

PRÁCTICA	CONFIGURAR EL NÚCLEO
OBJETIVOS	<ul style="list-style-type: none"> •Configurar y modificar características del núcleo dinámicamente •Conocer y manejar utilidades de configuración •Estudio de parámetros del núcleo
TRABAJO	<ul style="list-style-type: none"> •Utilizar la utilidad de configuración config •Utilizar la utilidad de configuración menuconfig •Utilizar la utilidad de configuración xconfig •Estudiar en cada una de ellas los parámetros de configuración del sistema. •Estudio y recorrido del pseudo sistema de ficheros /proc/sys. Estudio de los parámetros del sistema que contiene este sistema de ficheros y realizar algunos cambios de parámetros del sistema modificando los ficheros pertinentes.
DOCUMENTACION	Leer help de config, menuconfig y xconfig

Configurar el núcleo

Linux permite configurar y cambiar ciertas características del núcleo dinámicamente (**.config**), esto es, sin recompilar e iniciar el núcleo, mediante:

- utilidades
- llamada al sistema sysctl
- sistema de archivos virtual /proc

Hay que tener privilegios de root.

¡Mas vale prevenir que curar!

- Crear un disco de arranque si no se ha hecho.
- Realizar antes una copia del núcleo.

cp /boot/vmlinuz /boot/vmlinuz.old

Utilidades para configurar el núcleo

- Situarse en el directorio de los fuentes del núcleo
`/usr/src/linux`
- Ejecutar alguna de las tres utilidades para configurar el núcleo
 - **make config**
 - **make menuconfig**
 - **make xconfig**

difieren en la forma de presentación, pero las tres permiten modificar los mismos parámetros modificando el fichero **.config** del núcleo:

- parámetros generales del núcleo
- parámetros relativos a la memoria virtual
- parámetros del sistema de red
- parámetros del sistema de fichero
- otros

make config

- utilidad en modo carácter
 - la mas fea y larga en presentación
 - consume pocos recursos precisa el shell
 - requiere contestar con un carácter
 - y – respuesta afirmativa para incluir esa característica.
 - n – respuesta negativa para no incluir esa característica.
 - m – compilar como un modulo.
- Ctrl+c elimina todos las respuesta y comienza de nuevo.
- ? - para mostrar un menú de ayuda.

make menuconfig

- aplicación basada en un terminal de consola.
- requiere las librerías **curses** (manejo de cursores).
- nos movemos por las opciones mediante los cursores.
- es sencilla de utilizar.
- esta estructurada en bloques.
- tiene una ayuda.

make menuconfig

```
root@dis353sc:/usr/src/linux-2.4
Archive  Editar  Ver  Terminal  Ira  Ayuda
Linux Kernel v2.4.20-8custom Configuration

Main Menu

Arrow keys navigate the menu.  <Enter> selects submenus --->.
Highlighted letters are hotkeys.  Pressing <Y> includes, <N> excludes,
<M> modularizes features.  Press <Esc><Esc> to exit, <?> for Help.
Legend: [*] built-in  [ ] excluded  <M> module  < > module capable

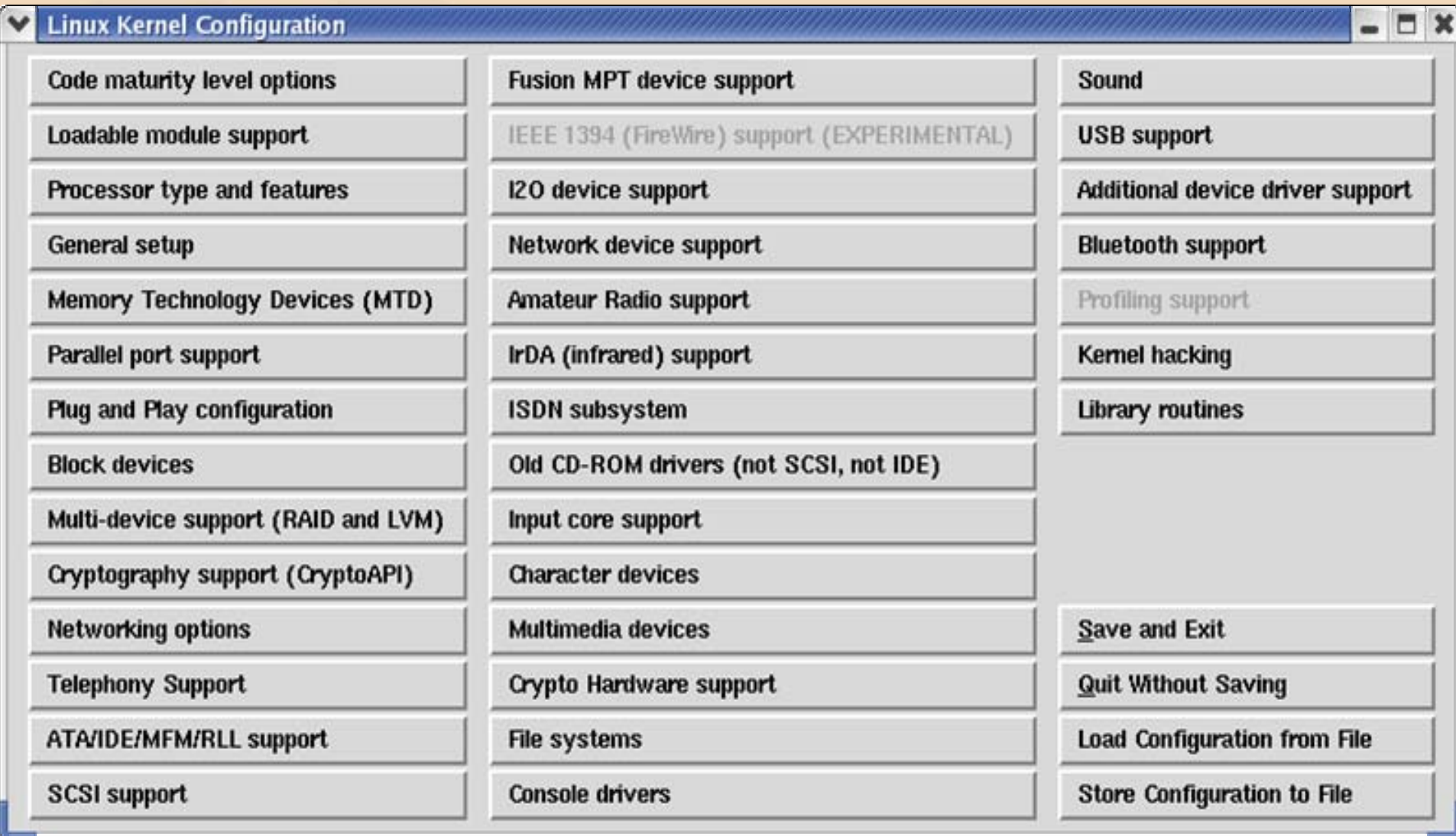
Code maturity level options --->
Loadable module support --->
Processor type and features --->
General setup --->
Memory Technology Devices (MTD) --->
Parallel port support --->
Plug and Play configuration --->
Block devices --->
Multi-device support (RAID and LVM) --->
Cryptography support (CryptoAPI) --->

<Select>  < Exit >  < Help >
```

make xconfig

- requiere un terminal x
- requiere interfase X Windows.
- es el mas agradable por su forma gráfica.
- utiliza un menú con ventanas.
- esta estructurada por bloques.
- tiene una ayuda para cada opción.

make xconfig



make oldconfig

- Actualiza a una nueva configuración partiendo del viejo .config, es rápida, solo hace unas pocas preguntas para los nuevos cambios.

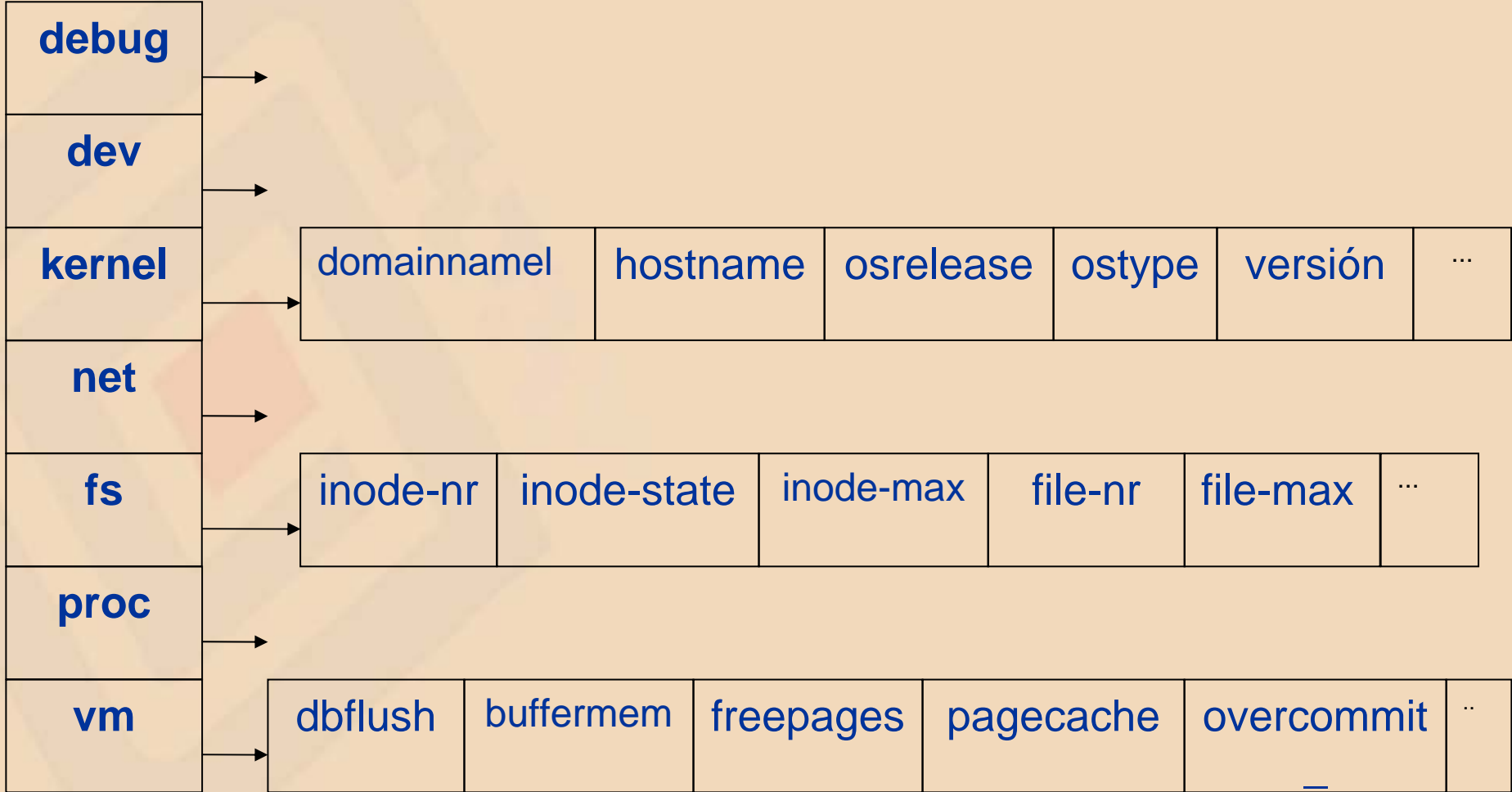
Llamada al sistema sysctl (kernel/sysctl.c)

- Esta llamada al sistema es muy potente y permite configurar mediante programa, dinámicamente ciertos parámetros del sistema.

Sistema de archivos /proc/sys

- Todos los parámetros que son cambiables, son accesibles en forma de un archivo en el directorio /proc/sys.
- ejemplo
/proc/sys/kernel/hostname - nombre de la máquina
- los parámetros son leídos al leer en el correspondiente fichero.
- los parámetros son cambiados al escribir en el correspondiente fichero.
- muchos usuarios pueden leer pero solo el root puede escribir.

/proc/sys



PRÁCTICA	COMPILAR EL NÚCLEO
OBJETIVOS	<ul style="list-style-type: none">• Aprender a compilar el núcleo• Utilizar el comando Make• Comprender los tipos de compilación
TRABAJO	<ul style="list-style-type: none">• Realice una copia del núcleo original antes de modificar y o recompilar• Compile el núcleo como un núcleo monolítico, sin módulos cargables.• Configure el cargador del núcleo LILO para que permita seleccionar un núcleo particular, uno de los cuales será la copia original del núcleo, modificando el fichero /etc/lilo.conf• Generar un disquete de arranque con el nuevo núcleo con make bzdisk. para permitir iniciar el sistema desde disquete.• Traer una nueva versión del núcleo y compilarlo.
DOCUMENTACION	README en el directorio Fuentes de linux

COMPILAR EL NÚCLEO

- La disponibilidad de los fuentes de linux en **`/usr/src/linux...`** permite modificar, compilar y crear un nuevo núcleo de linux.
- Esto hace que Linux sea muy atractivo pues permite:
 - Optimizar el núcleo.
 - Eliminar funciones o manejadores no utilizados.
 - Añadir mejoras al núcleo y nuevas versiones.
 - Añadir nuevas utilidades y funciones.
 - Saber que contiene que hace y como lo hace.

¿Cuándo hay que realizar una actualización del núcleo?

- Añadir una característica de mucha importancia.
- Quitar algún fallo detectado en la versión.
- Pasar a una nueva versión.
- Añadir un nuevo manejador.
- Tapar algún agujero de seguridad.

¡Si no está roto, no lo arregle!

- Antes de instalarlo en un sistema probar en otra máquina que las aplicaciones usadas soportan los cambios.

Directorio /usr/src/linux..

los fuentes de Linux

Está dividido en subdirectorios:

- **documentation** – para almacenar la información generada en la creación y desarrollo de Linux.
- **arch** – Linux soporta varias plataformas intel ix86, motorola 68000, spark64, digital Alpha, ..., es bajo el subdirectorio específico donde se almacenará la compilación.
- **fs** – contiene los fuentes de sistema de ficheros, para soportar el sistema de ficheros virtual de Linus VFS, y subdirectorios con los sistemas de ficheros soportados, minix, ext2, ext3, msdos, nfs,..
- **kernel** – ficheros fuente para la capa más baja del núcleo, planificador de CPU.
- **mm** – fuentes para manejar la memoria principal a bajo nivel.

Directorio /usr/src/linux..

los fuentes de Linux

- **ipc** – fuentes del software para la comunicación entre procesos.
- **net** – fuentes de red y protocolos de red soportados.
- **drivers** – contiene varios subdirectorios para los manejadores de dispositivo.
 - Carácter, bloque, sonido, SCSI,...
- **modulos** – se va cargando de enlaces a las distintas partes del núcleo que se compilan como módulos y donde se ubican.
- **init** – fuentes del programa init. El primero que se ejecuta el iniciarse el núcleo.
- **include** – ficheros cabecera para construir el núcleo. Contiene subdirectorios para las distintas plataformas, red,..
- **lib** – fuentes de las librerías utilizadas por el núcleo.
- **scripts** – scripst utilizados por el núcleo.

Pasos a seguir para construir un nuevo núcleo:

- Hay que tener privilegios de **root**

¡Mas vale prevenir que curar!

- Crear un disco de arranque si no se ha hecho.
- Realizar antes una copia del núcleo.

```
cp /boot/vmlinuz /boot/vmlinuz.old
```

- Situarse en el directorio de los fuentes del núcleo

```
cd /usr/src/linux...
```

Compilar el núcleo

- Existe un fichero **makefile** para el compilador C, que permite generar un núcleo llamado **zImage** de forma automática.

make zImage

- Se puede crear una imagen del núcleo **comprimida**, para evitar mensajes como “núcleo muy grande”.

make bzImage

Borrar ficheros objeto y dependencias

make clean

- Borra los archivos objetos (xxxx.o) creados en la compilación y elimina copias antiguas de la imagen del núcleo.
- Si no se ejecuta se ocupa espacio en disco pero posteriores compilaciones van mas rápidas ya que solo se compilan las partes modificadas del núcleo.

make dep

- Comprueba, actualiza y crea dependencias entre los distintos componentes del núcleo (**.depend**).

make mrproper

- Una puesta a cero, para borrar todo cuando hemos tenido errores en la compilación, borra los objetos creados, las dependencias y el fichero **.config**.

Si utilizamos módulos

make modules

- Si hay funciones del núcleo que se realizan con módulos, hay que compilarlas.

make modules_install

- instala las funciones de los módulos en el directorio **/lib/modules/<versión>**.

Pasos de la compilación

- El orden de los pasos de la compilación son:
make dep
make clean
make bzImage
make modules
make modules_install
- se pueden introducir los comandos en una sola línea y make los compila secuencialmente si no hay errores.

**make dep clean bzImage modules
modules_install**

Último paso: make install

Después de la compilación, el núcleo comprimido **bzImage** se encuentra en el directorio `/usr/src/linux/arch/i386/boot`

Hay que copiarlo a **/boot**

```
cp /usr/src/linux/arch/i386/boot/bzImage /boot/vmlinuz
```

```
cp /usr/src/linux/System.map /boot/System.map
```

System.map contiene símbolos del núcleo para que los módulos puedan interactuar con las funciones del núcleo.

De forma automática

make install

- Copia el nuevo núcleo y sus ficheros asociados a los directorios apropiados.
- Construye una nueva imagen **initrd**.
- Añade nuevas entradas para el fichero de configuración del gestor de arranque.

actualizar lilo

Lanzar el cargador lilo

lilo

Forma automatizada

El comando **make bzlilo** realiza los siguientes pasos de forma automática:

- **cp /boot/vmlinuz /boot/vmlinuz.old**
- **cp /usr/src/linux/arch/i386/boot/bzImage /boot/vmlinuz**
- **lilo**
- **cp /usr/src/linux/System.map /boot/System.map**

System.map contiene símbolos del núcleo para que los módulos puedan interactuar con las funciones del núcleo.

Realizar un **reboot** y ejecutar el nuevo núcleo