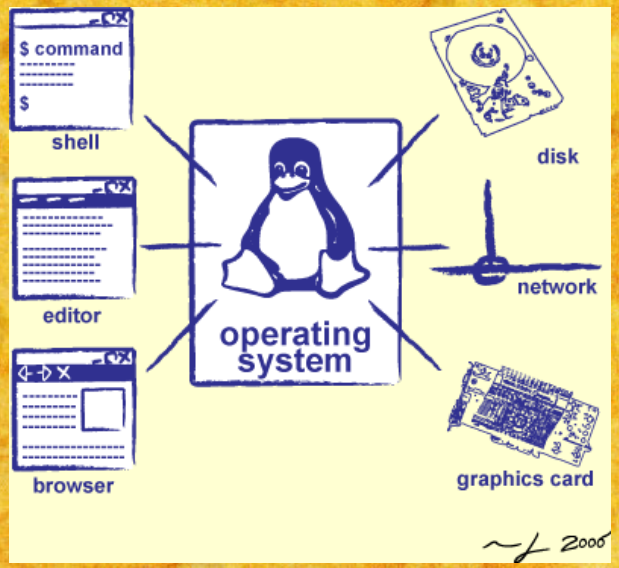
Construyendo un kernel en linux



Por qué recompilar el kernel

- Modificar el modelo predefinido
- Cambiar la frecuencia del timer
- Agregar un parche (patch)
- Agregar System calls personalizadas

Agregar nfsgroups patch

- Este parche permite “eliminar” la restricción del protocolo NFS/RPC la cual permite a cada usuario sólo pertenecer a 16 grupos.
- En Linux el límite es 32

Agregar el patch y construir el nuevo conjunto de paquetes

- Crea el directorio donde se generarán los paquetes del kernel como usuario norma e instalar todos los paquetes necesarios

```
apt-get install fakeroot build-essential devscripts kernel-package
apt-get build-dep linux
```

- Obtener los fuentes del paquete. Es necesario agregar los repositorio *deb-src* en */etc/apt/sources.list*

```
apt-get source linux
cd linux-*
```

- También es posible obtener los fuentes del kernel descargándolos directamente de https://lists.debian.org/debian-kernel/

Agregar el patch y construir el nuevo conjunto de paquetes

- Obteren las fuentes del parche

```
cd linux-*
svn export svn://svn.debian.org/svn/kernel/dists/
trunk/linux/debian
```

Agregar el patch y construir el nuevo conjunto de paquetes

- Aplicar el patch. En el directorio de los fuentes

  zcat patch46.gz | patch -p0

o bien

  patch –p0 < patch46

- Opcional: si no es el primer intente de recompilar ejecutar

  make –f debian/rules clean

Agregar el patch y construir el nuevo conjunto de paquetes

- Verificar que los parches funcionan y resolver conflictos

  make -f debian/rules source-all

en el primer intento el objetivo falla, ejecutar dos veces

- Configurar el kernel

  make menuconfig #es necesario tener instalado ncurses-dev

- Compilar

  fakeroot make-kpkg --initrd --revision=custom.1.0 kernel_image

Agregar el patch y construir el nuevo conjunto de paquetes

- Instalar su nuevo kernel

  dpkg -i ../linux-image-3.16-subarchitecture_custom.1.0_i386.deb

- *dpkg –i* instala algunas paquetes adicionales y configura el bootloader

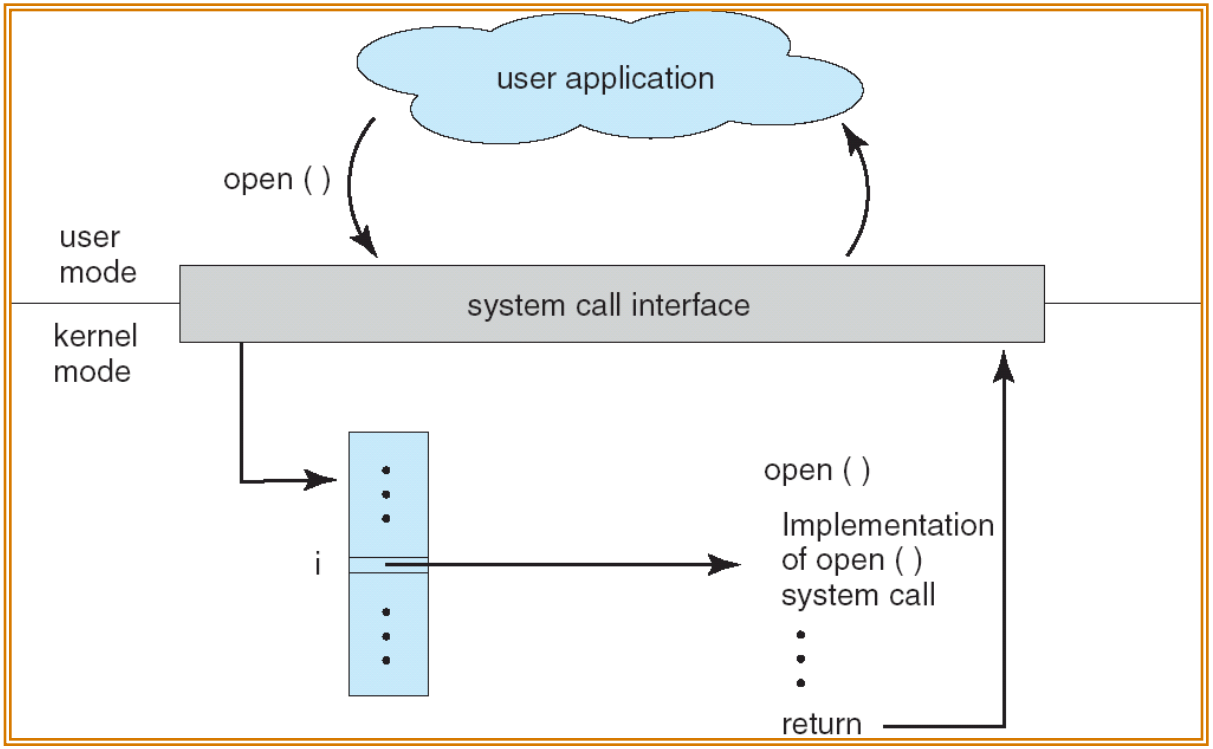
- Si se crearon modulos de paquetes adicionales también es necesario instalarlos

- Reiniciar el sistema

  shutdown –r now

Agregar un sysmtem call al kernel

- Recuerde como opera un system call



Agregar un sysmtem call al kernel

- Los id asociados con cada system call en muchas distribuciones de Linux están listados en (x86 es la arquitectura del ejemplo) :

  Linux-3.x/arch/x86/include/asm/unistd_32l64.h

- Los lista de punteros asociados con cada system call en muchas distribuciones de Linux están en:

  linux-3.x/arch/x86/kernel/syscall_table_32l64.S

Agregar un sysmtem call al kernel

- Ahora podemos agregar un sytem call al kernel. Se utiliza el API de C.

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

asmlinkage int sys_helloworld(){
      printk(KERN_EMERG ”hello world’);
      return 1;
}
```

- Definir una nueva entrada en *unistd.h* para __NR_helloworld e incrementar __NR_syscalls

```
#define __NR_helloworld 312

#define  __NR_syscalls 313
```

Agregar un sysmtem call al kernel

- Incluir en el archivo *syscall_table* la linea

```
.long  sys_helloworld
```

- Definir una nueva entrada en *syscalls.h*

```
asmlinkage long sys_helloworld(void);
```

Agregar un sysmtem call al kernel

- Crear un directorio *hello* en el directorio raíz de los fuentes
- En ese directorio crear el procedimento para le system call.

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

asmlinkage int sys_helloworld(){
    printk(KERN_EMERG ”hello world’);
    return 1;
}
```

- Crear un Makefile en *hello* con el siguiente contenido:

```
obj-y: helloworld.o
```

- Agregar a *core-y* del Makefile principal el directorio *hello*

```
core-y        += kernel/ mm/ fs/ ipc/ security/ crypto/ block/ hello/
```

Utilizar el system call

- Después de reiniciar.

```
#include <linux/errno.h>
#include <sys/syscall.h>
#include<linux/unistd.h>

 #define __NR_helloworld 349

long int helloworld(void){
      return syscall(__NR__helloworld);
}
Int main(){
     int output =helloworld();
     return 0;
}
```

- La ejecución del programa agregará una entrada *"hello world"* en */var/log/messages.log*