

Olivia

- So
- controla dispositivos
 - Grupos p. hacer cosas con dispositivos.
 - Interrupciones p. q' hardware avisa a so q' terminó
 - Manejo de errores → trapear error y mostrar algo más.
 - Manejo de unidades → controlar, para q' se tenga el control, evolución sin cambiar mucho S/O

- Problema**
- Muchos disp. distintos →
 - Velocidades \neq
 - \neq formas de hacer S/O → DMA
 - D programando si se termina, etc.

- ✓ **Unidad de Transferencia** → tamaño de 10 q' transfieren
 - ✓ Dispositivos por bloques (discos):
 - Operaciones: Read, Write, Seek
 - ✓ Dispositivos por Caracter (keyboards, mouse, serial ports)
 - Operaciones: get, put
- ✓ **Formas de Acceso**
 - ✓ Secuencial o Aleatorio

- Tipo de acceso**
- **compartido** → acceder varios prog. a la vez (disco)
 - **exclusivo** → acceder de a 1 (impresora)

- ✓ **Tipo de acceso:**
 - Read only: CDROM
 - Write only: Pantalla
 - Read/Write: Disco

Velocidad → se utilizan buffers p. coordinar tiempos de transferencia

se busca trabajar todo lo disp. de manera concurrente

• interfaz común

elaborar detalles del dispositivo con modelo de capas.

Capas

Disp. de S/O son mucho + lentos q' mem y cpu.

Multiprogramación permite q' procs espere su turno mientras otros ejecutan.

Plimitación

Orden reg-entornos y zona de disc. ej.

Orden reg-entornos.

Buffering

Almacén en mem para masas de transferir.

Soluciona problemas de vel. e/dup.

II de tamaño y/o forma de datos.

Caching (Cache)

Mantener en mem copia de los datos q' allí hace falta.

Spooling → administrar flujo de reg-entornos.

→ controlar acceso concurrente al dispositivo.

→ Registro de uso en impresoras.

Registro de dispositivos → Acceso exclusivo.

Manejo de errores → se disminuyen errores → se disminuye el error y se disminuye el error.

→ errores en los de error.

→ logs de errores.

Forma de LRU S/O

→ Bloquear → se en un browser se se un error.

→ No bloquear → se en un browser se se un error.

→ No bloquear → se en un browser se se un error.

Disco en CPU.

1 Software de S/O de LRU de user.

2 Software de S/O de LRU de user.

3 Controlador de dispositivos.

4 Módulo de LRU.

(modo Kernel).

- 2 Controlador de dispositivos
 - 3 Manejador de interrup.
 - 5 Hardware
- } modo Kernel

Capa 1

Librerías de funciones → permiten hacer a sys calls
 → servicios independientes del kernel → interfaz con él.

Proceso de usuario → proceso de impresión spooling

Capa 2

servicio de S/O

interfaz uniforme y modular

Capa 3

kernel maneja S/O de cada disp.

las emuladoras p. eso.

- ✓ Interfaz entre el SO y el HARD
- ✓ Forman parte del espacio de memoria del Kernel
 - ✓ En general se cargan como módulos
- ✓ Los fabricantes de HW implementan el driver en función de una API especificada por el SO
 - ✓ open(), close(), read(), write(), etc
- ✓ Para agregar nuevo HW sólo basta indicar el driver correspondiente sin necesidad de cambios en el Kernel

Drivers en linux

3 tipos de dispositivos → Caracter: S/O

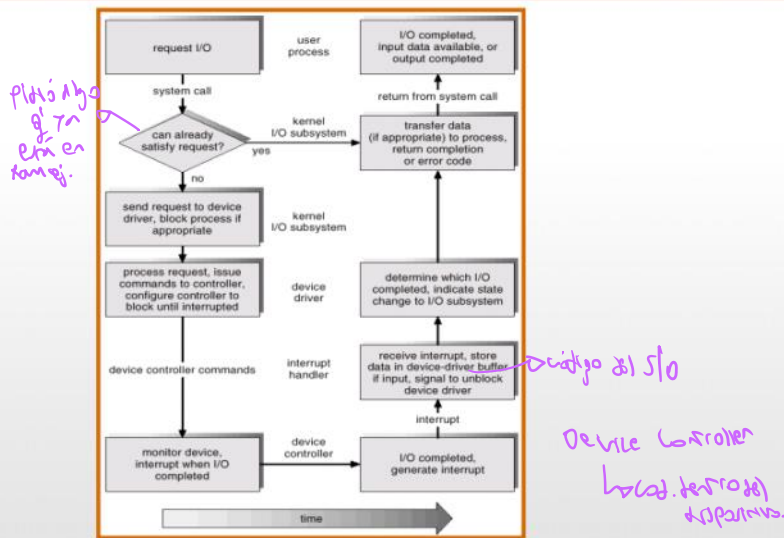
→ Bloque: OMA

→ Red: ports de comunicaciones

Drivers se cargan dinámicamente

Plan de tener init module y cleanup module

Ciclo de vida de un requerimiento de I/O



si la forma mejora performance.

hacer un context switch & interrupciones.

via bus de memoria al copiar datos.

una misma CPU p. ejecutar drivers.

- context switch.

- Caso de copia (si lo necesitan varias veces de copia 1 vez y comparando)

- interrupciones