Para ver la respuesta: **#dmesg(no uses el dmsg o tail -f /var/log/syslog :v)**
Recomendaciones: subir los archivos a github, o averiguar como pasar archivos mediante ssh.

Configurando servidor ssh

```
VBoxManage list vms -l
VBoxManage startvm "jessie32"
```

1. Maquina Virtual → Configuracion→ Red → Adaptador Solo anfitrion → nombre: vboxnet0
    a. Si no posee vboxnet0 habilitarlo
2. #cd /etc/network
3. # nano interfaces

```
source /etc/network/interfaces.d/*
# The loopback network interface
auto lo
iface lo inet loopback

allow-hotplug eth0
iface eth0 inet dhcp

allow-hotplug eth1
iface eth1 inet static
address 192.168.56.3
netmask 255.255.255.0
```

4. #/etc/init.d/networking stop
5. #/etc/init.d/networking start
6. #ifup eth0
7. #ifup eth1
8. Desde el sistema invitado conectarse al servidor ssh -p 22 usuario@192.168.56.3

Configurando el ambiente de trabajo.

Descargar el kernel 3.16 de linux en el directorio /usr/src
cd /usr/src
wget https://cdn.kernel.org/pub/linux/kernel/v3.x/linux-3.16.49.tar.xz
tar Jxvf linux-3.16.49.tar.xz
cd /usr/src/linux-source-3.16/arch/x86/syscalls

#nano syscall_32.tbl

```
357     i386    igmp                    sys_igmp
```

#cd /usr/src/linux-3.16.49/kernel
#nano igpm.c

Hacemos el programa
#nano Makefile
agregar en el inicio :

```
obj-y           = fork.o exec_domain.o panic.o \
                  cpu.o exit.o itimer.o time.o softirq.o resource.o \
                  sysctl.o sysctl_binary.o capability.o ptrace.o timer.o user.o \
                  signal.o sys.o kmod.o workqueue.o pid.o task_work.o \
                  extable.o params.o posix-timers.o \
                  kthread.o sys_ni.o posix-cpu-timers.o \
                  hrtimer.o nsproxy.o \
                  notifier.o ksysfs.o cred.o reboot.o \
                  async.o range.o groups.o smpboot.o igmp.o
```

#cd /usr/src/linux-source-3.16/include/linux

#nano syscalls.h

Al final del archivo agregamos esta linea:

```
asmlinkage long sys_igmp(int  *vector,  int  *mayor);
```

Compilando e instalando kernels
#make localmodconfig
#make-kpkg --initrd --append-to-version=-escribir-lo-que-sea kernel-image kernel-headers
#dpkg -i *.deb
dpkg -l | grep linux-image //aqui vemos qué imagen tenemos instalada
#mkinitramfs -k -o /boot/initrd.img-2.6.32.15+drm33.5 2.6.32.15+drm33.5
#update-initramfs -uk all

Borrar Kernels
#dpkg --get-selections | grep linux-image o dpkg -l | grep linux-image
#apt-get remove --purge linux-image-X.X.X-X

Fichero: igmp.c //ubicacion → /usr/src/linux-source-3.16/kernel/igmp.c

```
#include <linux/kernel.h>
#include <linux/linkage.h>
#include <linux/syscalls.h>

asmlinkage long sys_igmp(int *vector, int *mayor){
```

```
        int i=0;
        for(i; i<10; i++){
                if(*vector > *mayor){
                        *mayor = *vector;
                }
        vector++;
        }

        printk(KERN_ALERT "El numero mayor es: %d\n", *mayor);

        return *mayor;
}
```

Fichero: usuario-igmp.c //ubicacion → donde quieras

```
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>

int main(void){

  int vector[10];
  int i=0;

  int num;
  int ch;
  int ok;

for(i; i<10; i++){

 do{
     printf("Ingresa un entero posicion [%i]   : ", i+1);
     fflush(stdout);
     if ((ok = scanf("%d", &vector[i])) == EOF)
       return -1;

     if ((ch = getchar()) != '\n'){
       ok = 0;

       while ((ch = getchar()) != EOF && ch != '\n')
```

```
            ;
        }
    }while (!ok);
}

        long int mayor = syscall(357,vector,&mayor);
        printf("El numero mayor es: %d \n", mayor);

return 0;
}
```