



UPV / EHU



eman ta zabal zazu

universidad  
del país vasco

euskal herriko  
unibertsitatea

# Programación Concurrente en Linux

## Llamadas al sistema

OCW  
OpenCourseWare

Alberto Lafuente, Dep. KAT/ATC de la UPV/EHU, bajo Licencia Creative Commons



# Contenido

UPV / EHU

1. Interfaz de llamadas al sistema
2. Ejecución de una llamada al sistema



UPV / EHU

# 1. Interfaz de llamadas al sistema

# Interfaz de llamadas al sistema

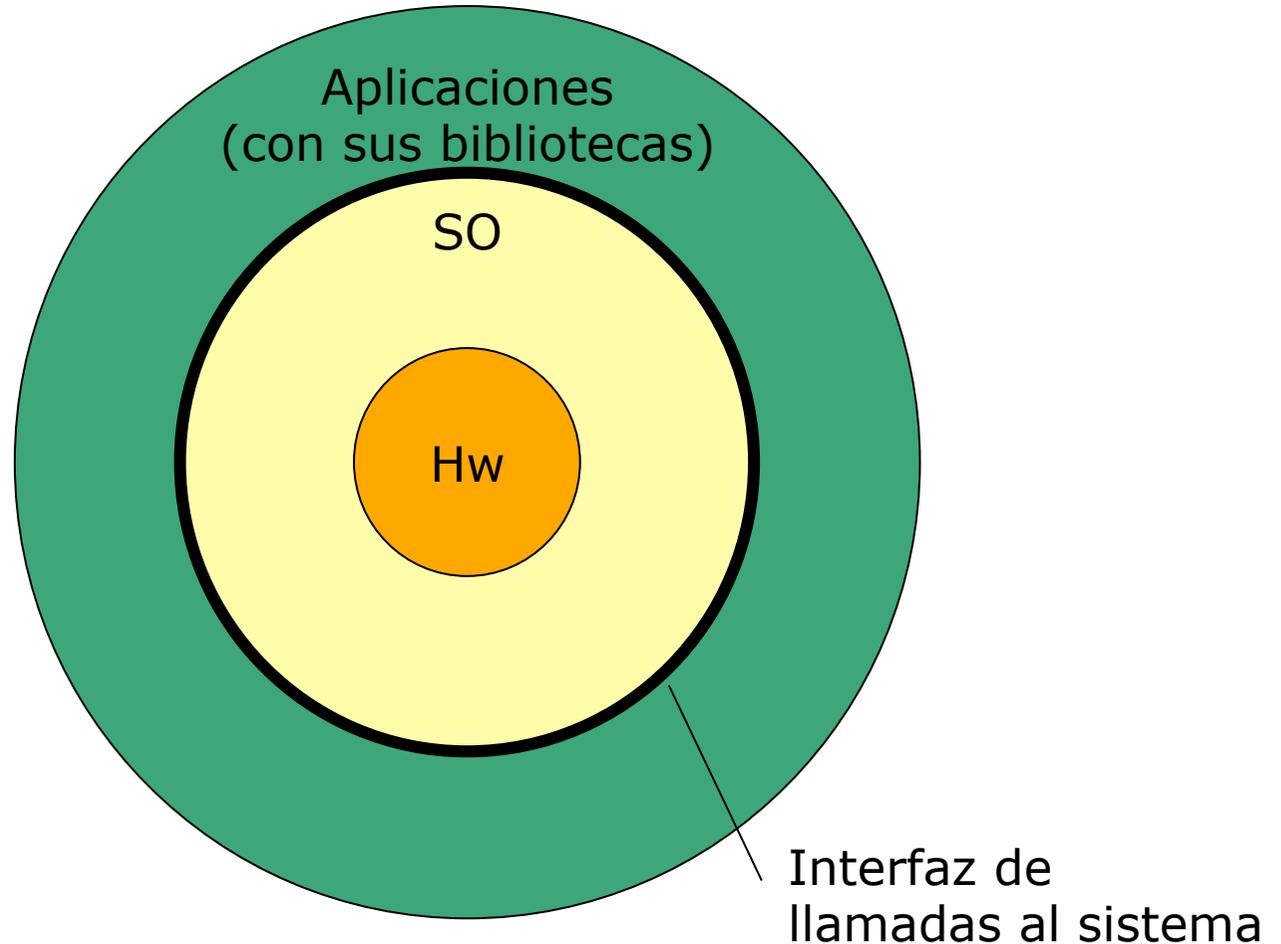
UPV / EHU

- Del sistema operativo se espera que:
  - controle y gestione adecuadamente el acceso a los dispositivos
  - proporcione una interfaz cómoda y unificada para estas operaciones

# Interfaz de llamadas al sistema

- Ejemplos de la interfaz que ofrece Linux a las aplicaciones:
  - *open()*, *close()*, *read()*, *write()*
  - *alarm()*, *pause()*, ...
- Estas funciones son *Llamadas al Sistema (System Calls)*
- Son características y definitorias del sistema operativo, Linux en este caso.
  - Se describen en la Sección 2 del *man*.

# El sistema operativo como interfaz para las aplicaciones



UPV / EHU

# Un programa Linux con llamadas al sistema

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

void fnula() {return;}

unsigned esperar_tiempo (unsigned seg) {
    unsigned s;

    s= alarm(seg);
    pause();
    return(s);
}

int main (int argc, const char * argv[]) {
    int i;
    unsigned t;

    signal(SIGALRM, fnula);

    for (i=1; i<5; i++) {
        printf("espera (%d segundos)\n", i);
        t= esperar_tiempo(i);
        printf("despertado (%d)\n", t);
    }
}
```

# ¿Qué tiene de especial una llamada al sistema?

UPV / EHU

- Siguen siendo funciones de biblioteca, pero...
- A diferencia de las funciones de biblioteca de un lenguaje de programación (p. ej, de C), las llamadas al sistema son específicas del sistema operativo.



UPV / EHU

## 2. Ejecución de una llamada al sistema

# Ejecución de una llamada al sistema

- Una llamada al sistema se utiliza como una función de biblioteca.
- Sin embargo, su ejecución implica la invocación de un servicio del sistema operativo.
  - Ha de ejecutarse en *modo protegido*.
- ¿Cómo proporcionar la ejecución de la llamada en modo protegido de forma segura?

UPV / EHU

# Modos de ejecución

- Las instrucciones máquina pueden ser normales o **reservadas**.
- Las instrucciones reservadas acceden a **espacios protegidos** (de memoria y de E/S).
- Dos modos de ejecución:
  - Modo **normal** o **usuario**
  - Modo **protegido**.
- Las instrucciones privilegiadas sólo pueden ejecutarse en modo protegido.
- Un programa se ejecuta, en principio, en modo normal.
- La instrucción de *cambio de modo* es privilegiada.
- ¿Cómo pasar entonces a modo protegido para ejecutar una llamada al sistema?

UPV / EHU

# Modos de ejecución

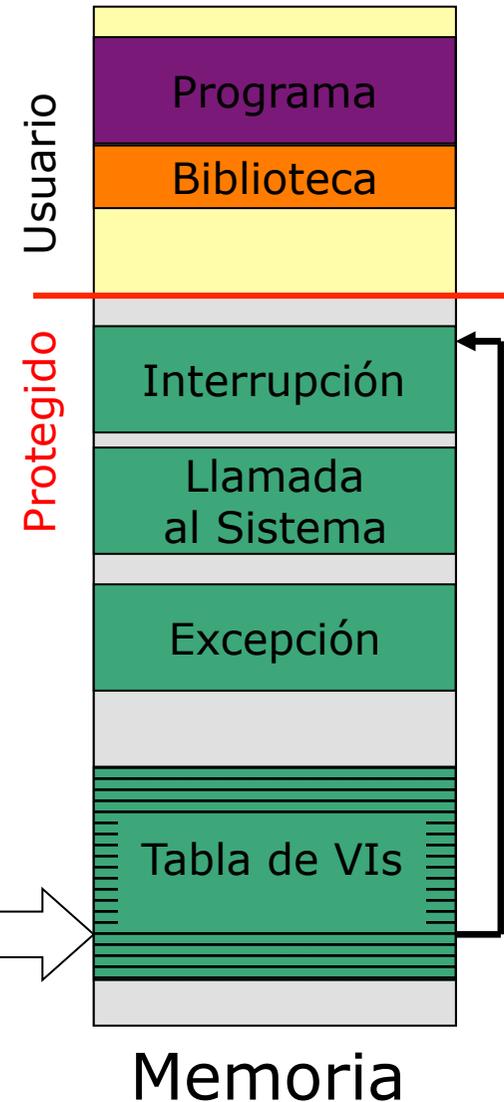
## Cambio de modo

UPV / EHU

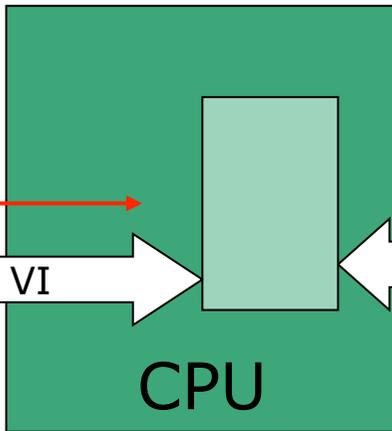
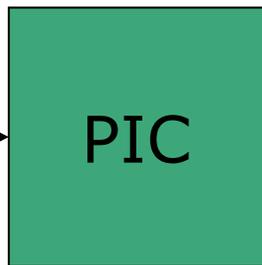
- Se aprovecha el mecanismo de paso de modo normal a modo privilegiado del Vector de Interrupciones.
- Las llamadas al sistema se implementan en espacio protegido mediante rutinas que se direccionan a través del VI.
- El acceso al VI desde memoria de usuario solo es posible a través de un instrucción especial
  - *INT* en arquitectura Intel.
- El retorno de la llamada al sistema restaura el modo de ejecución anterior.

# Acceso a modo protegido mediante VI

Acceso al código de una interrupción mediante VI



Interrupción externa



# Acceso a modo protegido mediante VI

Acceso al código de una llamada al sistema mediante VI

