



Preguntas guías

1. ¿Cómo se pueden calificar las memorias?

Los tipos que encuentro son muy variados en cuanto a la capacidad, utilidad y precio, todas son necesarias para el funcionamiento de la computadora ya que cumplen con distintos roles.

Dinámica

Por localización:

Microprocesador (CPU, registros), Interna, Externa

Unidades de transferencia: Interna (usualmente gobernada por el tamaño del bus de datos)

Externa (usualmente por bloques mucho mayores que un carácter)

Unidad de dirección (menor locación que puede ser unívocamente asignada, o sea la dirección más pequeña que puede ser asignada)

Capacidad: Tamaño de palabras, Cantidad de bytes

Formas de escritura: Big-Endian (la parte más significativa del dato se encuentra en la dirección más baja de la memoria. IBM Mainframe, Motorola, RISC), Little-Endian (la parte más significativa del dato se encuentra en la dirección más alta de la memoria. Intel)

EL ENCARGADO DE MANIPULAR EL DIRECCIONAMIENTO Y LAS COPIAS ENTRE MEMORIAS ES EL MICROPROCESADOR O EL SISTEMA OPERATIVO.

2. ¿A qué se denomina memoria virtual dentro de una computadora? ¿Qué ventaja aporta su empleo y que desventaja puede provocar?

La memoria virtual es cuando usamos los discos de almacenamiento masivo (discos rígidos o ssd) como una extensión de la memoria de trabajo. Su ventaja es clara, ya que es más barato para su coste de almacenamiento por bit, pero su desventaja es muy importante: al ser más lenta (por requerir del uso de un dispositivo lento), degrada la velocidad del sistema, ya que debe esperar a este dispositivo de almacenamiento masivo (lento).

3. ¿Qué métodos de accesos pueden ser usados en las memorias?

Acceso secuencial: Tiene que recorrer obligatoriamente la parte anterior de la memoria hasta llegar a la parte que buscamos. Ejemplo: Cinta de caset.

Acceso directo: Los bloques de datos tienen una dirección única basada en su estructura física. La velocidad de acceso a un bloque de datos puede variar de acuerdo a su posición. Puede acceder a cualquier posición sin leer la anterior, sin embargo, el desplazamiento varía según la posición en memoria. Ejemplo: Disco rígido

Acceso aleatorio: Puede acceder a cualquier bloque de datos que quiera, todos a la misma velocidad. Ejemplo: Memoria RAM.



Acceso asociativo: Se trata de un tipo de memoria de acceso aleatorio donde el acceso se hace basándose en el contenido y no en la dirección. Ejemplo: Caché

4. **¿Cuál es el objetivo de la jerarquía de memorias?**

La jerarquía de memoria es una organización piramidal de la memoria que tienen las computadoras. Su objetivo es el mejorar la velocidad y obtener el rendimiento de una memoria de alto costo a costos menores mediante el principio de cercanía de referencias. Los datos siempre se copiarán entre niveles adyacentes por vez. En niveles de velocidad y costo, de más rápido y caro a más lento y barato, tendremos primero a los registros del procesador, luego la memoria principal donde se encontrarán uno o más niveles de caché y la RAM, y de manera externa tendremos a los almacenamientos masivos (en el más lento de todos tenemos copias de memoria, si se los quiere tomar para esta pirámide de memoria). También, el que las memorias se encuentren adyacentes a memorias de velocidad distinta, nos permite que los dispositivos más rápidos no esperen a los más lentos de todos, sino que espere a los de velocidad similar, de forma tal que por ejemplo el procesador no deba esperar al disco duro y espere en su lugar al caché.

5. **¿Qué diferencia existe entre tiempo de acceso y tiempo del ciclo de memoria?**

El tiempo de acceso es el que se tarda en obtener el dato una vez que tenemos la dirección (lo que tarda en buscar algo con la dirección que le damos).

El tiempo de ciclo podríamos verlo como la frecuencia en la que podemos leer o escribir entre vez y vez. Es, en otras palabras, el tiempo transcurrido entre que se solicita un dato a la memoria hasta que esta se encuentra en disposición de realizar una nueva operación Read/Write. Es igual al tiempo de acceso + el tiempo de recuperación.

La tasa de transferencia es la velocidad a la que se pueden mover los datos.

6. **¿Qué entiende por principio de localidad espacial? De un ejemplo.**

PROBABILIDAD DE REPETIR INSTRUCCIONES CERCANAS ENTRE SÍ. (CERCANAS EN LAS DIRECCIONES DE MEMORIA)

La localidad espacial implica que los programas consumen mucho tiempo dando vueltas sobre el mismo punto en la memoria (se repite mucho la misma parte del programa).

LOS ACCESOS A MEMORIA SON MUCHO MÁS LENTOS QUE LA VELOCIDAD DE LAS INSTRUCCIONES, LO QUE GENERA CUELLOS DE BOTELLA. Si aprovechamos los principios de localidad espacial y temporal, el guardar los datos a usar en el caché nos permitirán ahorrarnos estos cuellos de botella.

7. **¿Qué entiende por principio de localidad temporal? De un ejemplo.**

PROBABILIDAD DE REPETIR LAS MISMAS INSTRUCCIONES.

Cuando un programa hace referencia a un espacio de memoria, normalmente accede a ella en el corto plazo, y en un programa la mayor parte de referencias a memoria se hacen respecto a una pequeña cantidad de direcciones. Si en un momento una posición de memoria particular es referenciada, entonces es muy probable que la misma ubicación



vuelva a ser referenciada en un futuro cercano. Existe proximidad temporal entre las referencias adyacentes a la misma posición de memoria

La mayor parte del tiempo la voy a ocupar en una pequeña parte del código (en un entorno de direcciones de memoria cercanas entre sí), normalmente de forma repetitiva en un plazo de tiempo corto; la mayor parte del código no será usada de forma tan repetitiva como la parte más pequeña anteriormente mencionada.

Implica que una pequeña parte del programa es la que más se repite y utiliza continuamente y con poco tiempo entre sí, manteniendo su acceso a la memoria. Esto quiere decir que una parte del programa va a ser la más utilizada y la que más referencias a memoria tome, y que cuando se haga referencia a un espacio de memoria es más probable que se acceda al mismo en el corto plazo.

LOS ACCESOS A MEMORIA SON MUCHO MÁS LENTOS QUE LA VELOCIDAD DE LAS INSTRUCCIONES, LO QUE GENERA CUELLOS DE BOTELLA. Si aprovechamos los principios de localidad espacial y temporal, el guardar los datos a usar en el caché nos permitirán ahorrarnos estos cuellos de botella.

8. **¿Cómo justifico que la transferencia más eficiente de la memoria sea el bloque y no el byte o bit?**

Nos evita tener que buscar byte por byte /bit por bit en la memoria. Esto se hace llevando un único bloque, así que, en lugar de buscar la dirección de cada byte, solo con la dirección de comienzo y final del bloque de datos, podemos transferir todo el conjunto y no buscar uno por uno. Además, por el principio de localidad espacial y temporal se llega a la conclusión de que es mejor transferir de a bloques y no en bytes, ya que como existe una alta probabilidad de que las instrucciones cercanas a la instrucción que estoy ejecutando o esa misma instrucción vuelvan a ser ejecutadas es preferible traer de a bloques y no de a una sola instrucción.

9. **¿Qué es una memoria cache? ¿Qué ventajas puede introducir su uso en una arquitectura de computadoras?**

Una memoria cache es un espacio intermedio entre el procesador y el resto de la memoria. Sus características son: Es estática; es pequeña y más rápida; tiene un bus exclusivo para comunicarse con el procesador; está entre la memoria y el CPU; también se encuentra dentro del chip del CPU, son más caros y rápidos comparados a la RAM.

Funciona así: El CPU requiere una dirección de memoria, revisa si está en el cache, si está lo carga desde ahí directamente, si no está, lo busca en la memoria principal y lo carga en el cache, enviándolo desde ahí al CPU. En el cache incluye tags (etiquetas) para saber de qué parte de la memoria se extrajo dicha información.



La ventaja de usar una memoria caché es el tiempo que nos ahorra buscar en la memoria principal: En lugar de buscar cada pedazo de código de la memoria una tras otra y luego volver a buscar el siguiente pedazo y así, carga el bloque de código completo desde la memoria al caché, y de ahí lo va leyendo a mayor velocidad gracias a la naturaleza de esta memoria, a evitar interrupciones y su bus exclusivo que evita cuellos de botella dado por el uso de buses compartidos (como la memoria caché tiene bus exclusivo, nos ahorramos ese problema). LOS ACCESOS A MEMORIA SON MUCHO MÁS LENTOS QUE LA VELOCIDAD DE LAS INSTRUCCIONES, LO QUE GENERA CUELLOS DE BOTELLA. Si aprovechamos los principios de localidad espacial y temporal, el guardar los datos a usar en el caché nos permitirán ahorrarnos estos cuellos de botella.

10. **¿Qué tipo de memoria RAM es la memoria cache? Describa su funcionamiento en R/W.**

Es un tipo de memoria asociativa y una SRAM (Static RAM).

No actúa como embudo (si bien está entre la memoria principal y el procesador, sino que forma parte de la arquitectura como un dispositivo más que comparte un canal)

LOS ACCESOS A MEMORIA SON MUCHO MÁS LENTOS QUE LA VELOCIDAD DE LAS INSTRUCCIONES, LO QUE GENERA CUELLOS DE BOTELLA. Si aprovechamos los principios de localidad espacial y temporal, el guardar los datos a usar en el caché nos permitirán ahorrarnos estos cuellos de botella.

Es más rápido y caro (en términos relativos). No necesita ciclo de refresco (Porque está compuesto por flip-flops)

El caché está dividido en líneas (mientras que la memoria está dividida en bloques). Un bloque de memoria cabe en una línea de caché.

Una memoria caché es una memoria RAM estática, que no participa del gran lugar donde se almacenan las instrucciones, sino que es un lugar especial que se utiliza para almacenar las instrucciones y datos que más se utilizan. Persigue como ventaja mejorar el desempeño de todo el sistema lo logra a través de varios ciclos de desempeño guardando en ella aproximadamente las instrucciones que más frecuentemente se utilizan ahorrando así buscarlas en la RAM (No es que es las que más se utilizan en sí; en realidad lo que hace es guardar el código que vayamos a estar ejecutando allí, de forma tal que podamos acceder pronto). Lo logra ya que la memoria cache es más chica que la RAM y tiene un método de búsqueda más ágil, además de que está en un canal de acceso más veloz que el resto.

En el caché se guardan pequeños bloques de código, así tiene menos lugar en el que buscar a la hora de cargar un programa. Si el caché fuera tan grande como la RAM, perdería su propósito, ya que terminaría teniendo una velocidad similar por tener que buscar en la misma cantidad de espacio.

Es un tipo de memoria asociativa.



El overhead de un caché es el espacio y tiempo extra requerido para guardar los datos de un caché, necesario para la organización del espacio a manipular (no es necesariamente para un caché, pero vale la pena mencionarlo). En un caché de mapeo directo, el overhead podría ser el bit de validez y las etiquetas.

11. **¿Cuántos niveles de cache puedo llegar a encontrar en una computadora, y dónde están?**

De nivel 1, apareciendo como una pequeña memoria extra para aumentar el espacio de trabajo de los procesadores.

De nivel 2, es el correspondiente externo del nivel 1, pero de forma externa y en la placa Cache interna. Nivel 1. Físicamente está ubicada en el mismo chip que el procesador. Los accesos a esta cache se efectúan muy rápido. La capacidad de esta cache es bastante pequeña.

Cache externa. Nivel 2. Físicamente se ubica fuera del procesador por lo que será más lenta que la cache de nivel 1 pero seguirá siendo más rápida que la memoria principal. Al estar fuera, la capacidad podrá ser mayor que la cache de nivel 1.

12. **¿Qué entiende por cache unificada y qué por cache partida? Indique si presentan ventajas.**

Caché unificado: Se usa un solo caché para instrucciones y datos, pero es más lento que el caché partida.

Caché partida (Split cache): Dos cachés independientes que operan en paralelo, uno para instrucciones y otro para datos. Si tiene canales de accesos distintos, puede hacer operaciones de lectura de forma simultánea (ganando más velocidad).

13. **¿Qué tipo de correspondencia puedo establecer entre el cache y la memoria RAM principal?**

Directa, asociativa y asociativa por conjunto.

14. **Describa el funcionamiento de la correspondencia Directa. Indique ventajas y desventajas.**

-A cada dato se le asigna un lugar en el caché de acuerdo a la ubicación que tenía en la memoria principal, y siempre le corresponderá esa misma posición. Esta relación se da mediante a este mecanismo/operación:

$(\text{bloque en caché}) = (\text{dirección de bloque}) \bmod (\text{número de bloques del caché})$

Mod es la operación del resto de una división.

-Si yo quiero indexar un caché con "x" cantidad de bloques, voy a necesitar encontrar un número "n" de forma tal que 2^n me de "x". O sea: $x = 2^n$. Ejemplo: quiero direccionar un caché de 8 bloques, entonces "n" debe ser 3 ya que $8 = 2^3$;

Ahora bien, ¿cómo se relacionan las posiciones de memoria con las posiciones del caché?

-Bueno, como ya se mencionó, si el caché es de 8 bloques, lo indexamos con 3 bits, acá está el truco: Si una posición en memoria termina en determinados 3 bits, esos 3 bits serán la



posición en el índice del caché. O sea que a las posiciones 00000, 001000, 10000 y 11000 les correspondería el índice 000; a las posiciones 00001, 001001, 10001 y 11001 les correspondería el bloque de caché con índice 001; y así sucesivamente.

-Para asegurar que el dato sea correspondiente a la dirección de memoria que buscamos y no otra que comparta su índice, usamos etiquetas (tags). Cada bloque tiene su tag y bit de validez que indicarán si el dato en el caché es el dato buscado; esta etiqueta, estará dada por los bits más significativos de la dirección de memoria. Si recordamos que usamos los 3 bits finales para indexar los bloques, los bits más a la izquierda que no usamos para indexar los bloques, serán los que usaremos para los tags (si eran 00001, 01001, 10001, 11001 y todos corresponden al bloque en caché 001, entonces los tags serían 00, 01, 10 y 11). El bit de validez nos indicará si hay datos dentro de ese espacio en el caché: si el bit está en false, trae los datos de memoria directamente; si el bit está en true, compara el tag, si es la misma, usa los datos del caché; si el bit está en true, compara el tag pero no es el mismo tag que busca, se produce un fallo y debe buscar el nuevo dato en memoria.

El mayor inconveniente que tiene es que gracias a que cada posición en memoria siempre tendrá el mismo sitio en caché, cuando a dos posiciones de memoria les corresponda el mismo espacio en el caché se producirá un “miss”, incluso habiendo más espacio en el caché que podría haber sido aprovechado. Su mayor ventaja, es que lo que yo busque ya va a estar mucho más próximo dentro de una memoria estática. Tiene más tasa de falla, pero es sencilla y menos costosa de implementar.

15. **Describa el funcionamiento del asociativo total. Indique ventajas y desventajas.**

En este sistema, los bloques se pueden colocar en cualquier posición del caché, comprobando la etiqueta de todos al buscar un dato hasta encontrar un acierto. Su desventaja es que debe comprobar cada tag por separado, realizando muchas más comparaciones, costosas en circuitería si las queremos en paralelo, además de que las comparaciones llevan tiempo (velocidad). Su ventaja es que aprovecha todos los bloques del caché. El tamaño del caché no está determinado por el tamaño de la memoria, ya que cualquier línea puede ir en cualquier bloque del caché.

16. **Describa el funcionamiento de la asociativa por Conjunto. Indique ventajas y desventajas.**

Cada bloque de la memoria principal tiene asignado un conjunto en el caché, donde se puede ubicar libremente (como si fuera la asociativa) en cualquier línea del conjunto. El conjunto al que pertenecerá cada bloque estará dado por un mapeo directo. (A cada bloque de la memoria se le asigna por mapeo directo un conjunto del caché, pero dentro de ese conjunto se podrá ubicar en cualquier posición, como en la forma asociativa). Esto nos brinda mucha más flexibilidad que la correspondencia directa pero sin tener que hacer la gran cantidad de comparaciones que realiza la asociativa total.

17. **Describa el método de actualización Write Through y Write Back. Indique para cada caso si presenta ventajas o desventajas, descríbalas si correspondiera.**



Write-Through: Escribe cada vez en el caché y en la memoria (se actualiza inmediatamente). Su desventaja es que es más lento, escribir en el caché y la memoria puede reducir la velocidad en un factor de 10.

Write-Back: Escribe únicamente en el caché y copia el dato a la memoria cuando la entrada en el caché va a ser reemplazada. (Se actualiza el contenido de la memoria cuando el del caché está por ser reemplazado)

18. **¿Por qué se necesita tener una política de ubicación en la gestión del cache?**

La necesidad de políticas de Actualización, Sustitución, y Ubicación dentro de una memoria caché son necesarias ya que la memoria cache es una memoria chica, por lo tanto es probable que no alcance el espacio para nuevas instrucciones o datos que quiera procesar o traer desde la memoria principal, por ello hay que reemplazar datos viejos para dar lugar a los nuevos.

Necesito saber dónde ubicar los datos dentro del pequeño espacio del caché.

19. **¿Por qué se necesita tener una política de sustitución en la gestión del cache?**

Necesito saber qué sacar en el pequeño espacio del caché para poner nuevos datos.

20. **¿El uso del Firmware en circuitos con componentes electrónicos qué ventajas brinda?**

-Deja una secuencia de pasos necesaria o frecuentemente utilizada, almacenada permanentemente a disposición del usuario y/o una secuencia de datos necesaria a manera de tabla.

-Funciona de interfaz entre en usuario y el hardware para evitar dejar el correcto funcionamiento a manos del usuario.

-Cuando necesito una secuencia de instrucciones o una serie de datos para que funcione el dispositivo, lo incrusto en una ROM, y eso es el firmware. Un algoritmo codificado almacenado permanentemente dentro de un dispositivo.

-El firmware me indica de qué manera puedo utilizar el dispositivo.

En el caso de una computadora, la ventaja de tener los programas fundamentales almacenados en la ROM consiste en que están constituidos en el ordenador, y no es necesario cargarlos en la memoria desde el disco como el DOS (Disk-based Operative System). Ya que son permanentes, los programas de la ROM se usan a menudo como la base para construir otros programas (incluido el sistema operativo).

21. **¿Qué ventaja aportó la incorporación del BIOS a la Motherboard de una PC?**

Que las instrucciones de mayor uso vengan incluidas en el firmware, todas dadas por el fabricante. Esto nos brinda la posibilidad de manejar el hardware sin saber cada pequeña parte del mismo, y también nos ayuda a evitar hacer cosas que no van. La ROM BIOS es la parte del ROM que está siempre activa mientras el equipo trabaja, y su función fundamental es la de proporcionar los servicios requeridos para la operatividad de la



computadora. Controla principalmente periféricos del equipo, cosas como el teclado, la pantalla y las unidades de disco.

22. **¿Qué significa POST, cuándo se realiza y en qué consiste?**

Power On Self Test o POST es un proceso realizado por el firmware o el software inmediatamente después de que se prende la computadora para verificar que todos los componentes físicos estén funcionando correctamente. (Principalmente busca evitar errores eléctricos, bloqueando la fuente en el caso de que haya algún valor eléctrico fuera de rango al iniciar el bios).

23. **¿Qué entiende por construcción del Vector de Interrupciones?**

La rutina de arranque separa una parte de la memoria RAM (la parte baja) donde se almacenará el vector de interrupciones. En este vector se almacenarán las direcciones de las rutinas para las interrupciones que se pueden llegar a producir. Como inicialmente no sabemos qué clase de rutinas habrá, primero se cargarán las direcciones de las rutinas del BIOS (que son las que no variarán), dejando espacios vacantes para otras posibles rutinas. Esto tiene una ventaja: si nuestro sistema operativo tiene rutinas mejores o diferentes, podremos cambiar la dirección al apuntador de rutina sin modificar la ROM: supongamos que nuestra ROM incluye la rutina de manejo de teclado pero nuestro sistema operativo tiene opciones extra para teclados de otros países u configuraciones de teclas adicionales; en lugar de tener que cambiar la ROM, simplemente cambiaremos la dirección en el vector de interrupciones para que apunte a la nueva rutina con teclados de otros idiomas y teclas personalizables.

24. **¿Qué función cumple el Setup y dónde se encuentra físicamente?**

El setup nos permite configurar varios parámetros de dispositivos fuera del estándar, como discos, entre otras, y se encuentra en el CMOS. Esto es muy importante, ya que en el setup podemos definir qué disco será el de arranque y contendrá la dirección para poder comenzar la rutina que carga el sistema operativo.

25. **¿Cuál es la función de la memoria CMOS en el arranque, qué ventaja introdujo su uso?**

El CMOS sirve para guardar parámetros para el arranque. Es una memoria RAM de muy bajo consumo energético (distinta a la de trabajo), que siempre se mantiene encendida y guarda cosas como la hora y todos los parámetros de dispositivos distintos al estándar. En un principio, servía para almacenar parámetros de los dispositivos y que, al arrancar nuestro equipo, el BIOS comparara sus datos con los de esta memoria, a fin de que si hay un match podamos seguir tranquilamente y en el caso contrario nos lo indique. Cosas como la identificación y configuración de dispositivos se almacenan en el CMOS. Parámetros como los de los discos, se deberían haber cargado manualmente en cada arranque del equipo si no fuera porque la memoria CMOS los guardaba, cosa que requería tiempo y daba espacio al error humano a la hora de cargarlo (esto ya no pasaría hoy en día porque todo suele tener un sistema de auto-detección plug n play). En el CMOS podemos definir qué disco será el de arranque y contendrá la dirección para poder comenzar la rutina que carga el sistema operativo.



26. **¿Cuál es la tarea que tiene a cargo la etapa de carga del sistema operativo dentro de las rutinas del arranque?**

La tarea que tiene a cargo la etapa de carga del sistema operativo dentro de las rutinas del arranque se llama "cargador de la secuencia del arranque". Una vez terminadas las verificaciones previas, el BIOS inicia un recorrido en busca de un dispositivo donde encontrar un programa que pueda continuar el proceso con la carga del sistema operativo; Este recorrido se denomina secuencia de carga y su orden puede ser establecido mediante un programa auxiliar (Setup del BIOS). La secuencia de carga es almacenada en la memoria de datos del BIOS, de forma que pueda ser recordada y utilizada la próxima vez que se reinicia. Básicamente, se encarga de darle run al sistema operativo, buscando su dirección de inicio.

Las rutinas de arranque hacen las siguientes tareas:

- Ejecutan un test de fiabilidad del ordenador y los programas de la ROM para asegurarse que todo esté bien.
- Inician los chips y el equipo conectado al ordenador.
- Establecen la tabla de vectores de interrupción.
- Exploran para descubrir el equipo opcional conectado al ordenador.
- Cargan el sistema operativo del disco.

27. **Definir qué se entiende por interrupción en una computadora.**

Una interrupción es una operación o pedido del hardware que indica al CPU que debe pausar la ejecución de lo que está haciendo, conserva el estado del proceso y envía a la CPU una dirección en la que tiene un administrador de interrupciones. A la hora de hacerlo, debe asegurar que el proceso que estaba haciendo quede de forma segura, de forma tal que se pueda continuar una vez finalizada la interrupción.

28. **¿Por qué es ventajoso usar un sistema basado en interrupciones?**

Nos permite trabajar y comunicarnos con dispositivos de diferentes velocidades sin tener que esperarlos (nos permite sincronizarlos). Si no fuera de esta forma, nuestro CPU que es muchísimo más rápido que el resto de dispositivos, estaría bajo una "espera dinámica", o sea que estaría esperando a un dispositivo mucho más lento para poder continuar.

El sistema de interrupciones, nos permite solicitar a los periféricos un servicio, y no monitorearlos ni esperarlos, sino que dichos periféricos nos vuelvan a solicitar atención



cuando hayan terminado, de forma tal que el CPU no tenga que esperarlos para seguir haciendo su trabajo.

29. **¿Qué ventajas introducen el uso de las interrupciones lógicas?**

Nos permite atender solicitudes solamente cuando se hacen y no tener que estar verificando constantemente si algún periférico, dispositivo o lo que sea necesita atención por parte del CPU. Un ejemplo muy útil es este: Debemos imaginar una computadora que todo el tiempo está verificando constantemente si se conectó un mouse, esto obviamente le llevaría tiempo y recursos; ahora, pensemos que la computadora debe verificar un mouse, un teclado, una pantalla, un auricular y muchísimas cosas más, consumiendo así infinidad de recursos y ralentizando a nuestro procesador muchísimo en la espera de dispositivos lentos. El uso de interrupciones nos permite que en lugar de que el procesador (que es rápido) pierda tiempo esperando a dispositivos, los dispositivos simplemente le avisen cuando requieran de su uso.

30. **¿Qué función cumple el vector de interrupciones cuando se produce una interrupción?**

¿Sería posible atender un pedido de servicio sin Vector Interrupciones, si fuera así que inconveniente traería?

El vector de interrupciones nos permite asociar una lista de direcciones de rutinas con sus correspondientes interrupciones. Un ejemplo sería si yo conecto un teclado, un mouse y un pendrive: el vector de interrupciones asocia las rutinas de manejo de interrupciones con las solicitudes de interrupciones, de forma tal que el teclado sea atendido por la rutina del teclado, el mouse por la rutina del mouse, y el pendrive por la rutina del pendrive.

Si bien sería posible atender a un pedido de servicio sin el vector de interrupciones, ¿cómo sabría que rutina utilizar ante determinada interrupción? Esto nos llevaría al problema de que quizás atendamos a la petición del teclado con la rutina del mouse, o usemos la rutina del mouse para atender al pendrive. El vector de interrupciones nos permite atender todo de manera organizada.

Algo importante a aclarar: en el vector de interrupciones solamente están las direcciones de las rutinas, estas direcciones apuntan hacia el lugar donde están las rutinas a utilizarse. El vector es un espacio pequeño y por ende no almacena el código de las rutinas.

31. **Determinar la diferencia entre las interrupciones enmascarables y la no enmascarables.**

Indicar ventajas y desventajas de cada una desde el punto de vista lógico y físico.

La diferencia fundamental entre las interrupciones enmascarables y no enmascarables reside en lo siguiente:

No enmascarables (NNI): Estas interrupciones de deben ejecutar obligatoriamente sin tener que esperar, y están ligadas a tareas que no se pueden retrasar (bajada del sistema por corte de energía, chequeo de la integridad de la memoria, entre otras). 3 ejemplos de interrupciones no enmascarables: 1 puede venir del coprocesador matemático Intel 8087,



falla en la paridad de la RAM o de algún canal de E/S. En la práctica, se pueden enmascarar por software, pero de todas formas están definidas como no enmascarables.

Las enmascarables (INTR): A diferencia de las no enmascarables, estas interrupciones sí pueden esperar y se ejecutan mediante una solicitud al periférico correspondiente siempre y cuando en el registro de flags las interrupciones estén habilitadas (hay operaciones que no se pueden interrumpir, en ese caso se deshabilitarían las interrupciones y por lo tanto no podrían pasar las interrupciones enmascarables, mientras que las no enmascarables ignorarían todo y pasarían de todas formas). En el caso de estar deshabilitadas las interrupciones, el CPU las ignoraría y seguiría con lo que estaba haciendo.

32. **Describir el funcionamiento de los registros del microprocesador cuando se produce una interrupción del programa en ejecución, indicar que valores son almacenados y cómo trabaja el Stack Pointer. ¿En qué lugar se debería construir el stack para su seguridad?**

Guarda las cosas que le permiten continuar la rutina en curso (IP, para saber por dónde continuar una vez finalizada la interrupción) en el stack pointer (las apila), una vez listo pasa a la tarea dada por la interrupción (la completa y la desapila), una vez “desapilada” la instrucción más reciente, puede continuar gracias a los registros guardados en el stack pointer (si desapilamos lo anterior, nos permite pasar a la instrucción siguiente, que sería la que guardamos antes de interrumpir con la nueva).

Se construye en el área de memoria que corresponde al sistema operativo o en un segmento de memoria reservado a fines de guardar estos datos sin que se sobrescriban.

(NO ES LA RESPUESTA A LA SEGUNDA PREGUNTA): En el BCP o bloque de control de proceso, almacena todos los datos de la rutina que se fue para darle paso a la rutina que está por interrumpir.

SP o Stack Pointer contiene la dirección del tope vacante de la pila.

33. **Dado las limitaciones físicas que presenta un microprocesador en la cantidad de entradas de interrupciones, que ventajas le aporta el uso de un controlador (administrador) de interrupciones, indique ventajas y desventajas.**

La ventaja del administrador físico de interrupciones es que funciona como un multiplexor con una lógica propia de prioridades de señal; suponiendo que recibe 5 señales, primero tomará la que tenga una prioridad superior, hasta el punto que una interrupción puede interrumpir a otra de menor prioridad. Esto, al ser en formato físico y no ir por software, nos permite ganar mucha velocidad y liberar al CPU.

Las desventajas son: usa un dispositivo electrónico inteligente (ocupa espacio y \$\$\$)



34. **¿Qué función cumplen los dispositivos de Entrada, Salida, y E/S? ¿Por qué son necesarios?**

Permiten a la computadora comunicarse con el mundo exterior, pudiendo recibir señales y datos que alteren su comportamiento y sus respuestas. De nada nos serviría una computadora si solo pudiera calcular valores definidos de fábrica, y tampoco nos serviría si no hiciera nada con esos valores. Por ello, las entradas nos permiten ingresar datos y las salidas nos permite a nosotros como usuarios recibir respuestas por parte de la máquina.

35. **¿Cuáles son las funciones de un controlador de dispositivo de E/S?**

El controlador de dispositivo de E/S nos permite tener un administrador por separado que controle las E/S para que así no deba hacerlo el procesador, ya que lo ralentizaría esperando a dispositivos lentos (siendo que el procesador puede trabajar rapidísimo en comparación). Su electrónica incluye controladores y chips especializados que le permiten recibir órdenes por parte del procesador, de esta manera lo libera de realizar las tareas de bajo nivel y con cualquier tipo de dispositivo. Todo esto teniendo en cuenta que los dispositivos de E/S son distintos entre sí, con lo cual el controlador además asegura que el procesador no deba controlar procesos de velocidad indefinida.

Actúa como interfaz entre la CPU y el dispositivo E/S (cada controlador puede encargarse de uno o varios dispositivos); control y temporización del flujo de datos; comunicación con la CPU (intercambio de datos de los dispositivos E/S con el CPU, informar del estado del dispositivo, reconocimiento de la dirección del dispositivo); comunicación con el dispositivo; almacenamiento temporal de datos (buffer); detección de errores

36. **Según la visión del microprocesador ¿cuáles son los espacios de direccionamientos?**

Los espacios de direccionamiento serían: Registros internos (ax,bx,etc.), la RAM y los dispositivos E/S.

Los controladores de dispositivos E/S actúan como interfaz con el CPU y la memoria, y, a su vez una interfaz externa que la comunica con los dispositivos E/S. Aparte de eso, se direcciona como otros espacios en la memoria, pero este a su vez actúa como interfaz con los diferentes dispositivos conectados.

37. **¿Cómo se puede mapear las direcciones de E/S?**

Tiene 2 opciones:

Espacio de direcciones propio de E/S: Dispone de un espacio de direccionamiento separado para los dispositivos E/S (además de un espacio para el direccionamiento de los registros internos y otro para la RAM, se le añade uno para los dispositivos E/S). Tiene 2 métodos de implementación. 1) Instalar un nuevo conjunto de buses de datos, dirección y control. 2) Compartir las señales existentes en los buses y añadir una nueva señal de control que indica si la operación en curso hace referencia al espacio E/S.

Compartiendo espacio de direcciones de memoria (direcciones mapeadas en memoria): Reserva direcciones de memoria para los dispositivos E/S. Dado que el mapa de memoria suele estar compuesto por zonas de memoria RAM, ROM y zonas de direcciones libres. Este



método se sirve de las mismas para aprovecharlas y reservarlas como espacio de direccionamiento hacia las E/S. ¿La desventaja? Ocupa espacio que podría ser ocupado para otras cosas. ¿Importa la desventaja? Casi no afecta en nada, ya que como hoy en día el direccionamiento es de 64bits, un grupo de E/S ocupa un espacio insignificante en comparación.

38. **¿Cuáles son las técnicas de E/s explícitas?**

Por sondeo (polling), interrupciones, o DMA (Direct Memory Access). Todas cumplen con una etapa de sincronización y otra de transferencia de datos. Por polling se usa el software/CPU tanto para la sincronización como para la transferencia de datos. Por interrupciones se usa el hardware para la sincronización y el software/CPU para la transferencia de datos. Por DMA se usa el hardware tanto como para la sincronización como para la transferencia.

39. **¿Por qué en la comunicación de E/S existe una etapa de sincronización?**

Porque como las E/S requieren comunicarse con el mundo exterior, su producción de datos proviene del mismo, variando mucho la velocidad de producción de datos con el consumo de los mismos por parte de la computadora. Un teclado debe esperar a que un usuario presione sus teclas, y como no sabemos cuándo va a pasar, necesitamos algo que lo sincronice.

El procesador y los dispositivos E/S hacen tareas distintas y trabajan a velocidades distintas, así que necesitarán sincronizarse antes de hacer transferencias entre sí. Además, el procesador no puede recibir información si no está listo, y quizás cuando esté listo no haya información para recibir, así que la sincronización se encarga de emparejar todo.

40. **La técnica de sondeo o polling ¿qué desventaja presenta?**

Es más lenta, ya que la sincronización y la transferencia de datos se hacen por software/CPU, cosa que además le consume recursos al procesador que debe esperar a un dispositivo lento. Funciona mediante un ciclo que verifica repitiéndose infinitamente si el dispositivo está listo

41. **¿Qué beneficio aporta el uso de interrupciones en el proceso de comunicación E/S?**

En comparación a la técnica de polling, en lugar de verificar continuamente si el dispositivo está listo continuamente, aprovecha el uso de interrupciones: El dispositivo solicitará una interrupción inmediatamente cuando esté listo, permitiéndonos salvar el tiempo que antes se utilizaba únicamente para verificar el estado del dispositivo.

42. **¿Qué ventajas o desventajas introduce el empleo del DMA en una computadora?**

Nos permite realizar operaciones que serían prácticamente imposibles con los métodos de interrupción y polling, como lo son el manejo de discos rígidos (en los cuales el manejo por interrupciones o polling sería imposible ya que si es una interrupción por cada bloque de datos transferidos ocuparía todo el CPU en ello, además de que limitaría la velocidad de transferencia de datos a la velocidad del CPU debido a que el mismo debería realizar las



operaciones de comprobar, mover, incrementar puntero, salvar estado, restaurarlo, incrementar contador, etc.), además, si lo hiciéramos por polling, ralentizaríamos la computadora muchísimo, debido a que debería emparejarse con el dispositivo E/S. Por eso recurrimos al empleo del DMA, que permite hacer esta tarea mediante un hardware específico, liberando al CPU para que pueda hacer sus propias tareas.

DESVENTAJA: Se requiere el hardware (espacio y dinero), el DMA ocupa el bus de datos (esto tiene solución y además aún si lo hace, es por un tiempo menor antes que producir una interrupción contra el CPU)

43. **¿Cómo funcionan los registros internos de un DMA al realizar una transferencia directa?**

Primero el procesador revisa si el DMA está disponible, si se produce un handshake (ambos están listos), el procesador configura y le envía los parámetros necesarios al DMA (Le da dirección de inicio, dirección de destino, operación R/W, número de bytes a transferir y contador). Una vez enviados los parámetros, el DMA empieza a trabajar en forma autónoma y el procesador se puede dedicar a sus propias tareas. A partir de la dirección de origen, comienza a transferir de a tamaño de buffer; en cada transferencia exitosa decrementamos el contador. Si todo tiene éxito el contador llega a 0, y se genera una interrupción al procesador para avisar que se finalizó la tarea. Si una transferencia no tiene éxito la vuelve a intentar como máximo 3 veces aproximadamente y corta.

Cuando la FDC tiene un dato listo para transmitir activará la pata de DREQ (data request) solicitando la posesión del bus, cuando el DMA tenga el control del bus, activará las señales de lectura IOR y escritura MW y le indicará a la memoria la dirección del dato procedente de la FDC, quien lo enviará al bus de datos para luego desactivar todas las señales mencionadas (DREQ, lectura y escritura, y DACK) hasta tener un nuevo dato, cosa que reiniciará el proceso mencionado y decrementará en 1 el contador. Si el contador llega a cero, envía una señal EOP end of process indicando que el proceso terminó, lo mismo con el FDC que activará su pata INT para indicarle al CPU que la lectura del sector terminó.

44. **¿Cómo soluciona el conflicto entre DMA y CPU al compartir el bus de sistema?**

El DMA busca apoderarse del bus, así que tiene estas opciones: MODO BLOQUE (BURST): Acumular datos hasta llenarse, apoderarse del bus por la fuerza, transferirlos todos juntos y luego ceder el uso del bus; ROBO DE CICLO: Donde se intercala el uso del bus entre procesador y DMA (se turnan), esto es más lento que el modo burst gracias a la intercalación, pero interfiere menos en el uso del bus; MODO TRANSPARENTE: En este modo el DMA solamente roba ciclos cuando el CPU no requiere el uso del bus (ya sea porque está realizando otras operaciones o porque en su lugar se está comunicando con el caché).

Algo importante a recordar, muy conectado con el tema del caché, es que como el CPU usa un bus de datos independiente para comunicarse con el caché, una vez que el caché tomó la información de la memoria, el procesador puede comunicarse con el caché a través de su bus propio, liberando el bus de datos.

45. **¿Cuál es la estructura interna de la superficie magnetizable de un disco rígido?**



Cada uno de los platos del disco rígido tiene dos superficies magnéticas (superior e inferior), las cuales están formadas por millones de elementos minúsculos capaces de ser magnetizados de forma positiva (1) o negativa (0), representando un bit de información.

Podemos agregar que la forma de cambiar la polaridad y leer el disco es a través de un cabezal magnético que se mueve mediante un brazo mecánico, que se mueve por sobre el disco de forma muy cercana pero sin tocarlo, gracias a la película de aire que se forma entre él y la superficie de metal debido a la gran velocidad a la que giran los platos.

46. **¿Qué entiende por formato de alto y bajo nivel de un disco rígido?**

Nosotros no podemos usar nuestro disco tal y como sale de fábrica, ya que nuestro sistema operativo no lo reconocería. Primero hay que darle un **formato de bajo nivel**, una o más **particiones** y luego un **formato** de archivo que pueda reconocer nuestro sistema operativo.

En el formato de bajo nivel, se recorre la superficie magnetizable y se la rotula. Esto hace que se divida en surcos y sectores, definiendo bloques. La superficie del disco se divide en surcos, separados por intersurcos (serían los anillos o cilindros), y sectores (que serían como cortadas de pizza) divididos por intersectores (los intersectores e intersurcos son espacios vacíos que “separan”). El tamaño mínimo de un bloque es de 512 byte. El formato de bajo nivel suele venir hecho de fábrica, de forma tal que el que lo diseñó sabe la mejor forma de aprovechar la memoria.

El formato de alto nivel está dado por el sistema operativo, que define qué utilidad se le dará a cada bloque/cilindro/sector. Podrá definir qué bloque será el área de booteo, cuál el del sistema de archivos, directorio de raíz, lugar de datos, etc.

47. **¿Qué tiempos tienen lugar en la recuperación de datos dentro de un disco rígido, desde que se da la orden hasta que el dato está disponible?**

Tiempo de búsqueda (el que se tarda la cabeza del lector en posicionarse en la pista a leer).

Retardo rotacional o latencia rotacional (tiempo que el controlador de disco espera hasta que por la rotación el disco se alinee con la zona a leer).

Tiempo de acceso (la suma del tiempo de búsqueda y el de rotación).

Tiempo de transferencia de datos (tiempo de lectura o escritura con la cabeza posicionada).

Otras demoras (demoras asociadas a operaciones E/S, cuando se solicita una operación con un dispositivo E/S se suele tener que esperar a que esté disponible, recién en ese momento comienza la búsqueda)

48. **¿Por qué el interleave era un valor relevante, y qué buscaba reducir cuando se le daba valor mayor a 1? ¿Qué sucede en la actualidad con dicho valor?**



El interleave es importante porque nos permite usar un mismo bus para acceder a distintos bancos de memoria, cosa que nos permite repartir los datos en distintos bancos de memoria sin pagar por más canales (es más lento que un bus ancho de múltiples buses, pero sigue siendo mejor que un ancho de bus de una palabra, y en costo es menor que un bus ancho, cosa que lo hace el más usado). Reduce el castigo por falla y cada banco puede escribir en forma independiente, cosa que disminuye las detenciones en write-through. Si el bloque es de un tamaño mayor a una palabra, nos permite aprovechar el principio de localidad espacial y traer instrucciones contiguas para que se ejecuten consecutivamente. Si el bloque es de más de una palabra, reduzco el espacio requerido para las etiquetas y la tasa de falla, pero mientras mayor sea el tamaño del bloque, más aumento el castigo por fallas. En interleaving, si hay más bancos, podremos usarlos de forma independiente, cosa que nos permite usar los bancos de forma continua para R/W, usando cada banco turnándose en lugar de el mismo banco en forma repetitiva, cosa que mejora la velocidad ya que hay menos tiempo de espera entre el uso de un banco que usando un mismo banco de forma repetitiva.

49. **¿Qué significa la palabra RAID?**

Redundant Array of Independent Disks o arreglo de discos con redundancia en español. Combina diferentes drives en una unidad, permitiendo crear redundancias, mejorar el rendimiento de lectura y frente a fallas.

Junta un grupo de discos para mejorar la calidad de la recuperación del dato (ganar confiabilidad y poder superar fallas).

Existen de 6 tipos (de los cuales se usan 3 y principalmente 1), no son jerárquicos, los datos se distribuyen a través de los distintos discos, pueden usar redundancia para guardar información de paridad.

Son varios discos que el sistema interpreta como uno solo. El dato se reparte entre los diferentes discos, mejorando su lectura (podemos leer en paralelo con las distintas cabezas de lectura/escritura de los diferentes discos, ganando velocidad)

50. **¿Qué procedimiento previo se debe hacer con un archivo que se graba en un RAID?**

Se formatea partiéndolo en pedacitos que se distribuirán a través de los distintos discos (¿?)

51. **¿Qué ventajas y desventajas presenta el empleo de RAID?**

Ventajas: Mejora la velocidad de lectura gracias a que puede leer en paralelo diferentes discos. Todos los RAID excepto el 0 permiten recuperar datos perdidos gracias a un disco arruinado mediante el aprovechamiento de datos de otros discos (y aprovechando un algoritmo).



Puede usarse para mejorar la fiabilidad de los datos y la recuperación de los mismos en caso de fallo.

Desventajas: Tener muchos discos cuesta plata.

52. **¿Todos los RAID tienen capacidad de recuperar datos perdidos? Justifique su respuesta.**

Todos pueden colaborar a la recuperación de datos perdidos excepto el RAID 0, que solamente se usa para mejorar la velocidad de lectura.

53. **En el caso de utilizar un arreglo de discos redundantes en un servidor informático, ¿de qué clase deberían ser los discos? Justifique su respuesta.**

Hot-swap, para no apagar el servidor a la hora de cambiar un disco

54. **¿Qué se entiende por paridad en un RAID, para que se la utiliza?**

La paridad son partes de datos que se utilizan para reconstruir los datos que se puedan perder por la caída de un disco.

55. **Describe RAID 1+0, RAID 3 y RAID 5.**

RAID 0: No tiene redundancia. Los datos son diseminados a través de todos los discos mediante un round robin. Incrementa la velocidad (el requerimiento de datos que quizás no estén almacenados en el mismo disco, los discos se buscan y leen en paralelo, un bloque de datos se distribuye en distintos discos).

RAID 1: Discos espejados. Los datos se distribuyen en distintos discos. Se realizan 2 copias de cada bloque en discos separados. Se leen desde cualquiera de ellos. Se escribe en ambos. La recuperación es simple. Es caro. Es básicamente como tener el mismo disco más de una vez, dándonos más seguridad y velocidad de lectura.

RAID 2: Los discos están sincronizados. Bloques muy pequeños (normalmente palabras o bytes únicos). Corrección de errores calculados en base a los bits correspondientes en los discos. Múltiples discos de paridad almacenan códigos de error Hamming. Mucha redundancia. Caro. NO SE USA.

RAID 3: Similar a RAID 2. Un solo disco redundante, sin importar el tamaño del array. Los datos en discos que fallan se pueden reconstruir en base a los datos sobrevivientes y la información de paridad. Muy altas transferencias de datos.

RAID 4: Cada disco opera independiente. Bueno para grandes requerimientos de E/S. Largos bloques de datos. Paridad bit a bit se calcula en cada disco. La paridad se almacena en discos de paridad.

RAID 5: Similar a RAID 4. Paridad distribuida a través de todos los discos. Usada normalmente en los servidores de red.

56. **¿Qué entiende por código de máquina, indicar si cada procesador presenta un código diferente?**



RISC: Reduced Instruction Set Computer. Menos instrucciones, todas miden lo mismo (tamaño predecible, permite pipelining).

CISC: Complex Instruction Set Computer. Más instrucciones, distintos tamaños (tamaño de instrucción no predecible)

57. ¿Qué ventajas introdujo usar lenguajes de alto nivel para codificar Sistema Operativos?

¿Introdujo alguna desventaja? Si es así ¿Cuál?