Instituto Politécnico Nacional

Escuela Superior de Cómputo

Sistemas Operativos

“Tarea 1. Hardware y Software en E/S”

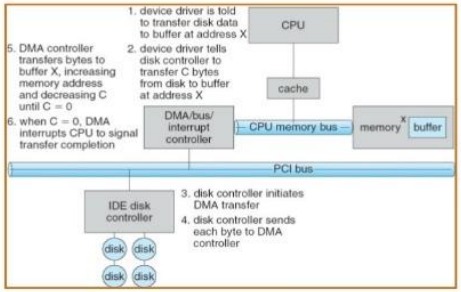
Grupo: 2CM9

Integrantes:

- Martínez Coronel Brayan Yosafat
- Monteros Cervantes Miguel Angel
- Ramírez Olvera Guillermo
- Sánchez Méndez Edmundo Josué

Profesor: Cortés Galicia Jorge

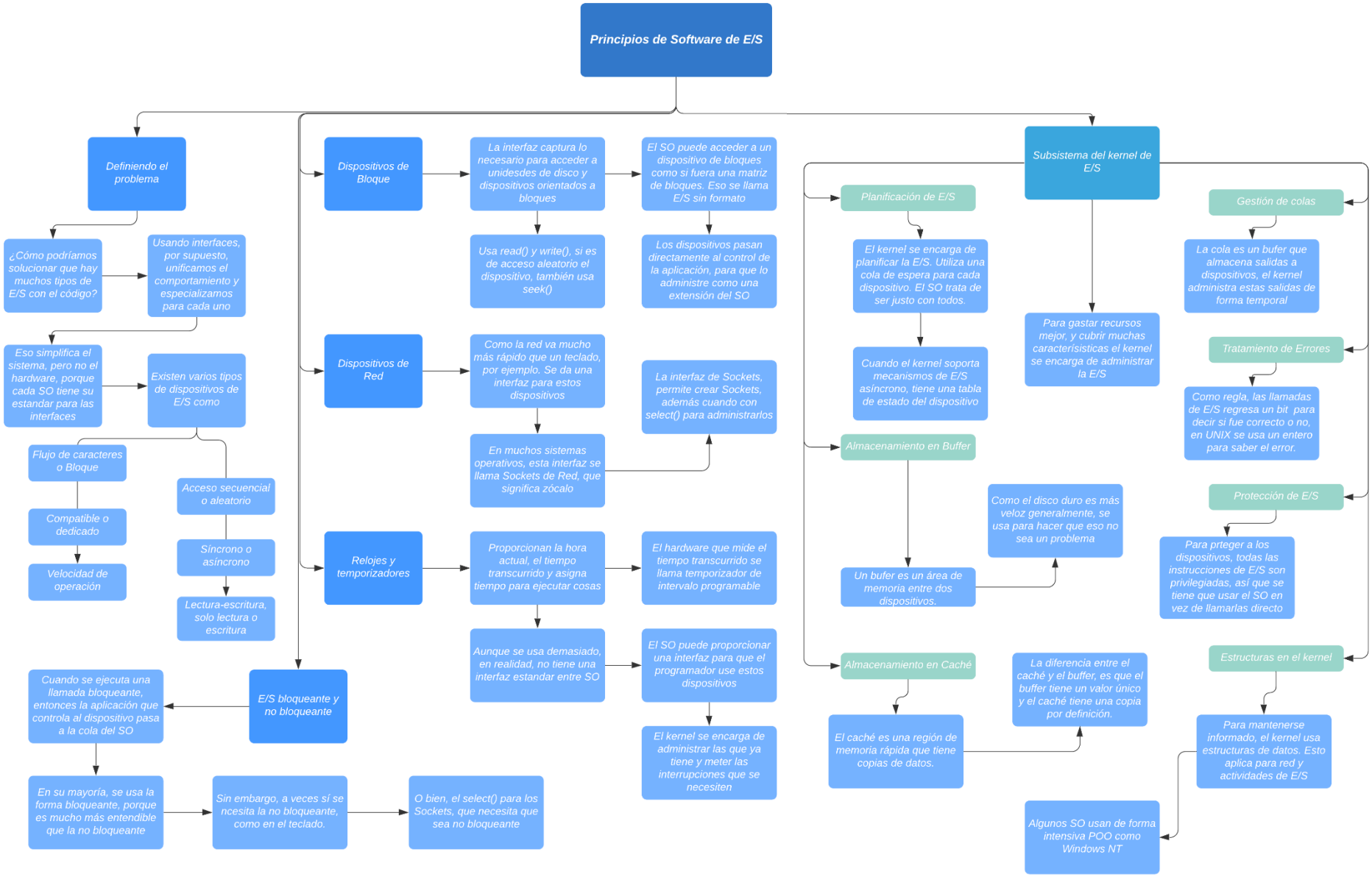
Pasos en una transferencia DMA



- Acceso a memoria
  - E/S programada(PIO)
    - Para evitar sobrecargar la CPU con tareas PIO se utiliza el acceso directo a memoria (DMA), En las computadoras de tipo PC, uno de los componentes estándar es una controladora de DMA simple, y las tarjetas de E/S con control maestro del bus para PC suelen contener su propio hardware de DMA de alta velocidad.
      - DMA-request
      - DMA- acknowledge
    - Se pueden ocupar direcciones de memoria física para las operaciones de DMA, o ocupando un acceso directo a memoria virtual (DVMA) utilizando direcciones virtuales que serán traducidas a direcciones físicas

- **Sistemas de E/S**
  - Controlador
    - Los controladores de dispositivo presentan al subsistema de E/S una interfaz uniforme de acceso a los dispositivos, de forma parecía a como las llamadas al sistema proporcionan una interfaz estándar entre la aplicación y el sistema operativo.
  - Las dos tareas principales de una computadora son las E/S y el procesamiento, en donde la tarea principal es la E/S y el procesamiento es meramente incidental.
  - Debido a que existe una variedad tan amplia de dispositivos de E/S se necesitan diversos métodos para controlar estos dispositivos, estos métodos forman el subsistema de E/S del kernel, que aísla el resto del kernel de la complejidad asociada con la gestión de los dispositivos de E/S.
  - Sondeo
    - 1)El host lee repetidamente el bit de ocupado hasta que dicho bit pasa a cero (Espera activa o sondeo)
    - 2)El host activa el bit de estructura en el registro de comando y escribe un byte en el registro de datos de salida.
    - 3)El host activa en bit de comando preparado
    - 4)Cuando la controladora observa que está activado el bit de comando preparado, activa el bit de ocupada.
    - 5)La controladora lee el registro de comando y ve el comando de escritura. A continuación, lee el registro de salida de datos para obtener el byte y lleva a cabo la E/S hacia el dispositivo.
    - 6)La controladora borra el bit de comando preparado, borra el bit de error en el registro de estado para indicar que la E/S de dispositivo ha tenido éxito y borra el bit de ocupado para indicar que ha finalizado.
  - Controladora
    - Una controladora es una colección de componentes electrónicos que permite controlar un puerto, bus o un dispositivo. Los controladores pueden ser sencillos (controlador de puerto serial) o complejos (controlador de bus SCSI)
    - Una controladora puede soportar una E/S mapeada en memoria, en este caso los registros de control del dispositivo están mapeados en el espacio de direcciones del procesador.
  - Hardware de E/S
    - La mayoría de esos dispositivos pueden clasificarse en una serie de categorías generales: dispositivos de almacenamiento (discos, cintas), dispositivos de transmisión (tarjetas de res, módems), dispositivos de interfaz humana (pantalla, teclado, ratón) o dispositivos más especializados
    - Puerto
      - ·Registro de estado
      - ·Registro de control dispositivo.
      - ·Registro de entrada de datos
      - ·Registro de salida de datos
      - Bus
        - Punto de conexión que permite que cada dispositivo se comunique con la máquina, un ejemplo sería un puerto en serie.
        - Bus de expansión
        - Bus PCI
  - return from interrupt
  - Interrupciones
    - Mecanismo hardware que permite a un dispositivo notificar los eventos a la CPU. El hardware de la CPU tiene un hilo denominado línea de solicitud de interrupción que se comprueba después de ejecutar cada instrucción.
    - Funciones necesarias
      - Se necesita poder diferir el tratamiento de una interrupción durante las secciones de procesamiento crítico.
      - Se necesita interrupciones multinivel, de modo que el sistema operativo pueda distinguir entre interrupciones de alta y baja prioridad, y pueda responder con el grado de urgencia apropiado.
      - Se necesita una forma eficiente de despachar la interrupción a la rutina de tratamiento de interrupciones apropiada para un cierto dispositivo sin necesidad de sondear primero todos los dispositivos para ver vual ha generado la interrupción.
    - Mecanismo de interrupciones
      - Vector de interrupciones y niveles de prioridad de interrupción
      - Líneas de solicitud de interrupción
        - Se necesita una forma eficiente de despachar la interrupción a la rutina de tratamiento de interrupciones apropiada para un cierto dispositivo sin necesidad de sondear primero todos los dispositivos para ver vual ha generado la interrupción.
        - No enmascarable
        - enmascarable

I/O address range (hexadecimal)	device
000–00F	DMA controller
020–021	interrupt controller
040–043	timer
200–20F	game controller
2F8–2FF	serial port (secondary)
320–32F	hard-disk controller
378–37F	parallel port
3D0–3DF	graphics controller
3F0–3F7	diskette-drive controller
3F8–3FF	serial port (primary)


Principios de Software de E/S
Definiendo el problema
¿Cómo podríamos solucionar que hay muchos tipos de E/S con el código?
Usando interfaces, por supuesto, unificamos el comportamiento y especializamos para cada uno
Eso simplifica el sistema, pero no el hardware, porque cada SO tiene su estandar para las interfaces
Existen varios tipos de dispositivos de E/S como
Flujo de caracteres o Bloque
Compatible o dedicado
Velocidad de operación
Acceso secuencial o aleatorio
Síncrono o asíncrono
Lectura-escritura, solo lectura o escritura
Dispositivos de Bloque
La interfaz captura lo necesario para acceder a unidesdes de disco y dispositivos orientados a bloques
Usa read() y write(), si es de acceso aleatorio el dispositivo, también usa seek()
El SO puede acceder a un dispositivo de bloques como si fuera una matriz de bloques. Eso se llama E/S sin formato
Los dispositivos pasan directamente al control de la aplicación, para que lo administre como una extensión del SO
Dispositivos de Red
Como la red va mucho más rápido que un teclado, por ejemplo. Se da una interfaz para estos dispositivos
En muchos sistemas operativos, esta interfaz se llama Sockets de Red, que significa zócalo
La interfaz de Sockets, permite crear Sockets, además cuando con select() para administrarlos
Relojes y temporizadores
Proporcionan la hora actual, el tiempo transcurrido y asigna tiempo para ejecutar cosas
El hardware que mide el tiempo transcurrido se llama temporizador de intervalo programable
Aunque se usa demasiado, en realidad, no tiene una interfaz estandar entre SO
El SO puede proporcionar una interfaz para que el programador use estos dispositivos
El kernel se encarga de administrar las que ya tiene y meter las interrupciones que se necesiten
E/S bloqueante y no bloqueante
Cuando se ejecuta una llamada bloqueante, entonces la aplicación que controla al dispositivo pasa a la cola del SO
En su mayoría, se usa la forma bloqueante, porque es mucho más entendible que la no bloqueante
Sin embargo, a veces sí se ncesita la no bloqueante, como en el teclado.
O bien, el select() para los Sockets, que necesita que sea no bloqueante
Subsistema del kernel de E/S
Planificación de E/S
El kernel se encarga de planificar la E/S. Utiliza una cola de espera para cada dispositivo. El SO trata de ser justo con todos.
Cuando el kernel soporta mecanismos de E/S asíncrono, tiene una tabla de estado del dispositivo
Para gastar recursos mejor, y cubrir muchas caracterísisticas el kernel se encarga de administrar la E/S
Almacenamiento en Buffer
Un bufer es un área de memoria entre dos dispositivos.
Como el disco duro es más veloz generalmente, se usa para hacer que eso no sea un problema
Almacenamiento en Caché
El caché es una región de memoria rápida que tiene copias de datos.
La diferencia entre el caché y el buffer, es que el buffer tiene un valor único y el caché tiene una copia por definición.
Algunos SO usan de forma intensiva POO como Windows NT
Gestión de colas
La cola es un bufer que almacena salidas a dispositivos, el kernel administra estas salidas de forma temporal
Tratamiento de Errores
Como regla, las llamadas de E/S regresa un bit para decir si fue correcto o no, en UNIX se usa un entero para saber el error.
Protección de E/S
Para prteger a los dispositivos, todas las instrucciones de E/S son privilegiadas, así que se tiene que usar el SO en vez de llamarlas directo
Estructuras en el kernel
Para mantenerse informado, el kernel usa estructuras de datos. Esto aplica para red y actividades de E/S