

Trabajo práctico

Sistemas Operativos

18 de agosto de 2008

Índice

| | |
|--|-----------|
| 1. Introducción | 3 |
| 1.1. Objetivos | 3 |
| 1.2. Normas de entrega | 3 |
| 1.3. Formato de entrega | 3 |
| 2. Instalación | 4 |
| 3. Consignas | 19 |
| 3.1. Comandos básicos de Unix | 19 |
| 3.2. Salida estándar y pipes | 21 |
| 3.3. Scripting | 21 |
| 3.4. Ejecución de procesos en background | 22 |
| 3.4.1. loop.c | 22 |
| 3.5. IPC y sincronización | 23 |
| 3.5.1. Pipes | 23 |
| 3.5.2. Threads | 23 |
| 3.6. El kernel Linux | 23 |
| 3.6.1. Funcionamiento del kernel Linux | 23 |
| 3.6.2. Módulos de kernel | 24 |
| 3.6.3. Compilando un módulo de kernel | 24 |
| 3.6.4. hello.c | 24 |

| | |
|-----------------------------------|----|
| 3.6.5. Makefile | 25 |
| 3.6.6. Un módulo propio | 25 |
| 3.6.7. check_kbleds.c | 25 |

1. Introducción

1.1. Objetivos

Al terminar este trabajo Ud. habrá aprendido a:

1. Instalar Linux sobre un sistema de virtualización.
2. Utilizar los principales comandos de Linux.
3. Compilar y ejecutar programas escritos en lenguaje C.
4. Modificar el kernel de Linux utilizando módulos.

1.2. Normas de entrega

1. La fecha y hora de entrega para este trabajo práctico es la que figura en el cronograma de la materia. Se alienta y acepta la entrega del trabajo, en su totalidad, en forma anticipada.
2. Los trabajos deben ser entregados **personalmente** a alguno de los docentes de la cátedra en los horarios de clase o de consulta. No se aceptarán trabajos depositados en otro lugar.
3. No se aceptarán trabajos incompletos.

1.3. Formato de entrega

Se deberá entregar la imagen de disco utilizada en el sistema de virtualización, así como los archivos que resuelven las consignas, **fuera** de la imagen, para poder ser eventualmente revisados por separado.

Se deberá entregar además un **documento impreso**. Ese documento debe reunir las siguientes características:

1. Formato de presentación:
 - a) Impreso en hojas de tamaño A4 encarpetadas.
2. Secciones obligatorias del documento:
 - a) Carátula:
 - 1) Signatura
 - 2) Número y descripción del trabajo práctico.
 - 3) Año y cuatrimestre de cursada.
 - 4) Identificación del grupo.
 - 5) Nombre, apellido y dirección de correo electrónico de **todos** los integrantes del grupo.
 - b) Sección principal: Aquí debe incluirse la resolución de cada uno de los problemas planteados. Para cada respuesta debe indicarse el número y título del problema al que corresponde tal como aparece en el enunciado y los comandos y/o programas utilizados para resolverlo. Se deberá indicar claramente en qué directorio y bajo que nombre se encuentran los fuentes, los ejecutables y los programas de prueba, en caso de haberlos.



Figura 1: Creando una nueva máquina virtual.

2. Instalación

Instalaremos una versión liviana de Ubuntu Linux en el sistema de virtualización VirtualBox. VirtualBox nos provee una *máquina virtual* que permite instalar un sistema operativo (al que llamaremos **guest**) en un archivo del disco rígido y ejecutarlo dentro de una ventana en nuestra máquina real. La ventaja de VirtualBox comparado con Bochs o QEMU es que brinda un rendimiento muy superior. VirtualBox es multiplataforma por lo tanto es posible usarlo tanto en Windows (XP o superior) o Linux.

1. Baje la ISO de Ubuntu JeOS 8.04 de:
<http://cdimages.ubuntu.com/jeos/releases/8.04/release/jeos-8.04.1-jeos-i386.iso>
Para usar BitTorrent:
<http://cdimages.ubuntu.com/jeos/releases/8.04/release/jeos-8.04.1-jeos-i386.iso.torrent>
2. Baje VirtualBox de <http://www.sun.com/software/products/virtualbox/get.jsp>.
Recuerde elegir la versión para su sistema operativo.
3. Instale VirtualBox de acuerdo a su sistema operativo.
4. Ejecute VirtualBox. Seleccione la opción de crear una nueva máquina virtual (figura 1).
5. Ingrese un nombre y seleccione Ubuntu como sistema operativo (figura 2).
6. Seleccione la cantidad de memoria para el sistema operativo guest (figura 3).
7. Ahora deberá crear un disco rígido virtual, haciendo click en **New...** (figura 4).
8. Se abre el asistente para crear nuevos discos rígidos virtuales (figura 5). La siguiente pantalla permite especificar si el archivo tendrá desde el principio el tamaño total del disco rígido virtual o se irá expandiendo a medida que el guest use más espacio en el disco. Recomendamos usar la primera opción (figura 6).

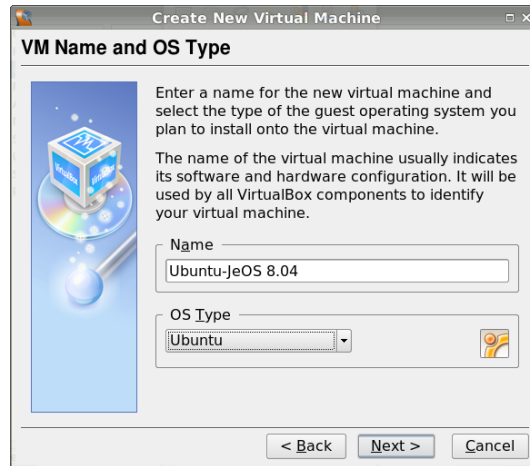


Figura 2: Nombre y tipo de sistema operativo guest.

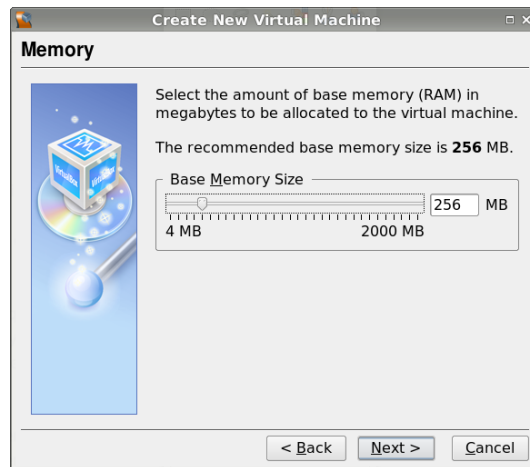


Figura 3: Cantidad de memoria para el guest

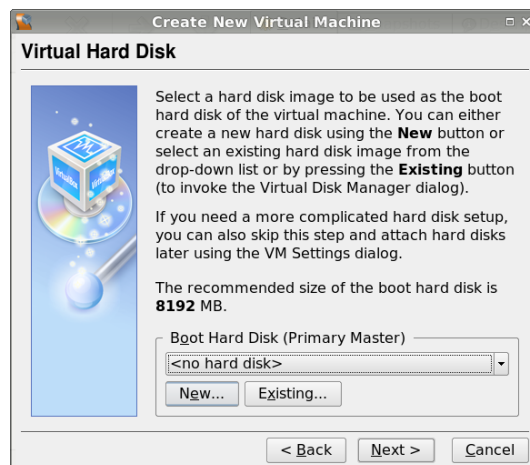


Figura 4: Creando un nuevo disco rígido virtual.



Figura 5: Asistente para discos rígidos virtuales.

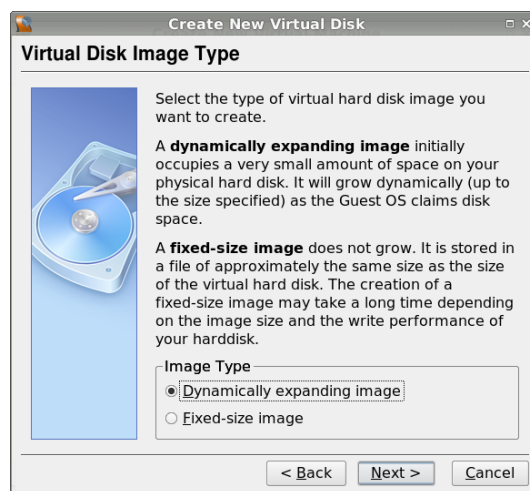


Figura 6: Tipo de disco rígido virtual.

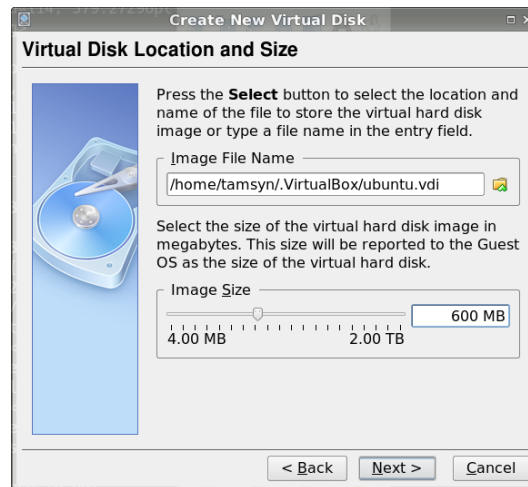


Figura 7: Tamaño del disco rígido virtual.

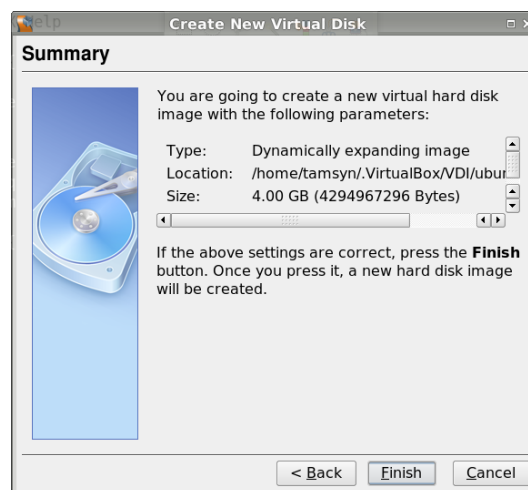


Figura 8: Terminado el proceso de creación del disco rígido virtual.

9. Finalmente, debe seleccionar el tamaño del disco virtual (figura 7). Si se seleccionó la opción **Dynamically expanding image** en la pantalla anterior, el archivo inicialmente pesará pocos megabytes, para luego ir aumentando a medida que el disco se llena. En otro caso, el archivo ocupará el tamaño del disco desde el primer momento. Recomendamos alrededor de 600 MB para el disco, ya que deberá entregar la imagen con el TP. En la siguiente pantalla se pueden revisar los datos del disco rígido virtual y terminar el proceso (figura 8), para volver al asistente principal (figura 9) y terminar la creación de la máquina virtual (figura 10).
10. Para terminar la configuración de la máquina virtual debe especificar que la ISO de Ubuntu-JeOS bajada al comienzo se utilice como CDROM virtual. En la ventana principal de VirtualBox (figura 11) haga click en **CD/DVD-ROM** para ver la configuración de la máquina virtual. VirtualBox seleccionará automáticamente la entrada correspondiente al CD-ROM (figura 12). Haga click en la casilla **Mount CD/DVD drive**, seleccione la opción **ISO Image File** (figura 13) y finalmente seleccione la carpeta a la derecha de la pantalla para especificar el archivo ISO bajado al inicio.
11. Se abrirá el administrador de discos virtuales de VirtualBox, **Virtual Disk Manager** en



Figura 9: El asistente principal con el disco virtual creado.

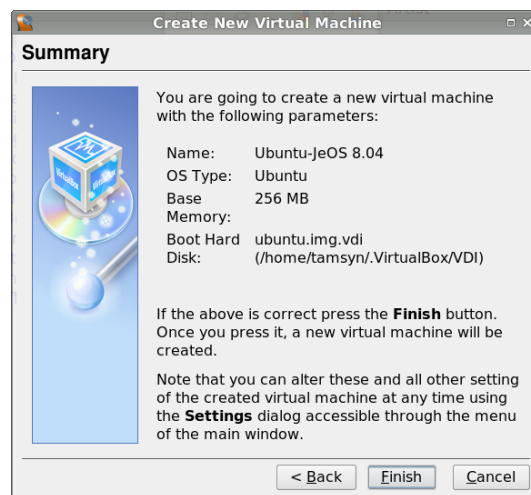


Figura 10: Terminada la creación de la máquina virtual.

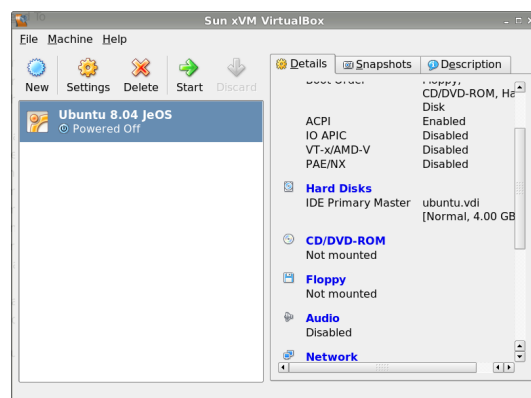


Figura 11: Ventana principal de VirtualBox.

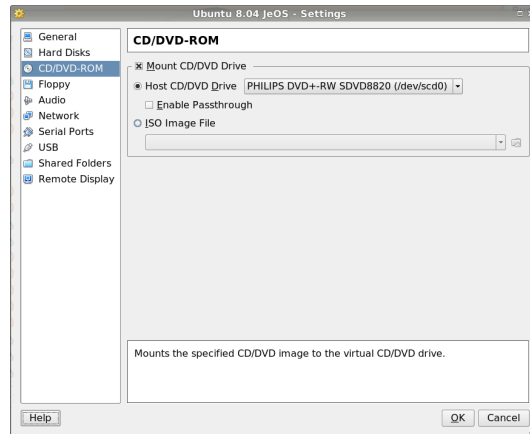


Figura 12: Configuración del CDROM en VirtualBox.

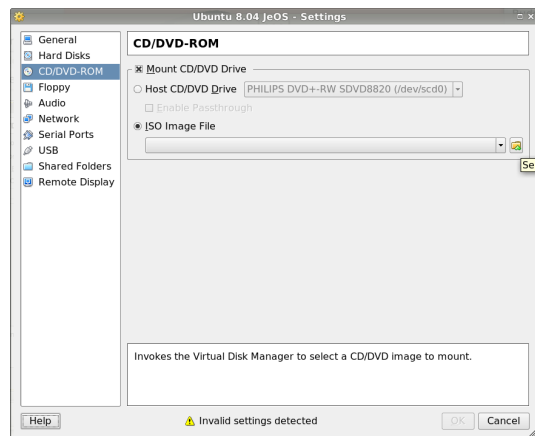


Figura 13: Seleccionando la opción de proveer un archivo ISO.

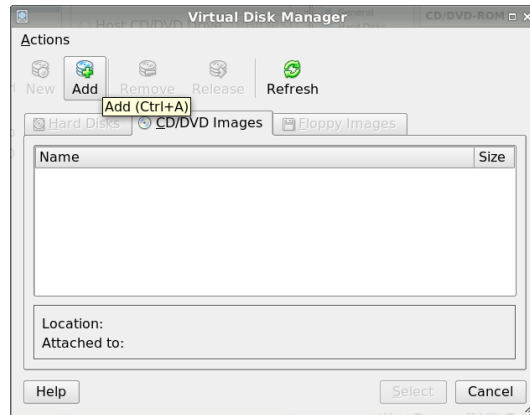


Figura 14: Agregar una ISO.

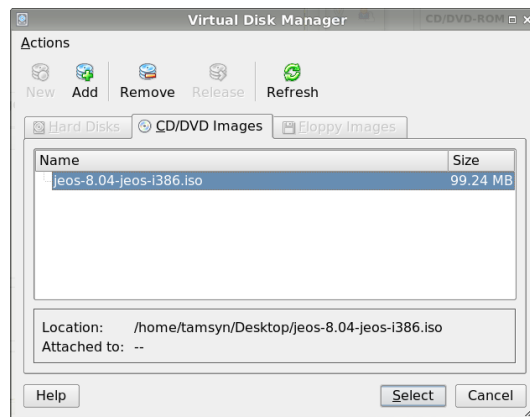


Figura 15: Seleccionando la ISO agregada.

la solapa **CD/DVD Images** (figura 14). Agregue una imagen haciendo click en **Add**, de manera de poder seleccionarla con **Select** (figura 15). De esta manera podrá cerrar la configuración de la máquina virtual (figura 16).

12. Inicie la máquina virtual haciendo doble clic sobre ella o utilizando el botón **Start**. La máquina virtual deberá bootear por defecto desde el CDROM virtual, y de esta manera iniciar la instalación del sistema operativo. El menú de inicio de Ubuntu (previo al inicio propiamente dicho del sistema operativo) le solicitará que seleccione el idioma (figura 17) y luego mostrará el menú (figura 18). Inicie el sistema operativo seleccionando **Install Ubuntu JeOS**

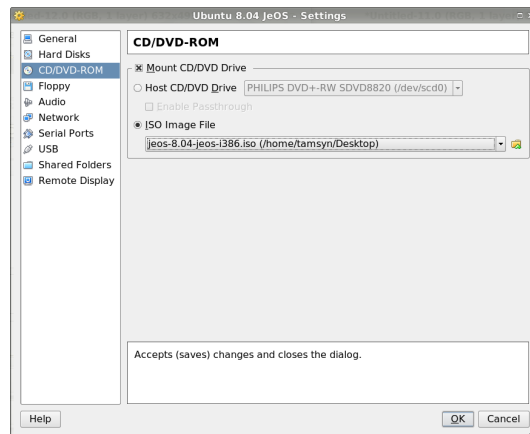


Figura 16: La máquina virtual ya está configurada correctamente.

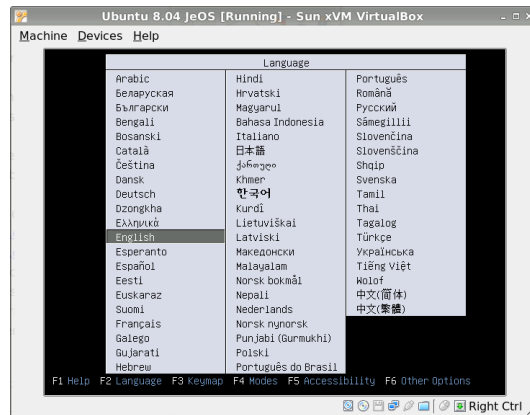


Figura 17: Seleccionar el idioma del instalador.

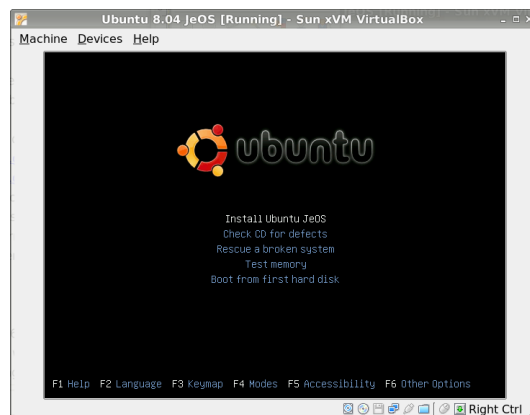


Figura 18: Menu de inicio.

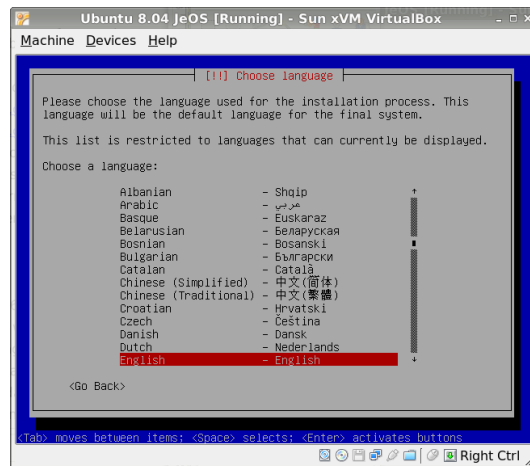


Figura 19: Seleccionar el idioma del sistema.

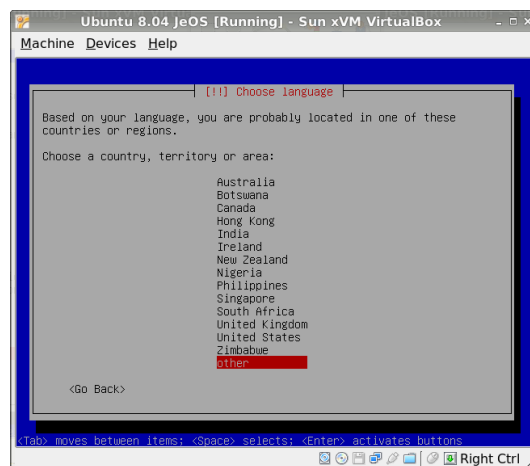


Figura 20: Seleccionando el país.

13. Una vez que haya iniciado el sistema en modo LiveCD, seleccione el idioma que desea utilizar en el sistema (figura 19). Luego deberá elegir su país, primero seleccionando **other** (figura 20) y luego (probablemente) **Argentina** ;-) (figura 21).
14. El siguiente paso es especificar la distribución del teclado. Seleccione la elección manual (figura 22), y luego elija su distribución, probablemente **Latin American** (figura 23) en su versión plana (sin **dead keys** - figura 24).
15. Luego de que el instalador detecte el hardware de la máquina virtual y cargue los componentes a instalar, se configurará la placa de red automáticamente con DHCP. El siguiente paso es configurar el nombre de la máquina virtual (figura 25), puede elegir cualquier nombre ya que es meramente administrativo.
16. El instalador detectará el disco rígido virtual y deberá particionarlo. Aquí empezamos a ver las ventajas de tener una máquina virtual: dado que el disco es también virtual, podrá particionarlo entero sin preocuparse por pisar otro sistema operativo. Seleccione entonces la primera opción (figura 26). En la pantalla que sigue seleccione el disco virtual (figura 27) y confirme la elección en la tercera pantalla (figura 28).
17. Luego de que el instalador termine de particionar el disco virtual procederá a instalar los

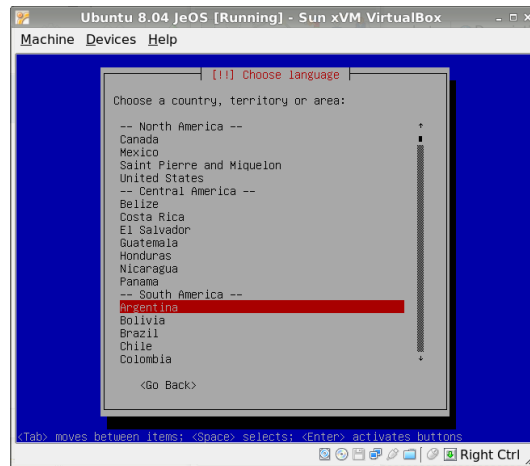


Figura 21: Seleccionando el país.

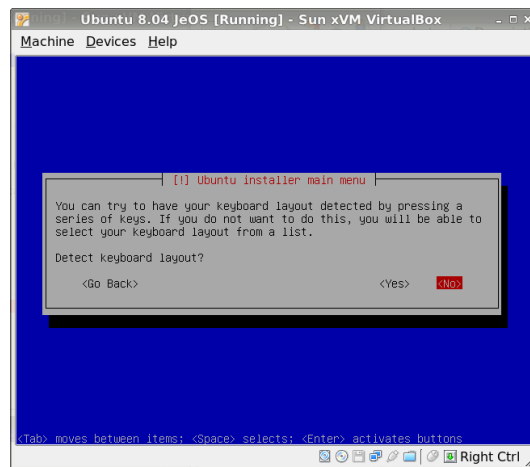


Figura 22: Seleccionando la distribución del teclado.

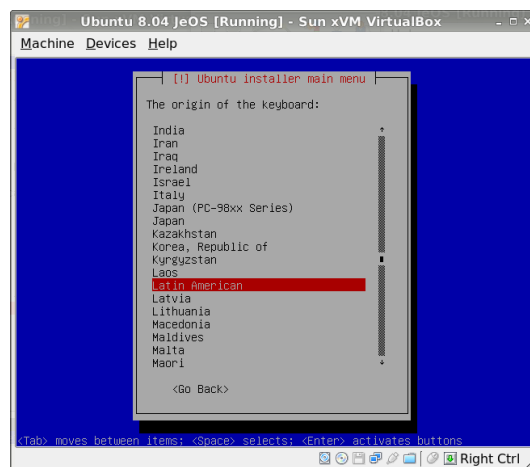


Figura 23: Seleccionando la distribución del teclado.

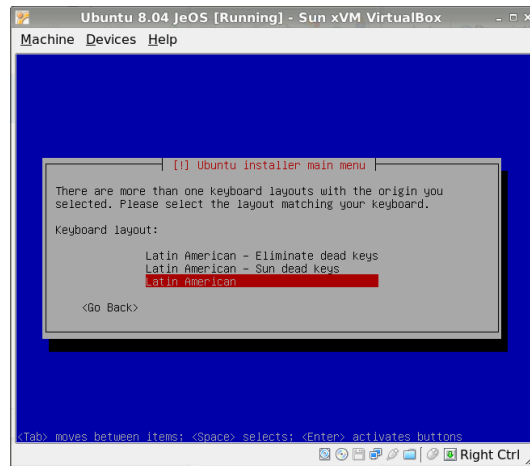


Figura 24: Seleccionando la distribución del teclado.

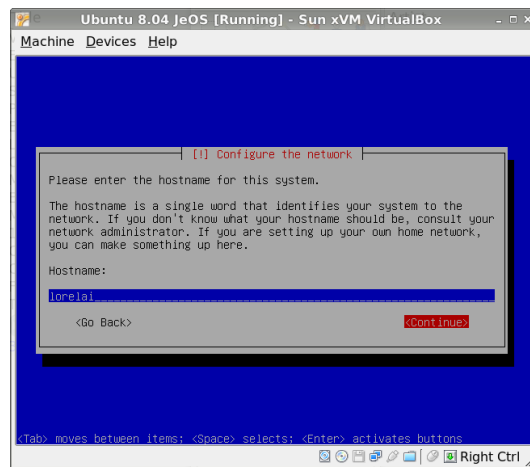


Figura 25: Configurar el nombre de la máquina virtual.

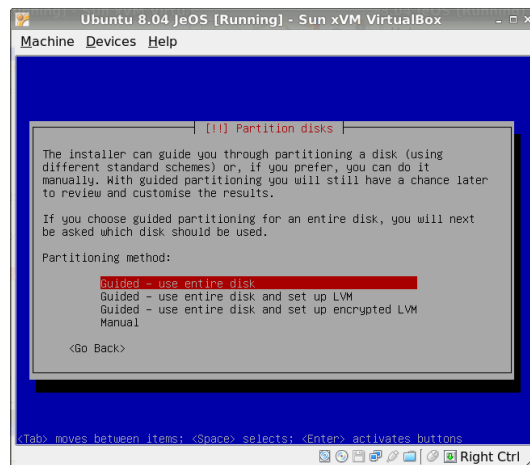


Figura 26: Particionado guiado.

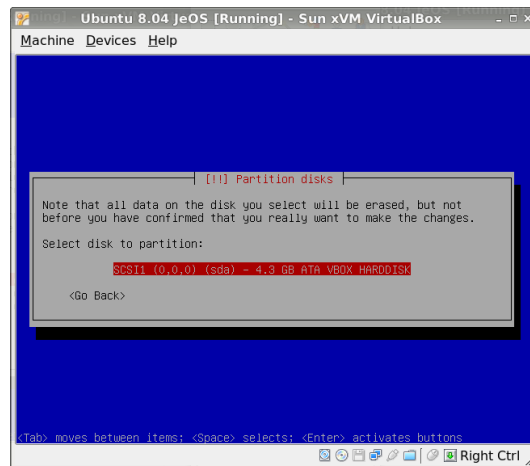


Figura 27: Disco virtual.

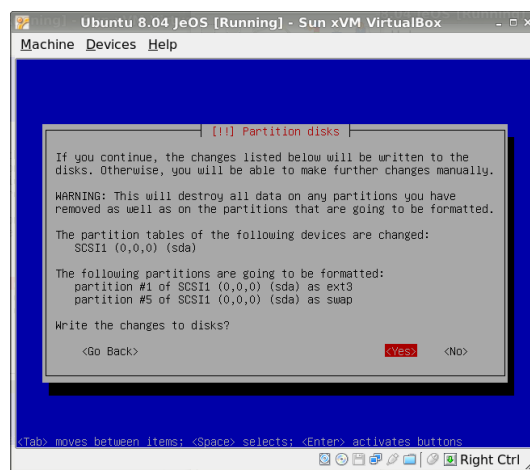


Figura 28: Confirmación.

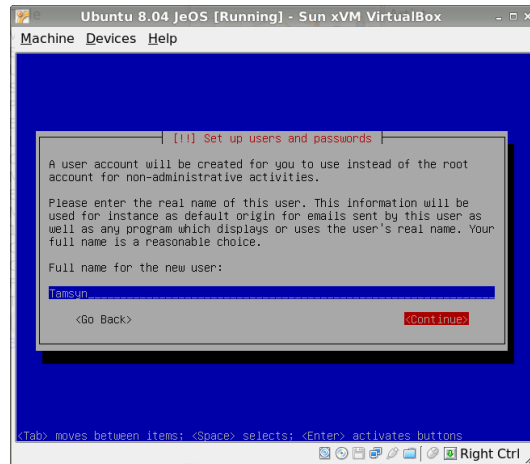


Figura 29: Nombre real del usuario.

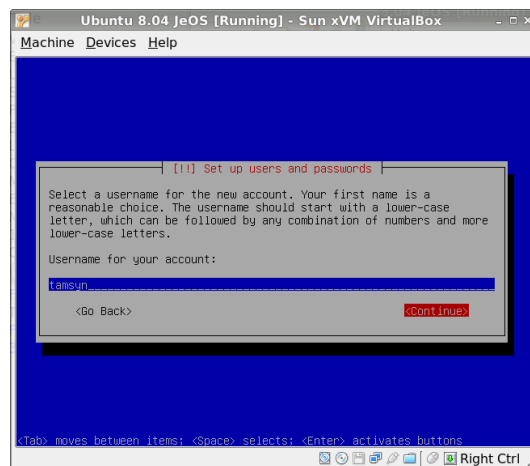


Figura 30: Username del usuario en el sistema.

paquetes básicos de la distribución. Cuando este proceso termine se le pedirá que ingrese el nombre completo del nuevo usuario (figura 29). Esto **no** es el nombre de usuario. El nombre de usuario se especifica en la pantalla siguiente (figura 30). Finalmente, deberá especificar y confirmar el password (figuras 31 y 32).

18. El instalador luego buscará actualizaciones en la web, para lo cual pedirá especificar un proxy, de ser necesario (figura 33). En caso de no necesitar ninguno (lo más probable) deje el campo en blanco. El instalador buscará actualizaciones en la web.
19. Ya terminando la instalación, se le pedirá especificar si el reloj del sistema está configurado en UTC o en la hora local. Nuevamente, como este sistema no comparte la máquina virtual, seleccione UTC (figura 34). La instalación termina y pide reiniciar la máquina virtual (figura 35).
20. Para terminar la configuración de la máquina virtual, desconfigure el archivo ISO como CDROM, de manera inversa a como hizo en un principio: click en CD/DVD-ROM de la pantalla principal, y luego click en **Mount CD/DVD Drive** para desactivar (figura 36). La máquina virtual está lista para ser booteada.

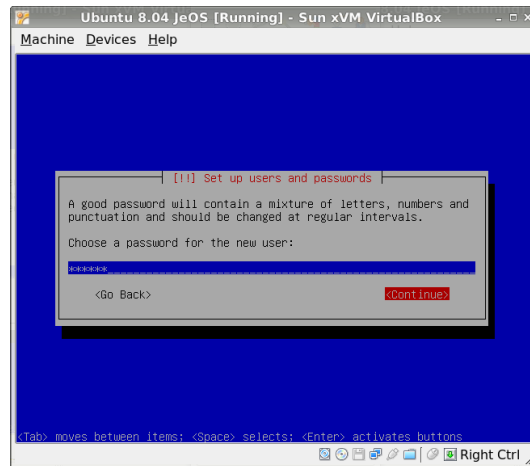


Figura 31: Password.

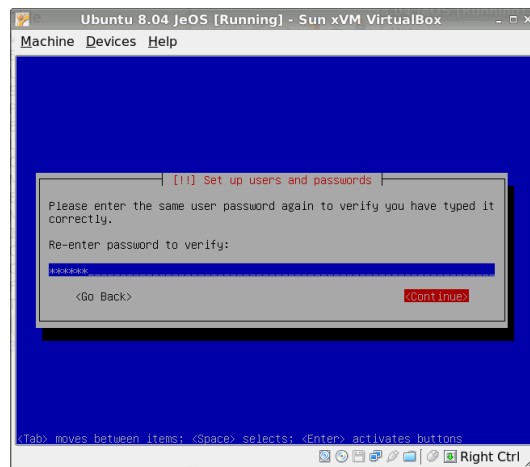


Figura 32: Confirmación del password.

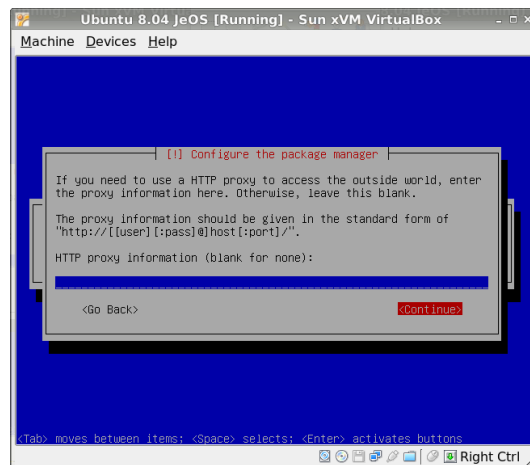


Figura 33: Configuración del proxy.

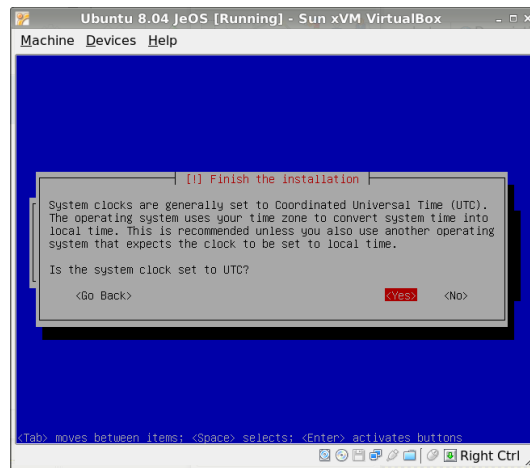


Figura 34: Configuración del reloj del sistema.

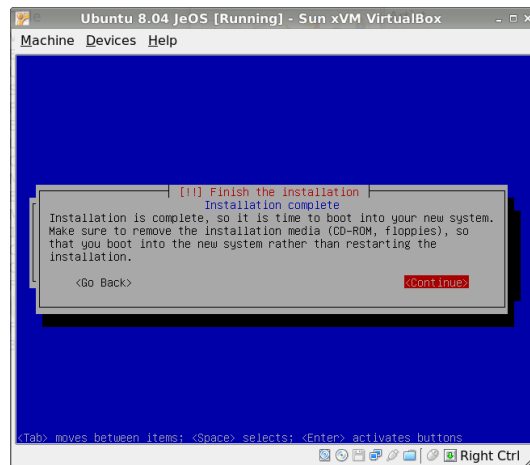


Figura 35: Instalación completada.

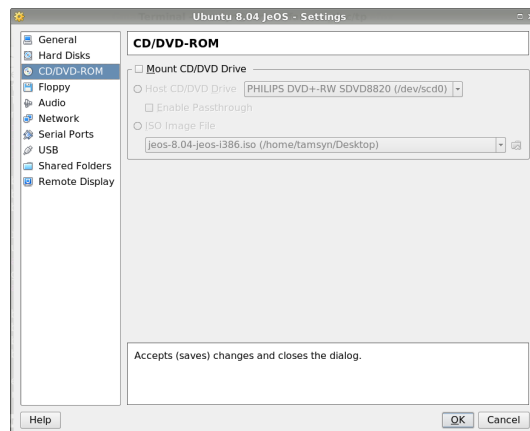


Figura 36: Instalación completada.

3. Consignas

Antes de empezar, ejecute:

```
sudo apt-get install man-db manpages manpages-dev
```

De esta manera tendrá acceso a ayuda en línea ejecutando:

```
man <comando>
```

Por ejemplo:

```
man cp
```

Puede además instalar la versión en castellano de la ayuda ejecutando:

```
sudo apt-get install manpages-es
```

Para acceder a la ayuda en castellano ejecute por ejemplo:

```
man -L es cp
```

Tenga presente que no todos los comandos poseen ayuda en castellano.

sudo permite a usuarios normales ejecutar comandos que requieren permisos de administrador. Al ejecutar un comando con **sudo** el sistema le pedirá su password, y no el password del administrador (llamado **root** en Linux, siguiendo la tradición de Unix). Esto sucede ya que el sistema permite que ciertos usuarios (que deberían corresponderse con usuarios “privilegiados” del sistema) puedan utilizar **sudo** ingresando solamente su propio password. El usuario por defecto creado en una instalación de Ubuntu tiene este permiso y por lo tanto en Ubuntu no es necesario una cuenta de administrador o **root**.

apt-get es el manejador de paquetes de la distribución Ubuntu. Permite instalar, actualizar y desinstalar programas. Más adelante lo utilizaremos para instalar las herramientas necesarias para compilar programas en Linux.

Si se encontrara detrás de un proxy, antes de utilizar **apt-get** debe configurar el proxy. Ejecute el siguiente comando:

```
sudo echo "Acquire::http::Proxy \"http://proxy.uba.ar:8080\";" >/etc/apt/apt.conf
```

3.1. Comandos básicos de Unix

1. **pwd** Indique qué directorio pasa a ser su directorio actual si ejecuta:

a) `cd /usr/bin`

b) `cd`

c) ¿Cómo explica el punto anterior?

2. **cat** ¿Cuál es el contenido del archivo `/home/<usuario>/.profile`?

3. **find** Liste **todos** los archivos que comienzan con **vmlinuz**.

Estos archivos son imágenes del kernel Linux.

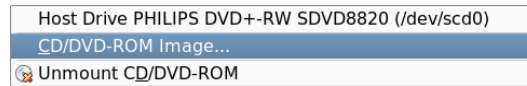


Figura 37: Usando una imagen ISO con la máquina virtual corriendo.

4. **mkdir** Genere un directorio `/home/<usuario>/tp`.
5. **cp** Copie el archivo `/etc/passwd` al directorio `/home/<usuario>/tp`.
6. **chgrp** Cambie el grupo del archivo `/home/<usuario>/tp/passwd` para que sea el suyo.
7. **chown** Cambie el dueño del archivo `/home/<usuario>/tp/passwd` para que sea su usuario.
8. **chmod** Cambie los permisos del archivo `/home/<usuario>/tp/passwd` para que:
 - el propietario tenga permisos de lectura, escritura y ejecución
 - el grupo tenga sólo permisos de lectura y ejecución
 - el resto tenga sólo permisos de ejecución
9. **grep**

Muestre las líneas que tienen el texto “localhost” en el archivo `/etc/hosts`.

Muestre todas las líneas que tengan el texto “POSIX” de todos los archivos (incluyendo subdirectorios) en `/etc`. Evite los archivos binarios y aquellos archivos y directorios que no tienen permiso de lectura para su usuario.
10. **passwd** Cambie su password.
11. **rm** Borre el archivo `/home/<usuario>/tp/passwd`
12. **ln**

Enlazar el archivo `/etc/passwd` a los archivos `/tmp/contra1` y `/tmp/contra2`.

Hacer un `ls -l` para ver cuantos enlaces tiene `/etc/passwd`.

Estos enlaces se llaman “hardlinks”. Cada nuevo enlace referencia el mismo espacio ocupado del disco rígido, y por lo tanto cada hardlink es igual de representativo de esos bytes ocupados del disco rígido. El espacio ocupado solamente se liberará cuando todos los enlaces hayan sido borrados.

Ahora enlace el archivo `/etc/passwd` de manera “soft” al archivo `contra3`.

Verifique con `ls -l` que no aumentó la cantidad de enlaces de `/etc/passwd`.

Estos enlaces se llaman “softlinks” y apuntan no a los bytes del disco rígido sino a la ruta del archivo a ser enlazado. Operar sobre el softlink es igual que operar sobre el archivo, sin embargo los softlinks no cuentan en la cantidad de enlaces (ya que no apuntan a los bytes ocupados del disco rígido) y pueden ser borrados sin afectar al archivo original, aunque si se borra el archivo original el softlink quedará huérfano y no apuntará a nada.
13. **mount**

Monte el CD-ROM de instalación de Ubuntu JeOS y liste su contenido.

Para hacer esto deberá especificar la ISO de instalación de Ubuntu JeOS como CD-ROM de la máquina virtual. Si bien puede hacer esto como lo hizo para instalar el sistema, si la máquina virtual está corriendo debe hacer click derecho en el ícono con forma de CD-ROM en la esquina inferior derecha de la máquina virtual, y seleccionar **CD/DVD-ROM Image...** (ver figura 37). En la ventana que aparece seleccione la ISO de instalación (ver figura 38).

Presente los filesystems que tiene montados.

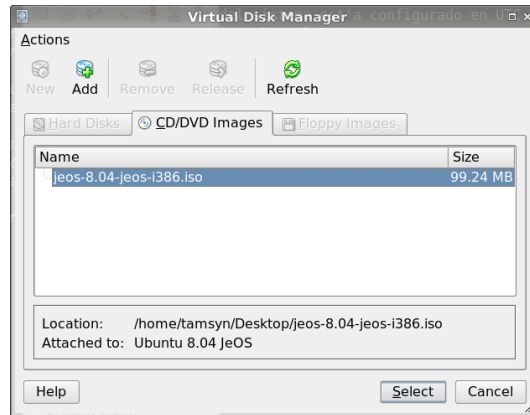


Figura 38: Seleccionando la ISO de instalación.

14. **df** ¿Qué espacio libre tiene cada uno de los filesystems montados?
15. **ps** ¿Cuántos procesos de usuario tiene ejecutando? Indique cuántos son del sistema.
16. **umount** Desmonte el CD-ROM de instalación de Ubuntu JeOS.
17. **uptime** ¿Cuanto tiempo lleva ejecutando su máquina virtual?
18. **uname** ¿Qué versión del kernel de Linux está utilizando?

3.2. Salida estándar y pipes

1. STDOUT

- a) Conserve en el archivo `/home/<usuario>/tp/config` la salida del comando `ls` que muestra todos los archivos del directorio `/etc` y de los subdirectorios bajo `/etc`.
- b) Presente cuantas líneas, palabras y caracteres tiene `/home/<usuario>/tp/config`.
- c) Agregue el contenido, ordenado alfabéticamente, del archivo `/etc/passwd` al final del archivo `/home/<usuario>/tp/config`.
- d) Presente cuantas líneas, palabras y caracteres tiene `/home/<usuario>/tp/config`.

2. Pipes

- a) Liste en forma amplia los archivos del directorio `/usr/bin` que comiencen con la letra "a". Del resultado obtenido, seleccione las líneas que contienen el texto `apt` e informe la cantidad de caracteres, palabras y líneas.
Está prohibido, en este ítem, usar archivos temporales de trabajo.

3.3. Scripting

Escriba un script de shell que sincronice dos carpetas. El script debe recibir las dos carpetas (origen y destino, en ese orden) desde la línea de comando, o preguntarlas interactivamente al usuario en caso de no recibirlas. Además, debe aceptar un modificador `-r` que indica modo de operación recursivo.

Una vez conocidas las dos carpetas con las que se operará, el script deberá copiar todos los archivos de la carpeta origen que no estén presentes en la carpeta destino, y además también deberá copiar todos los archivos presentes en ambas carpetas que tengan una fecha de modificación posterior en la carpeta origen que en la carpeta destino. De esta manera, al ejecutar el script, estará seguro de que la carpeta destino contiene toda la información de la carpeta origen, excepto lo que fue modificado posteriormente en la carpeta destino.

El modificador `-r` indica al script realizar la sincronización también con los archivos de todos los subdirectorios de la carpetas origen y destino.

Sugerencia Recuerde que generalmente el espíritu de los scripts es proveer la mínima lógica necesaria alrededor de otros programas ya presentes en el sistema que puedan proveer parte de la funcionalidad requerida para resolver un determinado problema.

3.4. Ejecución de procesos en background

Antes de resolver esta sección instale los siguientes paquetes en la máquina virtual:

- `nano`: editor de texto.
- `mc`: manejador de archivos.
- `gcc`: compilador de C.
- `libc6-dev`: biblioteca estándar de C.

Cree el archivo `/home/<usuario>/tp/loop.c`. Compílelo con `gcc`. El programa compilado debe llamarse `loop`.

3.4.1. `loop.c`

```
#include <stdio.h>

#define IDGRUPO 10 /* Completar con su numero de grupo */

int main() {
    int i, c;

    while(1) {
        c = 48 + i;
        printf("%d", c);
        i++;
        i = i % IDGRUPO;
    }
    return 0;
}
```

1. Correrlo en foreground. ¿Qué sucede? Mate el proceso con `Ctrl-c`.

2. Ahora ejecútelo en background: `/usr/src/loop >/dev/null &`. Mate el proceso con el comando `kill`.

3.5. IPC y sincronización

3.5.1. Pipes

Muestre con un ejemplo en lenguaje C como realizar exclusión mutua entre dos procesos utilizando un sólo pipe.

Sugerencia Revise la ayuda de la llamada al sistema `pipe` para construir el pipe y de `fork` para crear nuevos procesos.

3.5.2. Threads

Antes de resolver este ejercicio instale el paquete `glibc-doc`.

Resuelva el problema de productor/consumidor utilizando threads.

Sugerencia Revise la ayuda de `pthread`, la implementación de threads en Linux, para conocer los mecanismos de creación y destrucción de threads. Además, `pthread` provee mecanismos de sincronización que le ayudarán a resolver este ejercicio.

3.6. El kernel Linux

Antes de resolver esta sección instale los siguientes paquetes en la máquina virtual:

- `make`: utilidad para mantener grupos de programas.
- `linux-headers-<version>`: headers del kernel de Linux.

Sustituya `<version>` por el resultado del comando `uname -r`.

3.6.1. Funcionamiento del kernel Linux

1. Describa la administración del procesador utilizada por defecto en el kernel Linux.
2. Describa la administración de memoria utilizada por defecto en el kernel Linux.
3. Describa el sistema de archivos utilizado en la distribución de Linux que instaló en la máquina virtual. ¿Qué capas existen en el kernel Linux para soportar sistemas de archivos sobre dispositivos de bloques?

3.6.2. Módulos de kernel

El kernel de Linux permite ser ampliado en **runtime** con módulos. Los módulos son objetos de código compilado que pueden ser insertados en runtime al kernel, siendo linkeados contra el kernel al momento de ser insertados. De esta manera puede ampliarse la funcionalidad del kernel en runtime, sin tener que incluir todo el código en el binario original.

3.6.3. Compilando un módulo de kernel

A continuación compilaremos y probaremos un módulo de kernel muy simple. Este módulo simplemente escribe en la consola “Hola kernel!” al ser insertado y “Chau, kernel.” al ser removido. Cree el siguiente módulo en el archivo `/home/<usuario>/tp/hello.c`.

3.6.4. hello.c

```
/* hello.c
 *
 * "Hello, world" usando modulos de kernel
 *
 */

/* Headers para modulos de kernel */
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/init.h>

/* Prototipos de las funciones de inicializacion y destruccion */
static int __init hello_init(void);
static void __exit hello_exit(void);

/* Informamos al kernel que inicialize el modulo usando hello_init
 * y que antes de quitarlo use hello_exit
 */
module_init(hello_init);
module_exit(hello_exit);

/* Inicializacion */
static int __init hello_init() {
    printk(KERN_ALERT "Hola kernel!\n");
    /* Si devolvemos un valor distinto de cero significa que
     * hello_init fallo y el modulo no puede ser cargado.
     */
    return 0;
}

/* Destruccion */
static void __exit hello_exit() {
    printk(KERN_ALERT "Chau, kernel.\n");
}
```



```
/* Declaramos que este codigo tiene licencia GPL.
 * De esta manera no estamos "manchando" el kernel con codigo propietario.
 */
MODULE_LICENSE("GPL");
```

Para compilar este código deberá construir una **Makefile**. Este archivo Makefile es utilizado luego por el comando **make** para compilar el módulo con las opciones correctas. En el mismo directorio donde se encuentra **hello.c** cree un archivo **Makefile** conteniendo:

3.6.5. Makefile

```
obj-m      = hello.o
KVERSION   = $(shell uname -r)

all:
    make -C /lib/modules/$(KVERSION)/build M=$(PWD) modules
clean:
    make -C /lib/modules/$(KVERSION)/build M=$(PWD) clean
```

Luego ejecute:

```
make
```

Para compilar el módulo. Finalmente, pruebe insertar el módulo usando:

```
insmod hello.ko
```

Debería ver el mensaje “Hola kernel!” en la consola. Quite el módulo usando:

```
rmmmod hello.ko
```

Debería ver el mensaje “Chau, kernel”.

3.6.6. Un módulo propio

Escriba un módulo de kernel que permita controlar los LEDs del teclado **sin necesidad de escribir un programa en lenguaje C**. El módulo deberá permitir prender o apagar cada LED usando simplemente comandos del shell.

Dado que la máquina virtual provista por VirtualBox no muestra el estado de los LEDs, la cátedra provee abajo un programa que lo hace. Compile utilizando:

```
gcc -o check_kbleds check_kbleds.c
```

3.6.7. check_kbleds.c

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

```
#include <fcntl.h>
#include <sys/stat.h>
#include <linux/kd.h> /* IOCTLs del teclado */
#include <sys/ioctl.h> /* Funcion ioctl() */
#include <sys/types.h>

#define ERROR -1

void print_kbleds(long int arg);

int main(void) {
    int fd; /* fd de la consola (/dev/tty), para usar en ioctl() */
    long int state; /* Entero para guardar el estado de los LEDs */

    /* Para usar como fd en ioctl() */
    if ((fd = open("/dev/tty", O_NOCTTY)) == ERROR) {
        perror("open: Couldn't open /dev/tty");
        exit(ERROR);
    }

    /* El estado de los LEDs se guarda en state */
    if (ioctl(fd, KDGETLED, &state) == ERROR) {
        perror("ioctl: Couldn't do KDGETLED IOCTL");
        close(fd);
        exit(ERROR);
    }

    /* Imprimimos el estado de los LEDs */
    print_kbleds(state);
    return 0;
}

void print_kbleds(long int state) {
    /* LED_SCR = 0x1, LED_NUM = 0x2, LED_CAP = 0x4 */
    if (state & LED_SCR) printf("Scroll Lock LED is on.\n");
    if (state & LED_NUM) printf("Num Lock LED is on.\n");
    if (state & LED_CAP) printf("Caps Lock LED is on.\n");
}
```

Sugerencia Haga que su módulo cree un archivo en `/proc` y que las escrituras a ese archivo controlen los LEDs utilizando la IOCTL `KDSETLED` de la consola de Linux.