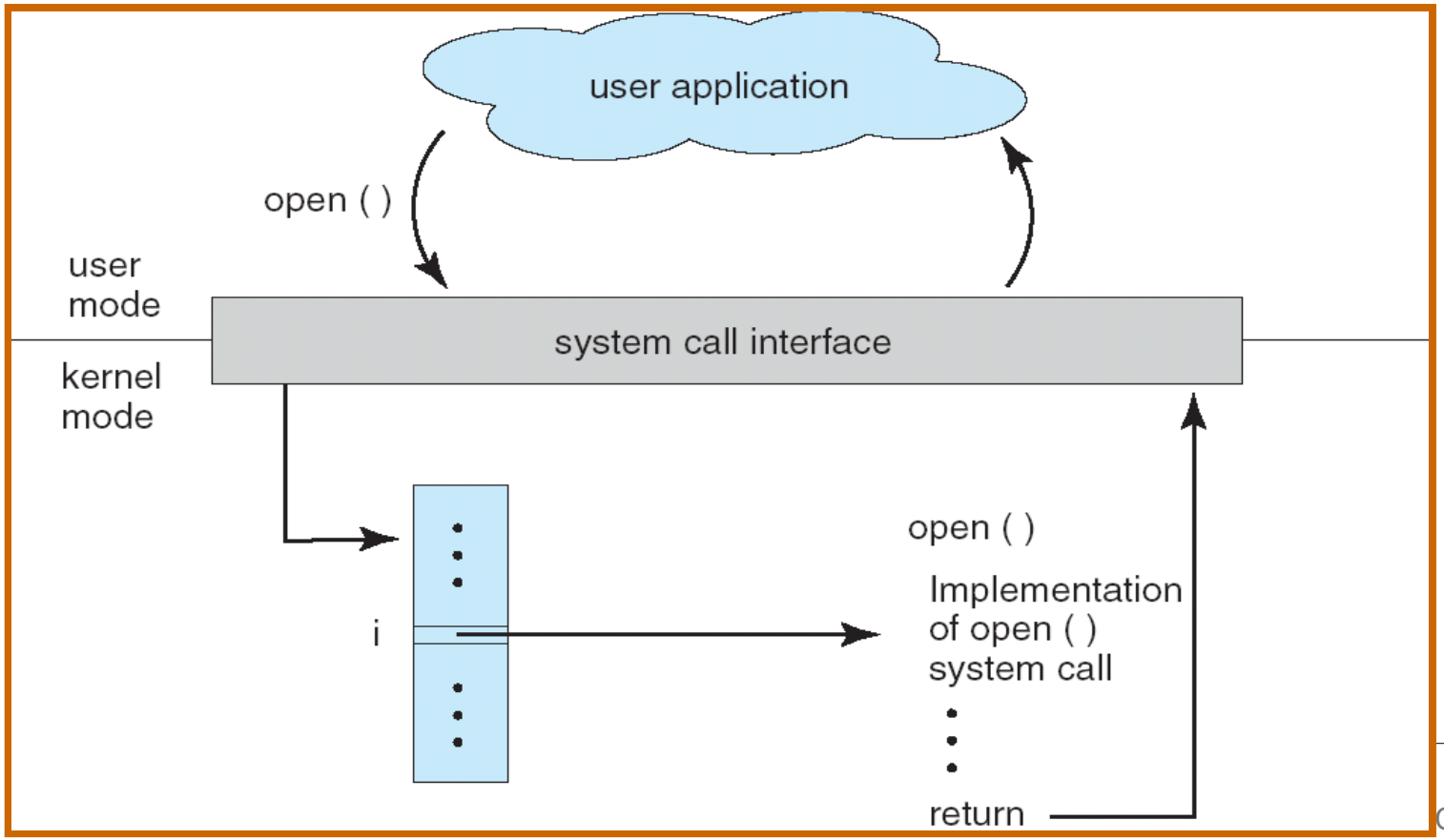
Llamadas Sistema

AISO

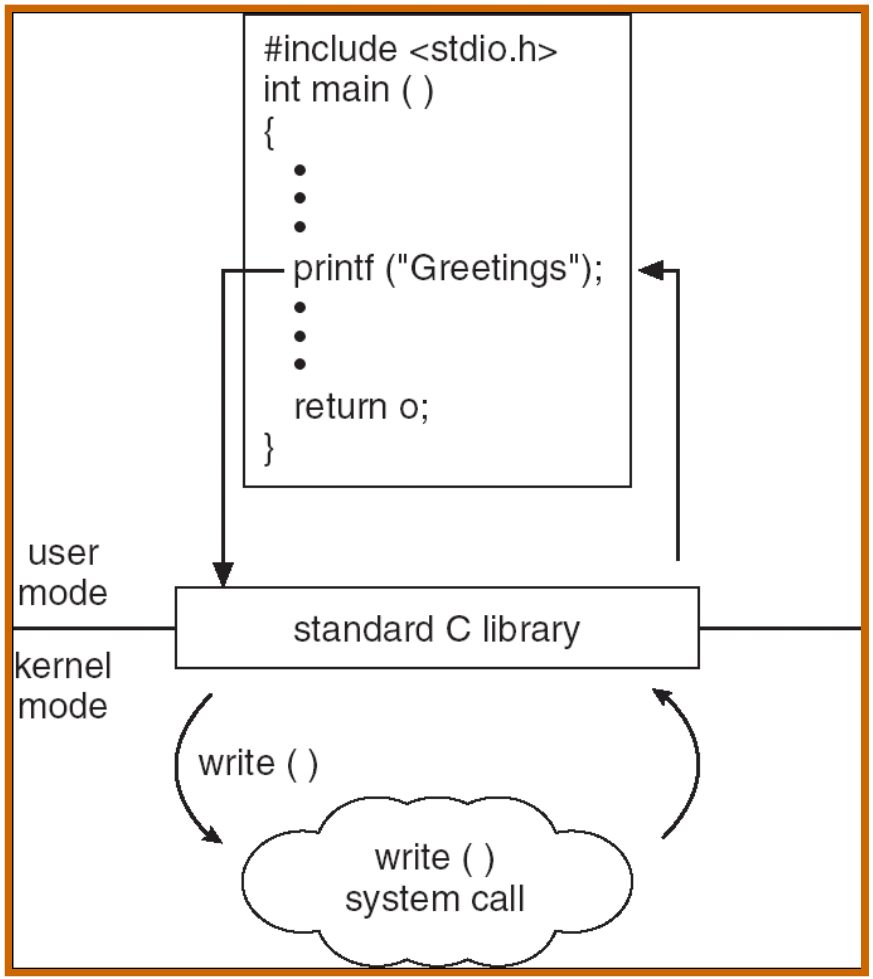
Manuel Prieto-Matías
High Performance Computing Division
ArTeCS Group
Universidad Complutense de Madrid

Llamadas al Sistema (I)

- Normalmente los procesos de usuario no hacen uso explícito de las llamadas al sistema, sino que las invocan via una función de biblioteca (libc)
- El mecanismo para implementar llamadas al sistema es dependiente de la arquitectura
- Cada llamada al sistema se identifica con un numero que puede ser distinto en función de la arquitectura



Llamadas al Sistema (II)



Manuel Prieto-Matías
Universidad Complutense de Madrid

Llamadas al Sistema (III)

- Dos partes:
  - **Transición Usuario-kernel-Usuario** – dependiente de la arquitectura
    - Gestión Paso parametros
    - Transición modo
    - Redirección a la función correspondiente
  - **Handler function** – Implementacion de la llamada (independiente de la arquitectura)
    - **kernel/timer.c**

```
SYSCALL_DEFINE0(getpid)
{
        return task_tgid_vnr(current);
}
```

    - **<include/syscalls.h>**

```
#define SYSCALL_DEFINE0(name)          asmlinkage long sys_##name(void)

asmlinkage long sys_gettid(void);
```

Llamadas al Sistema (IV)

- Paso de Parametros IA-32
  - A través de registros
    - NR SYSCALL **`eax`**
    - Parametros: **`ebx,ecx,edx,esi`** y **`edi`** (6 o + parametros: puntero al espacio de usuario)
      - **`copy_from_user, copy_to_user`**
- Conmutación modo. Varios métodos
  - Tradicional **`int $0x80`** (interrupción software 128) – call gate –
  - Pentium II y Superiores **`sysenter`** y **`sysexit`**
  - Se llaman indirectamente para conservar compatibilidad via "puntero a función" **`call 0xFFFFE000`**

Llamadas al Sistema (V)

- Tabla de llamadas al sistema
  - **`sys_call_table`**
  - Se genera con instrucciones en ensamblador
  - Se ubica en el segmento de datos del kernel
- Para la arquitectura IA-32
  - **`arch/x86/kernel/syscall_table_32.S (entry_32.S)`**
    - **`Ejemplo: .long sys_aiso_call  /* 337 */`**
  - La correspondencia entre llamada y numero de llamada se fija en
  - **`arch/x86/include/asm/unistd_32.h`**
    - **`Ejemplo: #define __NR_aisocall 337`**

Llamadas al Sistema (VI)

- **Valores de Retorno (long)**
  - IA-32: registro **`eax`**
    - **`0 o Positivo: exito`**
    - **`Negativo (-1 .. -511): error`**
  - Definición de errores (positivo)
    - **`<asm-generic/errno-base.h> - errores clasicos -`**
    - **`<asm-generic/errno.h>`**
  - Ej Handler (pasa el error via la pila del kernel)
    - **`return -ENOPERM`**

Llamadas al Sistema (VII)

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

#define __NR_gettid      224

long mygettid(void) {
    return (long) syscall(__NR_gettid);
};

main () {
    printf("El codigo de retorno de la llamada gettid es
%ld\n", mygettid());
}
```

Parche

- Creación parche

```
diff -urN /usr/src/linux-2.6.31 nuevasfuentes > patch1
```

- Aplicar el parche

```
cp -al /usr/src/linux-2.6.31 fuenteslimpias
cd fuenteslimpias
patch -p1 < patch1
```

AISO llamadas al sistema

Versión 0.1

© **Manuel Prieto Matias**

Este documento (o uno muy similar) esta disponible en
https://cv2.sim.ucm.es/moodle/course/view.php?id=3235