Entrada/salida

Todas las operaciones que involucran dispositivos de entrada/salida pasan forzosamente por el SSOO ya que el código necesario se ejecuta en modo privilegiado. Abstracción device es una ampliación de la abstracción fichero.La capa más próxima a los dispositivos son los drivers o controladores que son el código.

Modelo Entrada/Salida:Los dispositivos se comunican con la cpu usando protocolos (envio de mensajesy señales) y si la conexión ess compartida se denomina bus. La cpu se comunica con muchos buses diferentes y utiliza los bridges como circuito de arbitraje. Todo se comunica con un controlador(USB, GPU, SSD) que tiene ciertos registros para dar info del dispositivo.

**Direccionamiento:**Dos tipos—Puertos de memoria mapeados(port-mapped IO)Mapa de memoria muy reducido y se necesitaban instrucciones específicas para cada dispositivo—memory-mapped IO.Rango fijo de direcciones de memoria en las que mapear registros, mas simple para el programado. X86 mantiene el modo de puertos del 8086 pero funciona el nuevo también. Dos problemas: la caché, en cada pagina el bit de no quedarse en caché estará levantado y la adaptación a la velocidad entre la CPU y el dispositivo. Para este ultimo problema la primera solución fue la espera activa.Luego se inventó el DMA (direct memory access) que evitava que los batch de datos pasaran por la CPU y dejaba que fluyeran entre dispositivos.Esto era posible con el controlador del DMA(HW) y que los dispositivos aceptaran este método. Para no colasionar la CPU hay dos mnaneras: transferencia a ráfagas o un robo de ciclo. --------- En DMA hay que manejar direcciones físicas, no virtuales, ya que los dispositivos de E/S no entienden de memoria virtual. Se pueden transferir varios bloques en una misma operación, pero solo si las direcciones físicas son contiguas.

En unix un device puede ser de tres clases: Otrientado a bloques transfiere info en bloques de tam fijo y permiten acceso aleatorio. Orientados a carácter no son de acceso aleatorio si no que funcionan con buffers(colas). Orientados a red manejan paquetes de info fijos pero el acceso no es aleatorio.

**Terminal:**Dispositivo de interacción con el usuario por excelencia. Consta de teclado y monitor, en equipos antiguos cada tecla generaba una interrupción y la CPU recibía un código de tecla, la llegada era secuancial y se llamaba raw mode. Ahora con un uso de buffer y microcontrolador solo se envían las cosas cuando se aprieta enter, modo elaborado.

Imagen de la pantalla de un celular con texto

Descripción generada automáticamente con confianza bajaAlmacenamiento secundario es el que persiste tras un fallo de alimentación. Los discos de estado solido utilizan NAND Flash. SSD con interfaz SATA antiguo ahora con m.2 mas rápido. Un ssd es un microordenador.

Tienen sistemas de ficheros, para no nestresar zonas y que se echen a perder antes se utiliza el Wear Leveling.

Los sectores que es donde se guarda la información en vez de borrarse cuando se borra el archivo solo se marcan como libres para que se puedan sobrescribir.

**USB**: Universal Serial Bus

2.0 utilizaba dos hilos para alimentacion y tenia modo de transmisión semiduplex, se puede enviar en ambas direcciones pero no a la vez. Tiene un arbol con hasta 127 nodos hub(padre) slave(hijo). Su alimentacion es asimetrica, solo la proporciona el extremo hub.El host siemprees el @1 el @0 está reservado.

Una captura de pantalla de un celular con texto

Descripción generada automáticamente con confianza mediaUSB 4 tipos de transferencia: Control(config), interrupción(raton teclado), masiva(ficheros) e isócrona(video/audio).La comunicación entre host y dispositivo se hace mediante tokens, solo el propietario del token puede escribvir en el bus, solo se leen las tramas dirigidas a su end-point.

Handshake

**3.0:**8hilos, comunicación full duplex, alimentación desde cualquiera de los extremos.

**Drivers en Linux** Hay dos formas de instanciar drivers, compilandolos durante la generaciond el kernel o con el sistema plug and play. La primera no se usa mas que en estructuras muy grandes y que no quieren viruses por USB.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteLos dispositivos en Linux se identifican por las cifras major y minor. Major dice que tipo de dispositivo es y que driver lo maneja y minor distingue los dispositivos de la misma clase.

Los drivers no entienden el formato de la información, solo ven flujos de bytes. Po eso se usa open(), que devuelve el descriptor que usara pàra las operaciones mientras permanezca abierto y se cierra con close().