

Instituto Politécnico Nacional Escuela Superior de Cómputo



TAREA 11

Sistemas Operativos

Integrantes:
Mora Ayala José Antonio
Ramírez Cotonieto Luis Fernando
Torres Carrillo Josehf Miguel Ángel
Tovar Jacuinde Rodrigo

Profesor: Cortés Galicia Jorge

PRINCIPIOS DEL HARDWARE DE E/S Hardware de E/S El control de los dispositivos conectados a la computadora es de las principales preocupaciones de los diseñadores de SONDEO sistemas operativos. Debido a la variedad tan amplia de dispositivos E/S. El protocolo completo de interaccion entre el host y una Las computadoras interaccionan con una amplia se necesitan de diversos metodos para controlar estos controladora, puede ser intrincado, pero la nocion basica de negociacion dispositivos. Dichos emtodos forman el sibsistema de variedad de tipos de dispositivos. resulta muy simple La mayoria de esos dispositivospueden clasificarse E/S del kernel, que aisla el kernel de la complejidad en una serie de categorias generales asociada con la gentio de dispositivos Supongamos que se utilizan 2 bits para coordinar la relacion productor consumidor entre la controladora y el host. La **NEGOCIACION** controladora indica su estado mediante el bit de ocupado en el registro de estado. La controladora activa el bit de Dispositivos de almacenamiento dispositivos de interfez humana dispositivos de transmision ocupado, cuando esta ocupado trabajando y borra (discos,cintas) (tarjetas de red, modems) (pantalla, teclado, raton) **INTERRUPCIONES** el bit de ocupado cuando cuando esta en la lista aceptar el siguiente comando. El mecanismo basico de interrupcion funciona de la siguiente forma los dispositivos se comunican con los sistemas informaticos -El host lee repetidamente el bit de ocpado hasta que dicho bit enviando señales atraves de un cable o incluso atraves del aire. pasa a cero -El host activa el bit escritura en el registro de comando y escribe un byte El hardware de CPU tiene un hilo denominado linea de solicitud de interrupcion en el registro de datos de salida que la CPU comprueba despues de ejecutar cada instruccion. Cada dispositivo se comunica con la maquina a traves de un punto -El host activa el bit de comando preparado de conexion(o puerto), como por ejemplo un puerto serie -Cuando la controladora observa que esta activo el bit de comando preparado, activa el bit ocupada Cuando la CPU detecta que una controladora ha activado una señal atraves -La controladora lee el registro de comandos y ve el comando de escritura. A continuacion de la linea de solicitud de interrupcion, la CPU guarda el estado actual si los dispositivos utilizan un conjunto comun de hilos, dicha lee el registro de salida de datos para obtener el byte y lleva acabo la E/S del dispositivo y salta a la rutina de tratamiento de interrupciones situada en una direccion conexion se denomina bus -La controladora borra el bit de comando preparado, borra bit de error en el registro de estado fija de la memoria para indicar que la E/S de dispositivo ha tenido exito y borra el bit de ocupado para indicar que ha finalizado CONTROLADORA **BUSES** En muchas arquitecturas informaticas para sondear un dispositivo basta con tres ciclos de instrucciones de CPU: leer un registro del dispositivo, efectuar una operacion de and logica Un bus es un conjunto de hilos, junto con un protocolo para extraer un bit de estado y saltar si ese bit es distinto de cero. Obviamente la operacion basica de Una controladora es una colección de componentes electronicos rigidamente definido que especifica el conjunto de sondeo resulta eficiente que permiten controlar un puerto, un bus o un dispositivo mensajes que pueden enviarse atraves de esos hilos Los buses se utilizan en multitud de ocasiones EJEMPLO: un ejemplo seria la controladora de un puerto serie, dentro de la arquitectura de los sistemas informaticos En otros casos suele ser mas eficiente que la controladora hardware notifique se trata de un unico chip dentro de la ocntroladora que controla las señales a la CPU cuanto ha pasado el dispositivo a estar listo para el servicio, en lugar de forzar que se trasmiten atraves de los hilos de puertos serie. a la CPU a sondear repetidamente el dispositivo para ver si ha terminado la operacion E/S en terminos electronicos los mensajes se trasmiten mediante patrones de tensiones electricas aplicadas a los hilos, una controladora de bus SCSI no es tan simple; como el protocolo SCSI es muy con unos requisitos de temporizacion bien definidos. complejo, la controladora de bus SCSI se suele implementar mediante una tarjeta de circuito separada que se inserta en la computadora. 1. Necesitamos poder diferir del tratamiento de una interrupcion durante Cuando el dispositivo A tiene un cable que inserta en el dispositivo B, las secciones de procesamiento critico. y el dispositivo B tiene un cable que se inserta en el dispositivo C, Normalmente dicha tarjeta separada contiene un procesador, microcodigo 2. Necesitamos una forma eficiente de despachar la interrupcion a la rutina y el dispositivo C se inserta en un puerto de la computadora, y algo de memoria privada para poder procesar los mensajes del protocolo SCSI de tratamiento de interrupciones apropiada para un cierto dispositivo este tipo de dispositivo se le denomina conexion en cascada sin necesidad de sondear primero todos los dispositivos para ver cual ha generado la interrupcion 3. Necesitamos 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 El mecanismo basico de interrupciones permite a la CPU responder a un suceso asincrono, como es el caso en que una controladora de dispositivo pasa a estar lista para estar servida. Sin embargo en los siguientes sistemas operativos modernos se necesitan funciones mas sofisticadas de tratamiento de interrupciones COMO PUEDE PROPORCIONAR COMANDOS Y DATOS EL PROCESADOR A UNA CONTROLADORA PARA LLEVAR ACABO UNA TRANSFERENCIA E/S? El proposito de un mecanismo de interrupciones vectorizado es reducir la necesidad El hardware informatico moderno, estas tres funcionalidades de que una unica rutina de trtamiento de interrupciones tenga que analizar las proporciona el CPU y el hardware de la controladora de La controladora dispone de uno o mas registros para todas las posibles fuentes de interrupcion para determinar cual es la que interrupciones los datos y las señales de control. necesita servicio El procesador se comunica con la controladora leyendo y escribiendo La mayoria de los procesadores tienen dos lineas de solicitud de interrupcion. patrones de bits en dichos registros. Una forma para llevar acabo esta comunicacion Una de ellas es la interrupcion no enmascarable, que esta reservada para sucesos una forma comun de resolver este problema consiste en utilizar la tecnica es utilizando instrucciones E/S. La instruccion E/S configura las lineas de bus para tales como los errores de memoria no recuperables. La segunda linea de interrupciones del encadenamiento de interrupciones, en la que cada elemento del vector seleccionar el dispositivo apropiado y para leer o escribir bits en un registro del dispositivo es enmascarable; puede ser desactivada por la CPU antes de la ejecucion de secuencia de interrupciones apunta a la cabeza de una lista de rutinas de tratamiento de de instrucciones criticas que no debe ser interrumpida. interrupcion La controladora grafica tiene puertos de E/S para las operaciones basicas de control pero tambien dispone de una gran region de mapeada en memoria para almacenar El mecanismo de interrupcion acepta una direccion, que es el numeor que selecciona el contenido de la pantalla. Los sistemas operativos tienen otros usos adecuados para un mecanismo hardware una rutina especifica de tratamiento de interrupciones de entre un pequeño conjunto y software eficiente que guarde una pequeña cantidad de informacion de estado de rutinas disponibles. En la mayoria de arquitecturas esta direccion es un desplazamiento dentro de una tabla denominada vector de interrupciones del procesador y luego invoque una rutina privilegiada dentro del kernel La controladora grafica tiene puertos de E/S para las operaciones basicas de control pero tambien dispone de una gran region de mapeada en memoria para almacenar el contenido de la pantalla. INTERRUPCION DE SOFTWARE El mecanismo de interrupciones tambien implementa un sistema de niveles de prioridad de interrupcion. Este mecanismo permite a la CPU diferir el tratamiento de las interrupciones de baja prioridad sin enmascarar todas las interrupciones y hace El proceso envia la salida a la pantalla escribiendo datos en la region mapeada Esta instruccion tiene un operando que identifica el servicio del kerneldeseado. posible que una interrupcion de alta prioridad desaloje otra interrupcion de mas baja en memoria. Esta tecnica resulta muy simple de utilizar aparte de que resulta mas Cuando un proceso ejecuta la interrupcion de software, el hardware de facil de escribir millones de bytes en la memoria grafica que ejecutar millones de interrupcion guarda el estado del codigo de usuario, conmuta a modo suoervisor instrucciones de E/S. Pero la facilidad de escritura en una controladora de E/S y realiza el despacho a la rutina del kernel de implementa el servicio solicitado mapeada en memoria se ve compensada por una desventaja: los registros de dispositivos mapeados en memoria son vulnerables a la modificación accidental por parte de los programas. Las interrrupciones tambien pueden usarse para gestionar el flujo de control dentro del kernel La rutina de tratamiento de alta prioridad registra el estado de E/S, borra la interrupcion del dispositivo, comienza la siguiente E/S pendiemte y genera una interrupcion de baja prioridad para completar la tarea -El host lee el registro de entrada de datos para obtener la entrada En resumen las interrupciones de utilizan extensivamente en los SO -El host escribe en el registro de salida de datos para enviar la salida para gestionar sucesos asincronos y para saltar rutinas en modo supervisor La rutina de tratamientpo correspondiente completara la E/S de nivel de usuario dentro del kernel. Para que las tareas mas urgentes se lleven a cabo primero, copiando los datos desde el buffer del kernel al espacio de la aplicacion y luego -El registro de estado contiene bits que el host puede leer. Estos bits indican estados, Las controladoras de dispositivos, los fallos hardware y las llamadas al sistema invocando al planificador para colocar la aplicación en la cola de procesos separados como por ejemplo si se ha completado la ejecucion del comando actual, si hay un byte generan interrupciones para provocar la ejecucion de rutinas del kernel. Dado disponible para sel leido en el registro de la entrada de datos o si se ha producido que las interrupciones se utilizan de forma constante son utiles para una condicion de error en el dispositivo tener un buen rendimiento del sistema -El registro de control puede ser escrito por el host para iniciar un comando o para cambiar el modo de un dispositivo. Por ejemplo, un cierto bit del registro de control de un puerto serie permite seleccionar entre comunicacion full-duplex y semi-duplex, otro bit activa la comprobacion de paridad, un tercer bit configura la longitud de palabra para que sea de 7 u 8 bits y otros bits seleccionan algunas de las velocidades soportadas por el puerto serie. **ACCESO DIRECTO A MEMORIA** En un kernel de modo protegido el sistema opertivo impide generalmente que los Para un dispositivo que realice transferencias de gran tamaño como por ejemplo una unidad de disco parece poco procesos ejecuten directamente comandos de los dispositivos. Esto proteje apropiado utilizar un procesador de proposito general a los datos de frente a las violaciones de los mecanismos de control de acceso para comprobar dicho estado y para escribir datos en un registro y tambien protege al sistema frente al uso erroneo de las controladoras de dispositivo que pudiera provocar fallo catastrofico en el sistema de una controladora de byte a byte lo cual es un proceso que se denomina E/S programada Una vez finalizada la transferencia completa, la controladora DMA Para iniciar una transferencia DMA el host describe un bloque de comando interrumpe a la CPU. Cuando la controladora de DMA toma el control del bus DMA en la memoria. Este bloque contiene un puntero al origen de una transferencia, de memoria, se impide momentaneamente a la CPU acceder a la memoria un puntero al destino de la transferencia y un contador del numero principal, aunque podra seguir accediendo a lso elementos de datos de bytes que hay que transeferir almacenados Para iniciar una transferencia DMA el host describe un bloque de comando DMA en la memoria. Este bloque contiene un puntero al origen de una transferencia, un puntero al destino de la transferencia y un contador del numero La controladora del dispositivo coloca una señal en el hilo DMA-request de bytes que hay que transeferir cada que hay disponible para transferencia una palabra de datos. Esta señal hace que la controladora de DMA tome el control del bus de memoria coloque la direccion deseada en los hilos de direccion de memoria y coloque en el hijo DMA-acknowledge El proceos de negociacion entre la controladora de DMA y la controladora de dispositivo se realiza mediante un par de hilos denominados DMA-request y DMA-acknowledge

Un bus
Una controladora
Un puerto E/S y sus registros
El procedimiento de negociación entre
el host y una controladora de

Contamos como conceptos principales:

dispositivo

La ejecución de este procedimiento mediante un bucle de sondeo o

interrupciones

La descarga de este trabajo en una controladora de DMA para las transferencias de gran envergadura.

Dispositivos

Flujo de caracteres o bloque: Un dispositivo de flujo de caracteres transfiere los bytes uno a uno, mientras que un dispositivo de bloques transfiere un bloque de bytes como una sola unidad.

Acceso secuencial o aleatorio: Un dispositivo secuencial transfiere los datos en un orden fijo determinado por el dispositivo, mientras que el usuario de un dispositivo de acceso aleatorio puede instruir al dispositivo para que se posicione en cualquiera de las uvbivcaciones disponibles de almacenamiento de datos.

Sincrono o asíncrono: Un dispositivo sincrono realiza transferencias de datos con tiempos de respuesta predecibles. Un dispositivo asíncrono exhibe unos tiempos de respuesta irregulares no predecibles.

Compartible o dedicado: Un dispositivo compartible puede ser usado de forma concurrente por varios procesos o hebras, un dispositivo dedicado no puede ser compartido de esta forma.

 Velocidad de operación: LA velocidad de los dispositivos van desde unos pocos bytes por segundo a unos cuantos gigabytes por segundo.

Lectura-escritura, solo lectura o solo escritura: Algunos dispositivos realizan tanto entrada como salida, pero otros solo soportan una única dirección de transferencia de los datos.

Interfaz de E/S de las aplicaciones

La técnica utilizada aquí se basa en los conceptos de abstracción, encapsulation y descomposición del software en niveles. Podemos abstraer a diferencias del detalle existente entre los dispositivos de ES identificando unos cuantos tipos generales del dispositivo.

Para acceder a cada tipo general se utiliza un conjunto estandarizado de funciones, es decir, una interfaz. Las diferencias se encapsulan en módulos del kernel denominados controladores del dispositivo que están personalizados internamente para cada dispositivo pero exportan una de las interfaces estándar.

El propósito de la capa de controladores de dispositivo es ocultar a ojos del subsistema ES del kernel las diferencias existentes entre las controladora del dispositivo.

Hacer el su sistema ES independiente del hardware simplifica el trabajo del desarrollador del sistema operativo y también beneficia a los fabricantes de software.

En lo que respecta al acceso por parte de las aplicaciones, muchas de estas diferencias quedan ocultas gracias al sistema operativo, y los dispositivos se agrupan en unos cuantos tipos convencionales.

Los principales sistemas de acceso incluyen ES de bloque, la ES de flujo de caracteres, el acceso a archivos mareados en memoria y los sockets de red. Los sistemas operativos también proporcionan llamadas especiales al sistema para acceder a unos cuantos dispositivos adicionales, como por ejemplo un reloj o un contador.

Algunos sistemas operativos proporcionan un conjunto de llamadas al sistema para dispositivos de visualización grafica, de video y de audio.

Dispositivos de bloques y de caracteres

La interfaz de dispositivo de bloques captura todos los aspectos necesarios para acceder a unidades de disco y a otros dispositivos orientados a bloques. El dispositivo debe de comprender comandos tales como read() o write(), para especificar que bloque hay que transferir a continuación.

Acceder a un dispositivo de bloques como si fuera un simple matriz lineal de bloques se denomina en ocasiones ES sin formato o en bruto. Si la aplicación realiza su propio almacenamiento en. buffer creara almacenamiento adicional e inecesario.

Una solución de compromiso consiste en que el sistema operativo incluya un modo de operación con los archivos en el que se desactiven el almacenamiento en buffer y los mecanismos de bloqueo. En el mundo UNIX este modo se denomina ES directa.

Los mecanismos de acceso a archivos mareados en memoria pueden implementarse por encima de los controladores de dispositivos de bloques. En lugar de ofrecer operaciones de lectura y escritura, una interfaz de Mateo de memoria proporciona acceso al almacenamiento en disco mediante una matriz de bits de la memoria principal.

Las transferencias se gestionan en el mismo mecanismo que se emplea para el acceso a memoria virtual con paginación bajo demanda, ES mapeada en memoria.



Relojes y temporizadores

Principios del software E/S

Dispositivos de

red

La mayoría de los sistemas operativos

proporcionan una interfaz de E/S de red que es

diferente de la interfaz utilizada para los discos,

La utilización de select () elimina los mecanismos

de sondeo y de espera activa que serian

necesarios para la ES de red en caso de que esta

llamada no existiera. Estas funciones encapsulan

el comportamiento esencial de las redes,

facilitando la creación de hardware de redes.

llamada la interfaz de socket de red.

Disponen de funciones básicas:

Proporcional la hora actual

Proporcionar el tiempo transcurrido

Configurar un temporizador para que provoque la ejecución de la operación

X en el instante T.

El. hardware necesario para medir el tiempo transcurrido y para provocar la ejecución de operaciones se denomina temporizador de intervalo programable.

El subsistema de ES de disco utiliza este mecanismo para invocar la orden de volcar en el disco periódicamente los bufetes cache sucios y el sistema de red lo utiliza para cancelar aquellas operaciones que estén progresando de forma demasiado lenta debido a fallos de red o la congestion.

En muchas computadoras, la frecuencia de interrupciones generada por el reloj de hardware esta comprendida entre 18 y 60 tics por segundo.

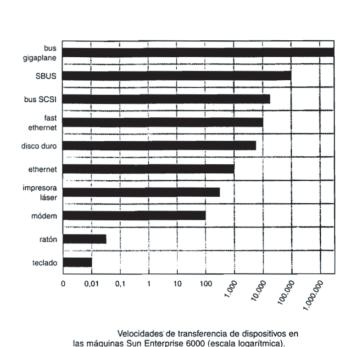
ES bloque ante y no bloqueante

Cuando una aplicación ejecuta una llamada al sistema bloque ante, se suspende la ejecución de la aplicación. En ese instante, ka aplicación se pasa de la cola de ejecución del sistema operativo a una cola de espera.

Las acciones físicas realizadas por los dispositivos ES son generalmente asíncronas, consumiendo una cantidad variable o impredecible de tiempo.

Algunos procesos de nivel de usuario necesitan una ES no bloqueante.

La diferencia entre las llamadas al sistema no bloqueantes y asíncronas es que una operación read () no bloqueante vuelve inmediatamente con los datos que haya disponibles, mientras que una llamada read () asíncronas solicita una transferencia que se realizara de modo completo pero que se completara en algún instante del futuro.



Subsistema de ES del kernel

Un kernel proporciona muchos servicios relacionados con la ES. Varios de los servicios de planificación como, almacenamiento en buffer, almacenamiento en cache, gestión de colas, reserva de dispositivos, y tratamiento de errores.El su sistema ES también es responsable de protegerse a si mismo de los procesos erróneos y de los usuarios maliciosos.

Planificación de ES

Planificación de solicitudes de ES significa determinar un orden de planificación a ejecutarlas. Mejora el rendimiento global del sistema, permite partir el acceso a los dispositivos equivalentes enttre los distintos procesos y puede reducir el tiempo de aspectos.

Los desaerrolladores de sistema operativo implementan los mecanismos de planificacion manteniendo una cola de espera de solicitudes para cada dispositivo .

Cuando un kernel soporta mecanismos de ES asíncrona, debe de ser capaz de controlar multiples solicitudes de ES al mismo tiempo. Con este fin, el sistema operativo puede asociar la cola de espera con una tabla de estado del dispositivo.

Almacenamiento de buffer

Un buffer es un area de memoria que queda almacenada datos mientras se están transfiriendo entre dos dispositivos o entre un dispositivo y una aplicación. Se realiza para una adaptación de velocidades entre el productor y el consumidor, realiza la adaptación entre dispositivos que tengan diferentes tamaños de transferencias de datos y soporta la semántica de copia en la ES de aplicaciones.



Un cache es una region de memoria rápida que alberga copias de ciertos datos. El acceso a la copia almacenada en cache es mas eficiente que el acceso al original. Cujando el kernel recibe una solicitud de ES de archivo, accede primero a la cache de buffer para ver si dicha region de archivo ya esta disponible en la memoria principal.



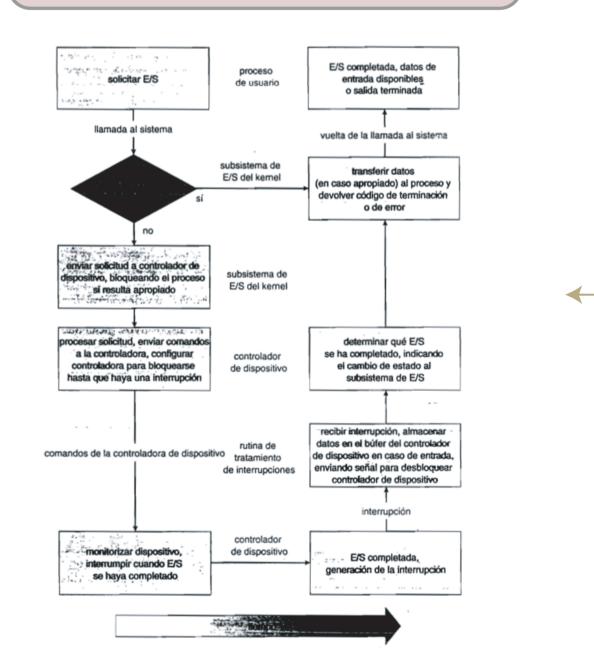
Una cola de dispositivo es un buffer que almacena la salida dirigida a un dispositivo, el sistema operativo proporciona una interfaz de control que permite a los usuarios y administradores del sistema visualizar la cala de solicitudes de impresión.

Resumen del su sistema de ES del kernel

- Gestión del espacio de nombres para archivos y dispositivos.
- Control de acceso a los archivos y dispositivos.
- Control de operaciones (por ejemplo, un módem no puede ejecutar una operación seek ()).
- Asignación de espacio en el sistema de archivos.
- Asignación de dispositivos.
- Almacenamiento en búfer.
- Almacenamiento en caché y gestión de colas de impresión.
- Planificación de E/S.
- Monitorización del estado de los dispositivos, tratamiento de errores y recuperación de fallos.
- Configuración e inicialización de controladores de dispositivos.

Los niveles superiores del subsistema de E/S acceden a los dispositivos a través de la interfaz uniforme proporcionada por los controladores de dispositivo.

Transformación de solicitudes



Protección de ES

ratamiento de

errores

Un sistema operativo que utilice protegida puede

defenderse frente a muchos tipos de errores de

hardware. Como regla general, una llamada de ES

al sistema devolverá un bit de información acerca

del estado de la llamada, mediante el que se indica

si este ha tenido éxito o no. Si falla se creara un

código de tipo adicional que indica la categoría

del fallo.

Para evitar que los usuarios realicen operaciones ES ilegales, definimos todas las instrucciones ES d como instrucciones privilegiadas el programa de usuario solo ejecuta estas.

Estructura de datos del kernel

