid6_recov
async_tx	16384	5	
async_pq,async_memcpy,async_xor,raid456,async_raid6_recov			
xor	20480	2	async_xor,btrfs
raid6_pq	126976	4	async_pq,btrfs,raid456,async_raid6_recov
libcrc32c	12288	4	nf_conntrack,btrfs,nf_tables,raid456
raid1	57344	0	
raid0	24576	0	
dm_mirror	24576	0	

dm_region_hash	24576	1	dm_mirror
dm_log	20480	2	dm_region_hash, dm_mirror
crct10dif_pclmul	12288	1	
crc32_pclmul	12288	0	
hid_generic	12288	0	
polyval_generic	12288	0	
usbhid	77824	0	
ghash_clmulni_intel	16384	0	
sha256_ssse3	32768	0	
hid	180224	2	usbhid, hid_generic
sha1_ssse3	32768	0	
psmouse	217088	0	
ahci	49152	2	
e1000	180224	0	
libahci	53248	1	ahci
pata_acpi	12288	0	
video	73728	0	
wmi	28672	1	video
aesni_intel	356352	0	
crypto_simd	16384	1	aesni_intel
cryptd	24576	2	crypto_simd, ghash_clmulni_intel

Em seguida, foi feita uma tentativa de **iniciar** o módulo do kernel:

```
~/Workspace/SistemasOperacionais/ModuloKernel$ make run
sudo insmod simple.ko
insmod: ERROR: could not insert module simple.ko: Invalid module format
make: *** [Makefile:28: run] Erro 1
```

Sem sucesso na tentativa de iniciar o módulo, foi feita uma nova tentativa de **compilar** o módulo, passando o compilador correto como variável de ambiente:

```
~/Workspace/SistemasOperacionais/ModuloKernel$ make CC=/usr/bin/x86_64-
linux-gnu-gcc
make -C /lib/modules/6.8.0-49-generic/build
M=/home/lilo/Workspace/SistemasOperacionais/ModuloKernel modules
make[1]: Entrando no diretório '/usr/src/linux-headers-6.8.0-49-generic'
CC [M] /home/lilo/Workspace/SistemasOperacionais/ModuloKernel/simple.o
/home/lilo/Workspace/SistemasOperacionais/ModuloKernel/simple.c:6:5:
warning: no previous prototype for 'simple_init' [-Wmissing-prototypes]
   6 | int simple_init(void) {
     |           ^~~~~~
/home/lilo/Workspace/SistemasOperacionais/ModuloKernel/simple.c:12:6:
warning: no previous prototype for 'simple_exit' [-Wmissing-prototypes]
   12 | void simple_exit(void) {
     |           ^~~~~~
MODPOST
/home/lilo/Workspace/SistemasOperacionais/ModuloKernel/Module.symvers
CC [M]
```

```
/home/lilo/Workspace/SistemasOperacionais/ModuloKernel/simple.mod.o
LD [M] /home/lilo/Workspace/SistemasOperacionais/ModuloKernel/simple.ko
make[1]: Saindo do diretório '/usr/src/linux-headers-6.8.0-49-generic'
```

Embora a mensagem de *warning* não tenha mais aparecido, o erro persiste ao tentar **iniciar** o módulo: **ERROR: could not insert module simple.ko: Invalid module format**. Tentou-se **desinstalar o gcc-13** para garantir que o compilador correto estivesse sendo usado, mas o erro persistiu.

Diversas informações sobre o arquivo **simple.ko** e o kernel foram verificadas:

```
~/Workspace/SistemasOperacionais/ModuloKernel$ uname -r
6.8.0-49-generic
~/Workspace/SistemasOperacionais/ModuloKernel$ file simple.ko
simple.ko: ELF 64-bit LSB relocatable, x86-64, version 1 (SYSV),
BuildID[sha1]=fb665fcda77c06957c38eb836bb34c350aabc18, with debug_info,
not stripped
~/Workspace/SistemasOperacionais/ModuloKernel$ modinfo simple.ko
filename:
/home/lilo/Workspace/SistemasOperacionais/ModuloKernel/simple.ko
author:      SGG
description:  Simple Module
license:     GPL
srcversion:  F30DB6AF5D0866D751E7B2B
depends:
retpoline:   Y
name:        simple
vermagic:    6.8.0-49-generic SMP preempt mod_unload modversion
```

As permissões do arquivo **simple.ko** foram alteradas para **755**, mas o erro persistiu.

Verificou-se também se o SO permitia o carregamento de módulos, e confirmou-se que sim:

```
~/Workspace/SistemasOperacionais/ModuloKernel$ cat /boot/config-$(uname -r)
| grep CONFIG_MODVERSIONS
CONFIG_MODVERSIONS=y
```

Mesmo assim, o erro persistiu.

Uma nova tentativa foi feita com o uso do **dmesg** para verificar o log do módulo:

```
~/Workspace/SistemasOperacionais/ModuloKernel$ sudo insmod simple.ko
dmesg | tail -n 20
insmod: ERROR: could not insert module simple.ko: Invalid module format
[ 4530.083164] module simple: .gnu.linkonce.this_module section size must
match the kernel's built struct module size at run time
[ 5002.429644] module simple: .gnu.linkonce.this_module section size must
match the kernel's built struct module size at run time
```

```

[ 5225.404716] module simple: .gnu.linkonce.this_module section size must
match the kernel's built struct module size at run time
[ 5225.666381] workqueue: vmstat_shepherd hogged CPU for >10000us 128
times, consider switching to WQ_UNBOUND
[ 5285.151905] module simple: .gnu.linkonce.this_module section size must
match the kernel's built struct module size at run time
[ 5410.840230] module simple: .gnu.linkonce.this_module section size must
match the kernel's built struct module size at run time
[ 5534.970692] workqueue: ata_sff_pio_task hogged CPU for >10000us 64
times, consider switching to WQ_UNBOUND
[ 5553.701209] module simple: .gnu.linkonce.this_module section size must
match the kernel's built struct module size at run time
[ 5628.457237] workqueue: blk_mq_requeue_work hogged CPU for >10000us 128
times, consider switching to WQ_UNBOUND
[ 5921.399859] workqueue: process_srcu hogged CPU for >10000us 64 times,
consider switching to WQ_UNBOUND
[ 5951.552165] module simple: .gnu.linkonce.this_module section size must
match the kernel's built struct module size at run time
[ 6749.861517] module simple: .gnu.linkonce.this_module section size must
match the kernel's built struct module size at run time
[ 7340.480939] module simple: .gnu.linkonce.this_module section size must
match the kernel's built struct module size at run time
[ 7463.485722] module simple: .gnu.linkonce.this_module section size must
match the kernel's built struct module size at run time
[ 7578.809289] workqueue: vmstat_update hogged CPU for >10000us 4 times,
consider switching to WQ_UNBOUND
[ 7869.600501] workqueue: e1000_watchdog [e1000] hogged CPU for >10000us
512 times, consider switching to WQ_UNBOUND
[ 7921.713408] module simple: .gnu.linkonce.this_module section size must
match the kernel's built struct module size at run time
[ 8119.143684] module simple: .gnu.linkonce.this_module section size must
match the kernel's built struct module size at run time
[ 8582.132384] module simple: .gnu.linkonce.this_module section size must
match the kernel's built struct module size at run time
[ 8615.347669] module simple: .gnu.linkonce.this_module section size must
match the kernel's built struct module size at run time

```

Foi tentando também compilar com permissões de super usuário, porém novos erros apareceram e ele nem compilou:

```

~/Workspace/SistemasOperacionais/ModuloKernel$ sudo make
CC=/usr/bin/x86_64-linux-gnu-gcc
[sudo] senha para lilo:
make -C /lib/modules/6.8.0-49-generic/build M= modules
make[1]: Entrando no diretório '/usr/src/linux-headers-6.8.0-49-generic'
make[3]: *** Sem regra para processar o alvo
'arch/x86/entry/syscalls/syscall_32.tbl', necessário por
'arch/x86/include/generated/uapi/asm/unistd_32.h'. Pare.
make[2]: *** [arch/x86/Makefile:249: archheaders] Erro 2
make[1]: *** [Makefile:240: __sub-make] Erro 2

```

```
make[1]: Saindo do diretório '/usr/src/linux-headers-6.8.0-49-generic'  
make: *** [Makefile:24: simple.ko] Erro 2
```

Uma tentativa foi feita para reconfigurar as configurações do kernel utilizando o **oldconfig**:

```
/usr/src/linux-headers-6.8.0-49-generic$ sudo make oldconfig  
#  
# configuration written to .config  
#
```

Porém, mesmo assim, o erro persistiu.

Por fim, tentou-se recompilar o kernel usando os seguintes comandos:

```
cd ~  
mkdir kernel_source  
cd kernel_source  
wget https://cdn.kernel.org/pub/linux/kernel/v6.x/linux-6.1.47.tar.xz  
tar -xvf linux-6.1.47.tar.xz  
cd linux-6.1.47  
cp /boot/config-$(uname -r) .config  
make -j$(nproc)  
...  
CC      kernel/context_tracking.o  
CC      kernel/iomem.o  
CC      kernel/rseq.o  
CC      kernel/watch_queue.o  
CHK      kernel/kheaders_data.tar.xz  
GEN      kernel/kheaders_data.tar.xz  
AR      kernel/built-in.a  
CC [M]   kernel/kheaders.o  
make: *** [Makefile:2012: .] Erro 2
```

A compilação do kernel falhou devido a outro erro.

No fim, não foi possível carregar o módulo do kernel.

## Parte II

A parte II envolve fazer um incremento ao código anterior da seguinte forma:

```
#include <linux/init.h>  
#include <linux/module.h>  
#include <linux/kernel.h>  
#include <linux/list.h>  
#include <linux/slab.h>
```



```

struct birthday {
    int day;
    int month;
    int year;
    struct list_head list;
};

static LIST_HEAD(birthday_list);

int simple_init(void) {
    struct birthday *person;
    int i;

    printk(KERN_INFO "Loading Module\n");

    for (i = 1; i <= 5; i++) {
        person = kmalloc(sizeof(*person), GFP_KERNEL);
        person->day = i;
        person->month = i + 1;
        person->year = 2000 + i;
        INIT_LIST_HEAD(&person->list);
        list_add_tail(&person->list, &birthday_list);
    }

    list_for_each_entry(person, &birthday_list, list) {
        printk(KERN_INFO "Birthday: %d/%d/%d\n", person->day, person->month, person->year);
    }
    return 0;
}

void simple_exit(void) {
    struct birthday *ptr, *next;

    list_for_each_entry_safe(ptr, next, &birthday_list, list) {
        list_del(&ptr->list);
        kfree(ptr);
    }

    printk(KERN_INFO "Removing Module and Freeing Memory\n");
}

module_init(simple_init);
module_exit(simple_exit);

MODULE_LICENSE("GPL");
MODULE_DESCRIPTION("Linked List Kernel Module");
MODULE_AUTHOR("SGG");

```

E consistiria em realizar os mesmos testes da **Parte I**, com a diferença de que agora o kernel está utilizando uma lista ligada para armazenar datas de aniversário.