

Informática
Guía Teórica
Ingeniería en Mecatrónica

BORQUEZ PEREZ, Juan Manuel-13567

Arquitectura de computadoras

1. ¿Por qué las computadoras se estructuran en capas?

La computadora está limitada a realizar operaciones aritméticas con números (suma, resta, multiplicación y división), realizar operaciones lógicas (por ejemplo realizar la comparación de dos números) y mover datos de una región de la memoria a otra. Estas acciones se comunican directamente a la computadora a través de instrucciones codificadas en una serie de bits (lógica digital), es decir un conjunto de unos y ceros que en el soporte físico están representados por niveles de tensión/corriente altos o bajos y que constituye el lenguaje que la máquina puede entender (lenguaje máquina). Una serie de instrucciones constituyen un programa, pero escribir las mismas en el lenguaje máquina es complicado e inconveniente para el usuario. Para eliminar esta complejidad el sistema de cómputo se estructura en capas, cada una de las cuales permite una abstracción de las capas inferiores y oculta la complejidad de las mismas aportando solo la funcionalidad más relevante a las capas superiores.

2. Describa la relación entre capas, máquinas virtuales y lenguajes

Cada capa de abstracción aporta funcionalidad a las capas superiores y se vale de la funcionalidad aportada por las capas de más bajo nivel.

En la capa de más bajo nivel (nivel 0) opera la máquina real (el hardware del sistema de cómputo) capaz de interpretar y ejecutar instrucciones en el lenguaje máquina, es decir basadas en la lógica digital ("unos y ceros"). A su vez en cierta capa se puede imaginar que opera una máquina virtual (dado que las capas de mayor nivel la ven como una máquina en sí misma aunque no tenga un soporte físico realmente) que solo puede trabajar con programas en su propio lenguaje. Estos programas son interpretados por un intérprete en una máquina virtual de menor nivel o son traducidos al lenguaje de una máquina virtual de menor nivel.

Entonces para que un programa escrito por el usuario en cierto lenguaje de programación sea entendible por la máquina real en el nivel 0, el mismo debe ser traducido a través de las capas a los lenguajes de las máquinas virtuales que operan en cada una de ellas hasta tener su representación en lógica digital.

3. Describa el concepto de trayectoria de datos

El conjunto de la ALU, los registros internos de la CPU y buses internos que interconectan estos componentes constituyen el camino de datos (es la trayectoria a través de la cual se pueden mover los datos).

La ALU así puede operar con los datos ubicados en dos registros (por ejemplo sumar el contenido de los registros o realizar otra operación) y el resultado de la operación almacenarse en otro registro de la CPU. A esto se denomina ciclo del camino de datos. A su

Informática
Guía Teórica
Ingeniería en Mecatrónica

vez este resultado puede ser posteriormente movido a alguna posición de la memoria principal de la computadora.

4. ¿Qué es el microprograma y cuál es su función?

Un microprograma es un conjunto de instrucciones el cual tiene la función de controlar la operación de la trayectoria de los datos.

Informática
Guía Teórica
Ingeniería en Mecatrónica

5. Realice una tabla con las capas de una arquitectura típica, y comente brevemente la función de cada una

| Capas | Descripción |
|---|---|
| Nivel 0: Nivel de lógica digital | Los elementos integrantes de este nivel son las compuertas digitales. Estas compuertas tienen una o más entradas digitales (que aceptan señales de tensión o corriente que representan 1 o 0) y la salida de las mismas es el resultado de una operación sencilla (como OR o AND). A base de compuertas pueden construirse memorias elementales y a base de esas se construyen los registros. A su vez se puede construir la ALU en sí a base de compuertas |
| Nivel 1: Nivel de microarquitectura | La función principal consiste en tomar dos registros internos, hacer que la ALU lleve a cabo una operación con los mismo y almacenar el resultado en otro registro (o sea básicamente operar la trayectoria de datos). |
| Nivel 2: Nivel de arquitectura del conjunto de instrucciones (Nivel ISA) | Su función consiste en interpretar un conjunto de instrucciones para que luego puedan ser ejecutadas por él microprograma o por los circuitos de ejecución en hardware. |
| Nivel 3: Nivel máquina de sistema operativo | Casi todas las instrucciones de este nivel están también en el nivel ISA y cumple más o menos la misma función que este nivel. La diferencia es que cuenta con un nuevo conjunto de instrucciones, diferente organización de memoria, la capacidad de ejecutar más de un programa en simultáneo y otras características. Estas nuevas funciones son interpretadas por el SO en el nivel 2 y aquellas instrucciones compartidas entre nivel 2 y 3 son ejecutadas por el microprograma o por el control por hardware. |
| Nivel 4: Nivel de lenguaje ensamblador | Su función principal es lograr la traducción entre el nivel 4 y los niveles anteriores. El lenguaje de este nivel es una representación simbólica entendible por el usuario de los lenguajes numéricos (incomprensibles para el usuario) de los niveles más bajos. Ofrece así, un método para escribir programas para las máquinas en niveles más bajos. El programa antes de ser interpretado en esos niveles es traducido a través de un programa denominado ensamblador. A través de este nivel se pueden resolver problemas relacionados con el diseño e implementación de máquinas virtuales para máquinas de los niveles inferiores |
| Nivel 5: Nivel de lenguaje orientado hacia problemas | Proporciona lenguajes para ser utilizados directamente por programadores de aplicaciones que requieren resolver problemas específicos. Los programas en este nivel se traducen al nivel 3 o 4 |

Informática
Guía Teórica
Ingeniería en Mecatrónica

| | |
|--|--|
| | mediante compiladores aunque en ocasiones son interpretados. |
|--|--|

6. ¿Qué es la arquitectura de una computadora, y cuáles son los aspectos de los que se ocupa?

La arquitectura de una computadora hace referencia a aquellos atributos que repercuten directamente en la ejecución lógica de los programas y por lo tanto son visibles o importantes para un programador. Por ejemplo, el tamaño de la palabra que maneja el CPU, el set de instrucciones o la memoria disponible son atributos arquitectónicos de una computadora

7. ¿Qué contiene la CPU?

La CPU contiene:

ALU (Unidad Aritmético Lógica): que es la unidad encargada de realizar las operaciones aritméticas (suma, resta, multiplicación y división) y las operaciones lógicas (AND, OR, determinar si un número es mayor a otro, etc).

UC (unidad de control): que es la unidad encargada de decodificar las instrucciones que llegan al CPU y enviar las correspondientes señales de control para llevarlas a cabo (ya sea una operación en la ALU o mover algún dato)

Registros: que son básicamente un conjunto de unidades de memoria de acceso rápido que sirven de almacén temporal de ciertos datos, instrucciones o información de control. Pueden ser de propósito general (los registros de la ALU) o registros especializados como por ejemplo los registros de dirección a memoria, registro de datos, registro de instrucción, contador de programa, etc.

8. Mencione los 3 grandes pasos que realiza la CPU para ejecutar una instrucción, y describa brevemente cada uno

Búsqueda: El PC indica la dirección en memoria de la próxima instrucción que debe ejecutarse. Se lee la posición en memoria indicada para obtener la instrucción y a través del bus de datos se lleva y almacena en el registro de instrucción

Decodificación: Se separa de la instrucción el código de instrucción asociado para determinar el tipo de operación a realizar. La UC decodifica. Además se determina si es necesario colocar una palabra de la memoria principal en algún registro de la CPU, en caso de ser así, la dirección en memoria del dato se almacena en un registro de dirección para buscar la palabra

Ejecución: La UC activa las señales de control correspondientes para ejecutar la instrucción ya sea una operación de la ALU o un movimiento de alguna palabra

Informática
Guía Teórica
Ingeniería en Mecatrónica

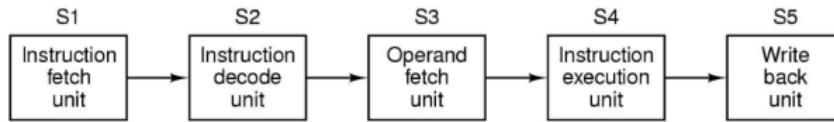
9. ¿Cuál es la ventaja de las CPU RISC sobre CISC?

Dado que las instrucciones implementadas en RISC son simples, la ejecución de las mismas es más rápida en comparación con las instrucciones complejas de CISC. La ejecución de una instrucción en una arquitectura RISC requiere de una menor cantidad de ciclos de reloj mientras que en una arquitectura CISC la ejecución de una instrucción puede requerir de varios ciclos. Esto favorece la segmentación en RISC respecto a CISC. Por otro lado, el código de operación de las instrucciones en RISC es de tamaño fijo mientras que en CISC es variable lo cual implica unidades de decodificación más complejas en CISC. Esta complejidad mayor en CISC también significa una mayor cantidad de transistores en el CPU y por lo tanto un mayor consumo energético. Por esto en los dispositivos de poco consumo (como los smartphones) es habitual el uso de procesadores basados en RISC.

Informática
Guía Teórica
Ingeniería en Mecatrónica

10. Verdadero o Falso: Todas las instrucciones se ejecutan en 1 ciclo de CPU.

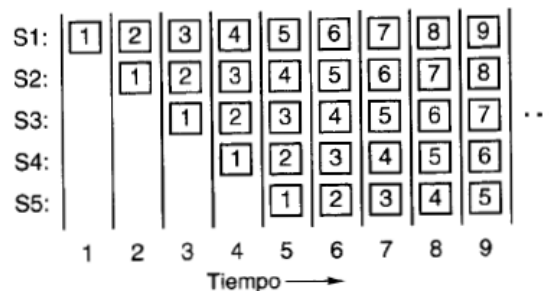
Falso dado que la ejecución de una instrucción requiere de 5 etapas cada una de las cuales implica un ciclo de la CPU como se indica:



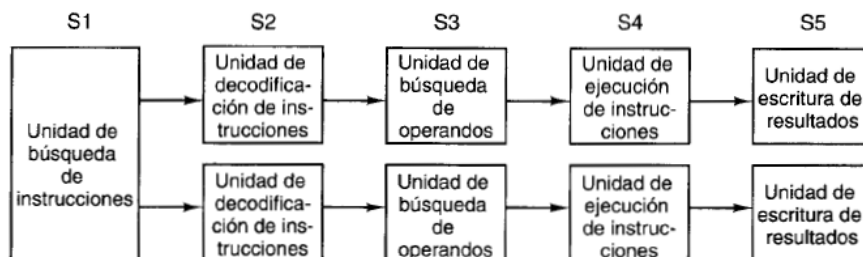
Por otro lado cuando la aseveración del enunciado también es falsa dado que en el caso de una arquitectura RISC, la ejecución de una meta instrucción requiere de varios ciclos de instrucción.

11. ¿Cuál es la diferencia entre el uso de filas de procesamiento y las arquitecturas superescalares?

El concepto de fila de procesamiento (pipeline) consiste en la segmentación en etapas de la ejecución de una instrucción, cada una de las cuales depende de hardware dedicado y que pueden operar de forma paralela. La figura del inciso anterior representa una línea de procesamiento de 5 etapas. De esta manera en cada ciclo de reloj se pueden ejecutar de forma simultánea etapas distintas de instrucciones distintas. Con esto se consigue que en cada ciclo de reloj concluya la ejecución de una instrucción. La siguiente imagen ilustra el concepto



Entendiendo que una fila de procesamiento es buena, para acelerar la ejecución de las instrucciones se pueden utilizar más filas de procesamiento individuales como se indica a continuación.



Para poder implementar este sistema, el par de instrucciones que se ejecutan en simultáneo en filas de procesamiento paralelas han de ser compatibles y no tener interdependencia. Condición que tiene que ser verificada en la compilación del programa o con el agregado de hardware para la detección y corrección de los errores.

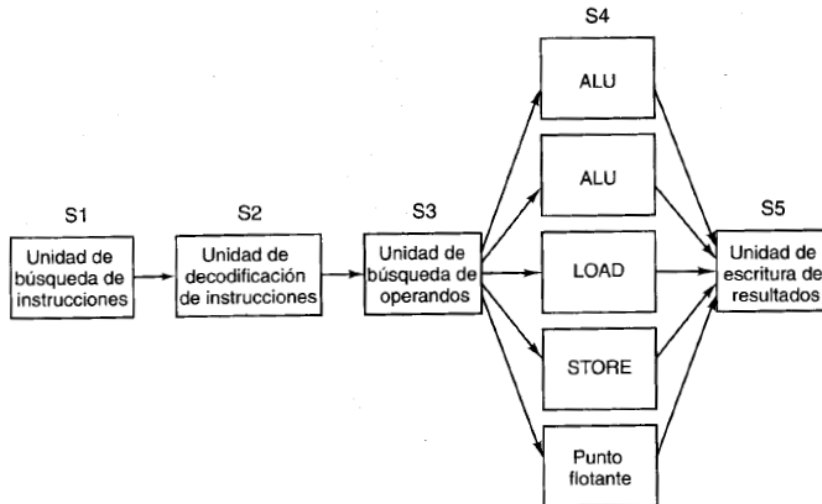
En principio es posible colocar más filas de procesamiento, pero esto aumenta la cantidad

Informática

Guía Teórica

Ingeniería en Mecatrónica

de hardware necesario. En su lugar la estrategia que se implementa es utilizar una sola línea de procesamiento pero a la misma se incorporan varias unidades de ejecución que operan en paralelo como se indica a continuación:



Esta estrategia se basa en la idea de que la tasa de que es mayor la tasa de emisión de instrucción por la etapa S3 es mucho mayor que la tasa de ejecución de las mismas por cada una de las unidades funcionales. Esto es cierto y en general las unidades de la etapa 4 tardan más de un ciclo del reloj en ejecutarse

12. ¿Cuál es la diferencia entre los multiprocesadores y las multicomputadoras?

Un multiprocesador es un sistema de cómputo con varias CPU cada una de las cuales dispone de una memoria local, pero que además dispone de una memoria compartida a la que se conectan a través de un bus común. En cambio las multicomputadoras están constituidas de muchas computadoras individuales que se interconectan a través de conexiones de red y no tienen una memoria compartida

13. Verdadero o Falso: el tamaño de la palabra de memoria es igual para todas las memorias

FALSO

El tamaño de la palabra que maneja una CPU viene dado por el tamaño de sus registros internos. El tamaño de estos registros varía de arquitectura en arquitectura. Así por ejemplo se pueden tener tamaños de palabra de 32 bits o de 64 bits. El tamaño de las unidades de memoria se formatea de acuerdo a este tamaño de palabra.

14. ¿Qué diferencia hay entre el esquema little endian y el esquema big endian?

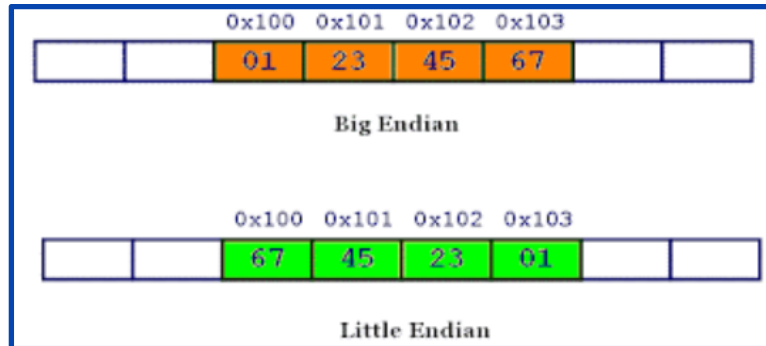
La terminación (endiannes) hace referencia al modo en que se almacenan en memoria los datos que ocupan más de un Byte en memoria.

En la terminación big endian los bytes más significativos se ubican en posiciones en memoria anteriores a las posiciones de los bytes menos significativos mientras en la

Informática
Guía Teórica
Ingeniería en Mecatrónica

terminación little endian los bytes menos significativos ocupan posiciones de memoria anteriores a la de los bytes más significativos de un mismo dato.

La siguiente imagen ilustra el concepto cuando se quiere guardar por ejemplo el dato: **0x01234567**



15. ¿Cuál es la función de la memoria caché?

La caché es una unidad de memoria de acceso más rápido que la memoria principal y que se encuentra más cerca del procesador (en la misma pastilla). Esta unidad almacena de forma especulativa (usando técnicas estadísticas) instrucciones y datos. Es decir que almacena los datos y las instrucciones que son usadas con más frecuencia para que cuando sean requeridas, la CPU las busque en la caché en lugar de buscarla en la memoria principal ya que esta última es más lenta. Si la palabra buscada no está en la caché entonces se busca en memoria. Pero si una gran parte de las palabras requeridas por la CPU se encuentran en la caché el tiempo de acceso medio a los datos se reduce considerablemente.

16. ¿Cómo se conecta la CPU con la memoria principal y dispositivos de E/S?

El CPU se conecta con la memoria y los dispositivos de E/S a través de buses de datos (para transmitir datos), de direcciones (para acceder a ciertas direcciones de memoria) o de control (para transmitir señales de control).

Para la conexión entre la memoria principal y el CPU se utiliza un bus y para la interconexión de los periféricos se emplea otro bus. A su vez se pueden emplear distintos estándares de buses para la conexión de distintos tipos de periféricos. Estos buses se interconectan a través de puentes (bridges).

17. ¿Para qué sirve el registro denominado “Program Counter” (Contador de Programa)?

El PC es un registro que almacena la dirección en memoria de la próxima instrucción que debe ejecutarse.

Informática
Guía Teórica
Ingeniería en Mecatrónica

18. Defina y enuncie las diferencias entre compilación, ensamblado e interpretación.

La compilación es la conversión de código de un lenguaje a otro, en un paso previo a su ejecución.

En la forma más tangible la compilación da un código binario ejecutable como salida y se denomina compilación ahead-of-time (AOT) (que sería antes de tiempo). Este tipo de compilación permite llevar a cabo optimizaciones y adaptaciones complejas a la máquina donde se ejecute el programa. Una vez compilado el programa no es necesario volver a compilarlo a menos que se lleven a cabo modificaciones y se puede ejecutar tantas veces como se quiera. Una de las características que ofrece es la posibilidad de ocultar el código fuente entre diferentes plataformas aunque es más complicado asegurar la compatibilidad

La forma alternativa de ejecutar un programa a partir del código en un lenguaje de programación es analizar el código y realizar los cálculos que éste indique de forma secuencial. En este caso se dice que el programa es interpretado. El programa que lleva a cabo la interpretación se denomina intérprete y debe ejecutarse en el sistema mientras se ejecuta el programa. Esta alternativa no ofrece la posibilidad de ocultar el código fuente pero sí permite la compatibilidad del código entre plataformas que cuenten con el intérprete. Dado que la traducción del código debe hacerse cada vez que se ejecuta el programa y en la medida que es necesario, en este caso la ejecución suele tomar más recursos en tiempo de ejecución y ser más lenta.

Si las instrucciones del lenguaje fuente son esencialmente representaciones simbólicas entendibles por el usuario de las instrucciones en lenguaje numérico de las capas de más bajo nivel, entonces el traductor se denomina ensamblador (como también se denomina al lenguaje fuente del programa) y al proceso de traducción se le denomina ensamblado

Cuando el lenguaje fuente es de alto nivel y el lenguaje objetivo es un lenguaje de máquina numérico o una representación simbólica del mismo el traductor se denomina compilador.

Una de las ventajas del lenguaje ensamblador es que permite el acceso a ciertos recursos de la máquina objetivo (recursos de hardware) que no son accesibles a través de los lenguajes de más alto nivel. Sin embargo todas las características que ofrece un lenguaje de alto nivel pueden obtenerse también a través de un programa escrito en ensamblador.

Una desventaja del ensamblador respecto de los lenguajes de alto nivel es la falta de compatibilidad a través de distintas plataformas. Las instrucciones del lenguaje ensamblador, al ser representaciones simbólicas de las instrucciones de las máquinas virtuales de más bajo nivel (y la máquina real), están en estrecha relación con la propia arquitectura de una familia de computadoras en específico.

Informática
Guía Teórica
Ingeniería en Mecatrónica

19. Indique el valor de los siguientes números en sistema decimal, hexadecimal y octal:

| | | | |
|----|----------------------------|----------------------------|--|
| | 01000101 00100101 11001001 | 11010011 11000100 10001010 | |
| 16 | 45 25 C9 | D3 C4 8A | |
| 8 | 105 045 311 | 323 304 212 | |
| 10 | 4531657 | 13878000 | |

20. Indique el valor de los siguientes números en sistema binario, hexadecimal y octal

| | | | |
|----|-------------------|-------------------|--|
| | 7225 | 6234 | |
| 2 | 00011100 00111001 | 00011000 01011010 | |
| 8 | 016071 | 014132 | |
| 16 | 1C39 | 185A | |

21. ¿Cómo representan las computadoras los números con punto flotante?

La representación de números con punto flotante se hace bajo el estándar IEEE 754. Este define 8 formatos de representación, 5 básicos y 3 especiales. Dentro de los formatos básicos vamos a referirnos a binary32 (simple precisión) y binary64 (doble precisión) que usan respectivamente 32 bits y 64 bits para la representación.

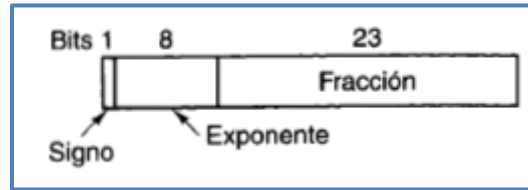
De forma general el formato consta de las siguientes partes de izquierda a derecha de la palabra:

- El bit de extrema izquierda es un bit de signo. Para números negativos se utiliza el 1 y para números positivos se utiliza el 0
- Un conjunto de números para representar el exponente en exceso.
 - Para binary32: Se utilizan 8 bits para representar el exponente en exceso a 127 ($2^7 - 1$). Es decir que se pueden tener exponentes desde -127 a 128 que se representan con números desde 0 (para -127) a 255 (para 128) en base decimal.
 - Para binary64: Se utilizan 11 bits para representar el exponente en exceso a 1023 ($2^{10} - 1$). Es decir que se pueden tener exponentes desde -1023 a 1024 que se representan con números desde 0 (para el -1023) a 2047 (para 1024) en base decimal.
- La última parte de la palabra para representar la fracción o mantisa.
 - Para binary32: Una fracción de 23 bits
 - Para binary64: Una fracción de 52 bits

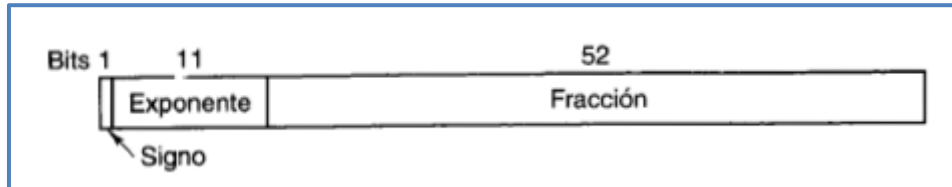
Informática

Guía Teórica

Ingeniería en Mecatrónica



Representación en coma flotante para binary32



Representación en coma flotante para binary64

Números normalizados:

Son de la forma $1. \text{fracción} \times 2^{\text{exponente}}$

El conjunto formado por $1. \text{fracción}$ se denomina **significando**. El **1 antes del punto binario y el punto binario son implícitos**, es decir que no se representan de forma explícita en los bits de la fracción.

El **exponente de la fracción de un número normalizado no puede ser el mínimo** (0 binario de 8 o 11 bits que representa al -127 para binary32 y -1023 para binary64 respectivamente) **ni el máximo** (255 o 2047 correspondientes a 128 o 1024 para binary32 o binary64 respectivamente) que se reservan para otros casos especiales

Ejemplo

En **binary32** el número **2,3125** tiene una representación dada en hexadecimal por **40140000**

Números desnormalizados:

Son de la forma $0. \text{fracción} \times 2^{-127}$

El **0 antes del punto binario y el punto binario son implícitos**. El exponente para los números desnormalizados es el cero binario de 8 bits para binary32 o el 0 de 11 bits para binary64.

Este formato permite trabajar el problema del subdesbordamiento de los números normalizados reduciendo gradualmente la significancia en lugar de redondear cualquier resultado menor a 1.0×2^{-127} (menor número normalizado representable) a cero.

Además de estos existen otros formatos especiales. Todos los formatos se resumen en el cuadro siguiente

Informática

Guía Teórica

Ingeniería en Mecatrónica

| | | | |
|----------------|---|-------------------------------|---|
| Normalizado | ± | $0 < \text{Exp} < \text{Máx}$ | Cualquier patrón de bits |
| Desnormalizado | ± | 0 | Cualquier patrón de bits distinto de cero |
| Cero | ± | 0 | 0 |
| Infinito | ± | 1 1 1...1 | 0 |
| Ningún número | ± | 1 1 1...1 | Cualquier patrón de bits distinto de cero |

↙ Bit de signo

Tanto la representación del infinito como la representación NaN se pueden utilizar como operadores en operaciones dando resultados predecibles.

22. ¿Qué es el desbordamiento y el subdesbordamiento en números de punto flotante?

El **desbordamiento** ocurre cuando el resultado de una operación tiene un valor absoluto mayor al máximo valor absoluto de un número representable en el formato dado. Esto vale tanto para los números negativos como para los números positivos.

Por ejemplo, en el formato binary32 la máxima magnitud de un número normalizado representable es la que corresponde a un exponente con los primeros 7 bits a la izquierda a 1 y el último a 0 (corresponde a un exceso de 254 que representa al 127) y una fracción con todos los bits a 1. Esto es una magnitud de aproximadamente 1.9999×2^{127} que es casi 2^{128} . Si el resultado de una operación tiene un valor absoluto mayor a este, ocurre desbordamiento y se maneja con la representación del infinito en IEEE 754.

El **sub-desbordamiento** ocurre cuando el resultado de una operación es un número cuyo valor absoluto es menor al mínimo valor absoluto de un número representable en el formato dado. Esto vale tanto para los números negativos como para los números positivos.

Por ejemplo, en el formato binary32 la mínima magnitud de un número normalizado representable es la que corresponde a un exponente con los primeros 7 bits a la izquierda a 0 y el último a 1 (corresponde a un exceso de 1 que representa al -126) y una fracción con todos los bits a 0. Esto es una magnitud de 1.0×2^{-126} . Si el resultado de una operación es un número con un valor absoluto menor a este, ocurre sub-desbordamiento de la representación en números normalizados.

Para manejar el sub-desbordamiento en IEEE 754 se utiliza el formato de los números des-normalizados. En este formato se puede representar una máxima magnitud de aproximadamente $0,999 \times 2^{-127}$ que es casi la mínima magnitud para los números normalizados y una mínima magnitud de 2^{-150} .

Informática
Guía Teórica
Ingeniería en Mecatrónica

23. Verdadero o Falso: el error de redondeo absoluto en punto flotante es igual para números pequeños y para números grandes.

Falso. Por ejemplo el error absoluto al redondear un número comprendido entre el máximo número normalizado positivo representable en binary32 y el anterior, que dados en hexadecimal son respectivamente **7F7FFFF** y **7F7FFFE** es de alrededor de 2^{103} mientras que el error absoluto de redondear un número comprendido entre el número normalizado positivo más pequeño con la **misma fracción** que **7F7FFFE** y el siguiente, que dados en hexadecimal son respectivamente **00FFFFFFE** y **00FFFFFFF** es de alrededor de 2^{-150} . Evidentemente el error absoluto cometido en el segundo caso es infinitamente menor que el error en el primer caso.

24. Verdadero o Falso: el error de redondeo relativo en punto flotante es menor para números pequeños y para números grandes.

Falso. A diferencia del error absoluto, el error relativo que se comete tanto para números grandes como para números pequeños es el mismo. Tomando los mismos ejemplos que en el inciso anterior podemos ver esto tomando la distancia entre los números consecutivos relativa a la magnitud del número más pequeño. Para **7F7FFFF** y **7F7FFFE** es de:

$$\frac{2^{104}}{2^{127} * fraccion} = \frac{2^{-23}}{fraccion}$$

Donde *fracción* es la de **7F7FFFE**

Mientras que para **00FFFFFFE** y **00FFFFFFF** la distancia relativa es de:

$$\frac{2^{-149}}{2^{-126} * fraccion} = \frac{2^{-23}}{fraccion}$$

Donde *fracción* es la de **00FFFFFFE** que es la misma que la de **7F7FFFE**

Informática
Guía Teórica
Ingeniería en Mecatrónica

Sistemas Operativos

25. ¿Cuáles son las 2 principales funciones del Sistema Operativo (SO)? Explíquelas brevemente y ejemplifique.

Las funciones básicas del SO son:

- **Proveer una máquina extendida:** Esto significa proporcionar a los programadores de aplicaciones y a las aplicaciones en sí de un conjunto de recursos que son abstracciones de los complejos recursos de hardware.

El funcionamiento como máquina extendida permite a los programadores de aplicaciones crear aplicaciones con independencia o abstracción del hardware que subyace. Por ejemplo, la mayoría de los editores de texto o editores de imagen ofrecen la opción de imprimir el documento. Sin embargo en el momento de programar está opción en la aplicación, el programador no tuvo que preocuparse por el tipo de impresora que lleve a cabo la operación o el modo de funcionamiento de la misma, etc. En su lugar el SO ofrece al programador de un medio de acceso simplificado al recurso independientemente del funcionamiento del mismo, dado que el SO "sabe" comunicarse con él o como acceder al mismo. Por esta razón la aplicación funcionaría igual de bien en distintos equipos que utilicen distintas impresoras en tanto al menos el SO sea el mismo. Por esto se dice que las aplicaciones se programan para el SO y no para una arquitectura en específico. Otro ejemplo de abstracción aportada por el SO es el sistema de archivos, que permite definir y acceder a distintas partes de un medio físico de almacenamiento como unidades identificables (un archivo).

- **Administración de los recursos de cómputo:** El objetivo es proporcionar una asignación controlada y organizada de los distintos recursos del sistema de cómputo ya sean de hardware u otros a las distintas aplicaciones o usuarios que compiten por ellos. Para lograr esto, el SO debe funcionar como mediador de las peticiones de recurso en conflicto que provienen ya sea de varios usuarios o de varios programas y llevar registro del uso de los recursos.

Una de las acciones que intervienen en la administración de recursos es el multiplexaje tanto en el tiempo como en el espacio. El multiplexaje en el tiempo es en esencia el acceso a un mismo recurso solicitado por varios usuarios o programas en turnos. Es el SO el encargado de determinar la forma en que se lleve a cabo el multiplexaje en el tiempo (por ejemplo determinar el tiempo de acceso al CPU por parte de cada programa y la secuencia en la que accedan). El mutiplexaje en el espacio es la asignación simultánea a varios usuarios o programas de porciones de un mismo recurso. Es el SO quien debe determinar la forma de llevar a cabo el multipleaje en el espacio y de resolver los conflictos que se presenten (por ejemplo en una tarea del SO es la de asignar espacio en disco a archivos de distintos usuarios y de llevar un registro de los bloques que están siendo utilizados y por quién).

Informática

Guía Teórica

Ingeniería en Mecatrónica

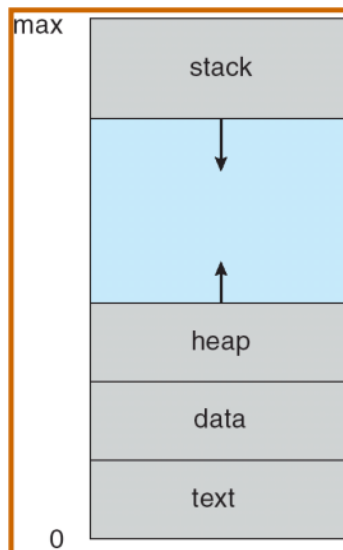
26. ¿Qué es una llamada al sistema? ¿Para qué sirve? Ejemplifique.

Una llamada al sistema es el método que tiene un programa de usuario que se ejecuta en modo usuario de acceder a los servicios que ofrece el kernel del SO. Las llamadas al sistema permiten preservar una separación estricta entre los programas de aplicación y el kernel del SO con motivos seguridad y estabilidad del sistema al no permitir acceso directo por ejemplo al CPU o a la memoria del sistema a los programas de aplicación. **En resumen, una llamada al sistema es requerida siempre que un programa de aplicación quiera acceder a un recurso del sistema de cómputo al que solo puede accederse en modo kernel.**

Por ejemplo siempre que un programa de aplicación quiere acceder a una operación típica con archivos como por ejemplo la lectura, escritura, cierre, creación o eliminación debe hacerlo a través de llamadas al sistema. Un ejemplo de llamada al sistema es la llamada **read** de los sistemas operativos bajo la norma POSIX para la lectura de archivos.

27. ¿Qué es un proceso?

Esencialmente un proceso es un programa en ejecución. A un proceso hay asociado un espacio de direcciones que es un conjunto de direcciones de memoria que va de un mínimo a un máximo en donde puede escribir y leer información. En el espacio de direcciones se encuentra el propio programa ejecutable con los datos del mismo y su pila. Al proceso hay asociado además un conjunto de recursos como registros, una lista de archivos abiertos, alarmas pendiente, lista de procesos relacionados e información necesaria para el programa



NOTA: Esta es una representación esquemática del espacio de direcciones de un proceso.

Stack hace referencia a la pila, en donde se almacenan datos temporales, heap es la memoria dinámica del proceso, data son los datos globales del proceso y text es el conjunto de instrucciones del proceso

Informática
Guía Teórica
Ingeniería en Mecatrónica

28. Verdadero o Falso:

Un proceso tiene asociado un único programa

Verdadero. A cada proceso corresponde un único programa. Cuando un programa se ejecuta por duplicado cuenta como dos procesos distintos

Un programa puede tener asociado un único proceso

Falso. Como se indicó, cuando un programa se ejecuta por duplicado se crean dos procesos distintos. Entonces a un programa pueden corresponder más de un proceso

29. Defina

1. Directorio

Directorio es la forma que tiene el SO de agrupar los archivos. En un directorio puede haber no solo archivos sino que también otros directorios lo que da origen a una jerarquía y estructura de árbol común a todos los sistemas de archivos

2. Ruta de acceso (path)

Es la forma de referenciar de forma exacta la ubicación de un archivo en la estructura jerárquica. Esta ruta de acceso indica la serie de directorios que tienen que recorrerse para llegar al directorio del archivo y al archivo en sí. Se dice que es una ruta absoluta cuando comienza desde el directorio raíz, en cambio es relativa si comienza desde un directorio que no es la raíz. El formato de la ruta depende del sistema de archivos y del SO

3. Directorio de trabajo

El directorio de trabajo de un proceso es un directorio asociado dinámicamente con cada proceso. Cuando el proceso se refiere a un archivo por un nombre simple o por una ruta relativa, la misma es interpretada en relación con este directorio de trabajo. Entonces se podría decir que es el directorio que contiene los directorios o los archivos a los que accede el proceso en determinada instancia del mismo.

30. ¿Qué son los bits rwx? ¿Para qué sirven?

En el sistema operativo UNIX los permisos de acceso de los distintos usuarios a un archivo (protección del archivo) se especifican con un conjunto de 9 bits divididos en 3 campos de 3 bits. El primer campo especifica los permisos del propietario, el segundo campo el de los demás miembros en el mismo grupo del propietario y el último los de otros usuarios. En cada campo el primer bit (el izquierdo) indica el permiso de lectura (**es el bit r**), el segundo bit (el intermedio) indica el permiso de escritura (**es el bit w**) y el tercero indica el permiso de ejecución (**es el bit x**). El bit r a 1 indica que se tiene el permiso de lectura y de la misma manera para los bit w y x.

Informática
Guía Teórica
Ingeniería en Mecatrónica

31. ¿Cuál es la diferencia entre los siguientes?

1. un archivo “regular”

Los archivos regulares son los archivos más comunes y se utilizan para contener datos. Los archivos consisten en una secuencia de bytes que se escriben en un dispositivo de E/S

2. un archivo “especial”

En UNIX un archivo especial es un archivo que permite tratar a los dispositivos de E/S como archivos y que de esta manera se pueda leer desde los mismos o escribir a ellos las mismas llamadas utilizadas para leer o escribir archivos. Hay de dos tipos, **los de bloque** permiten modelar dispositivos a los que se puede acceder en bloques direccionables al azar. De esta manera al acceder a un bloque específico del archivo se puede acceder al bloque físico de la unidad. **Los de carácter** permiten modelar dispositivos que entregan o reciben un flujo de datos como impresoras o modems

3. una tubería (pipe)

Un **canal (pipe)** es un tipo de pseudoarchivo que se utiliza para interconectar procesos. Si dos procesos *A* y *B* desean comunicarse mediante el uso de un canal, deben establecerlo por adelantado. Cuando el proceso *A* desea enviar datos al proceso *B*, escribe en el canal como si fuera un archivo de salida. El proceso *B* puede leer los datos a través del canal, como si fuera un archivo de entrada. Por ende, la comunicación entre procesos en UNIX tiene una apariencia muy similar a las operaciones comunes de lectura y escritura en los archivos.

32. Describa cómo se implementa la multiprogramación (multiprocessing)

La multiprogramación consiste en la ejecución de forma concurrente de más de un programa con el objetivo de minimizar los tiempos que el CPU se encuentra ocioso. De esta manera, en lugar de esperar a que una operación de lectura de una unidad de E/S de un determinado proceso finalice (lo cual puede corresponder a varios ciclos de reloj) se ocupa al procesador con las instancias de otro programa en ejecución (ejecuta el próximo proceso en la cola de procesos). Es decir que la multiprogramación se logra haciendo que el procesador conmute entre distintos procesos de forma rápida. De esta forma, al cabo de cierto intervalo de tiempo (corto para los tiempos que el usuario suele manejar) el procesador ha logrado ejecutar distintos porcentajes (o procesos completos) de distintos procesos dando la apariencia de que ejecuta de forma simultánea varios procesos aunque en la realidad en cada instante el procesador solo ejecuta un proceso.

33. Indique el estado en que se encuentra un proceso en cada caso:

1. El proceso tiene todo lo que necesita para correr, pero no es su turno de utilizar la CPU

Listo

Informática
Guía Teórica
Ingeniería en Mecatrónica

2. El proceso está esperando datos por la red y no puede continuar

Bloqueado

3. El proceso recibió los datos de red que estaba esperando

Listo

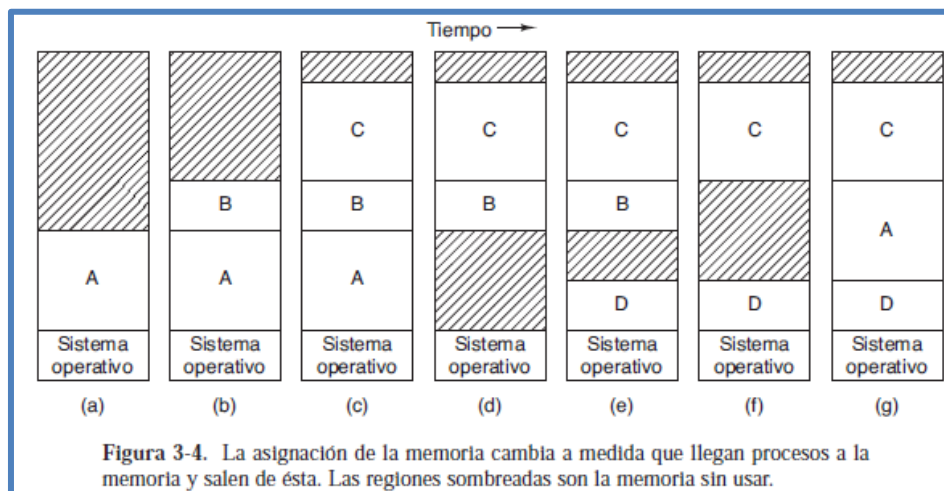
34. ¿Qué son las secciones críticas de un proceso?

Las secciones críticas de un proceso son las partes del mismo que hacen acceso a recursos compartidos por varios procesos, como por ejemplo la memoria principal. Las secciones críticas de distintos procesos deben ejecutarse de forma mutuamente excluyente para evitar condiciones de carrera. Estas condiciones se dan cuando dos procesos hacen uso de los mismos recursos compartidos y el resultado depende de cuál se ejecute primero y de cuándo lo haga.

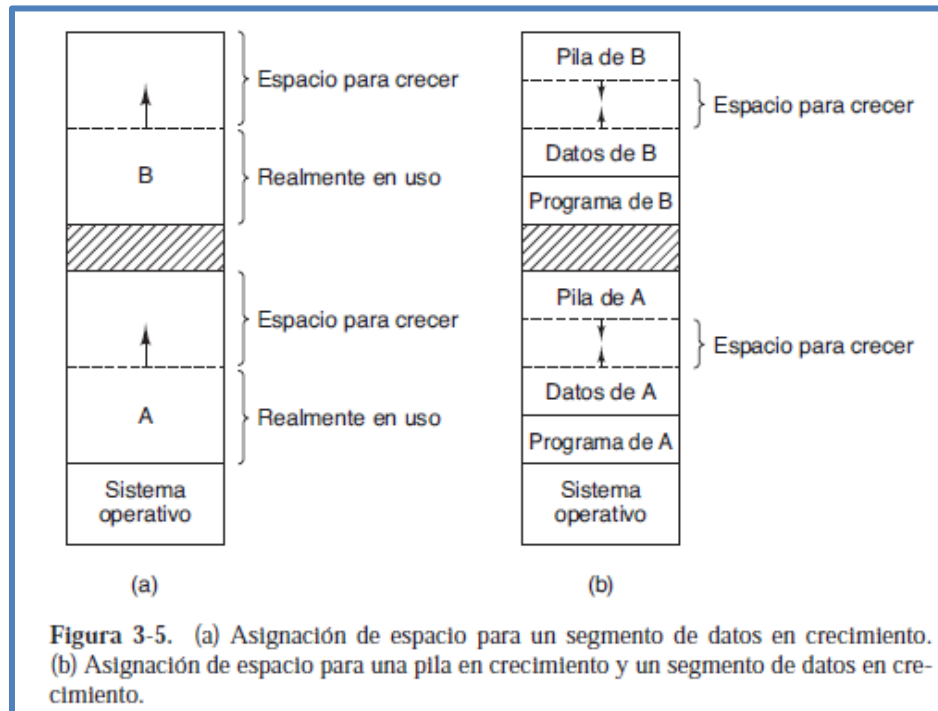
35. ¿En qué consiste la técnica de gestión de la memoria denominada intercambio (swapping)?

Es un mecanismo para mover programas entre memoria principal y secundaria, normalmente disco. Con swapping, los programas pueden salir/entrar de/a memoria durante su tiempo de ejecución. Normalmente, un programa abandona la memoria para dejar espacio a otro.

La asignación de memoria a los procesos puede hacerse en tamaños fijos, en cuyo caso hay que llevar a cabo movimientos de los procesos en la propia memoria o intercambios con la memoria secundaria cuando el segmento de datos de un proceso ha de crecer para colocar al mismo al lado de un hueco libre en la memoria si es que lo hay (lo cual es ineficiente). Por otro lado, si el crecimiento del segmento de datos de un proceso es previsto, es posible hacer una asignación de memoria dinámica agregando un espacio adicional.



Informática
Guía Teórica
Ingeniería en Mecatrónica



36. Mencione y describa brevemente 2 técnicas de administración de la memoria.

- **Paginación:** Los procesos se dividen en trozos pequeños denominados páginas todos del mismo tamaño y también la memoria principal se divide en trozos del mismo tamaño que se denominan marcos de página. Después las páginas de un programa se pueden asignar a los marcos de página. De esta manera, el espacio de memoria desperdiciado por un proceso es, como mucho, una fracción de la última página.
- **Segmentación:** La segmentación permite que el programador vea la memoria constituida por múltiples espacios de direcciones o segmentos. Los segmentos tienen un tamaño variable, dinámico. Usualmente, el programador o el sistema operativo asignarán programas y datos a segmentos distintos. Las referencias a memoria se realizan mediante direcciones constituidas por un número de segmento y un desplazamiento.

37. ¿Para qué sirve la paginación?

La paginación sirve principalmente para poder administrar eficientemente la memoria disponible minimizando los tiempos muertos del procesador y teniendo en cuenta el hecho de que en la memoria no pueden entrar de forma simultánea todos los programas que deben ejecutarse "en simultáneo". A diferencia de la segmentación, en este caso el tamaño de las páginas es fijo y determinado por el SO y por lo tanto se dice que este método es transparente al programador

Informática
Guía Teórica
Ingeniería en Mecatrónica

38. ¿Para qué sirve la segmentación?

La segmentación, al igual que la paginación, sirve para efectuar una eficiente administración de la memoria en un sistema multiprogramado y evitar que el procesador permanezca inactivo por falta de memoria. A diferencia de la paginación, la segmentación es usualmente visible para el programador y proporciona una forma conveniente de organizar los programas y los datos, para asociar los privilegios y los atributos de protección con las instrucciones y los datos

39. Mencione las operaciones básicas que pueden realizarse sobre los archivos

Las operaciones básicas que se pueden realizar sobre un archivo son:

- Buscarlo y localizarlo junto con la información necesaria para acceder a él
- Crearlo
- Abrirlo
- Escribirlo
- Leerlo
- Cerrarlo
- Eliminarlo de la memoria
- Cambiarle el nombre
- Modificar los permisos del mismo

40. ¿Para qué sirve la estructura de directorios?

La estructura de directorios permite una organización jerárquica eficiente de los archivos. Esto permite el acceso de forma cómoda a los archivos y también compartir archivos de forma sencilla entre miembros de un mismo grupo a con dispositivos de E/S

41. ¿Cuáles son los objetivos del software de E/S a nivel de Sistema Operativo?

- Independencia de dispositivos: Generar abstracción al respecto de los dispositivos de E/S a los que los programas van a acceder. Es decir que las características relativas al tipo o funcionamiento de los mismos es transparente a los programas y a los programadores.
- Denominación uniforme: Organizar los archivos de tal forma de no generar conflictos y con independencia del tipo de dispositivos.
- Manejo de errores: Solo si los niveles inferiores no pueden lidiar con el problema, los niveles superiores se deben encargar de resolverlo.
- Administrar las transferencias síncronas (de bloqueo) contra las asíncronas (controladas por interrupciones):
- Uso de búfer: Los datos que provienen de un dispositivo que no se pueden almacenar directamente en su destino final se colocan en un buffer.

Informática
Guía Teórica
Ingeniería en Mecatrónica

42. Indique las 4 capas en las que se estructura el software de E/S en un Sistema Operativo, y mencione brevemente la función de cada una.

- **Manejadores de interruptores:** Entre sus principales funciones se encuentran:
Guardar y cargar los registros
Organizar, reconocer y ejecutar el procedimiento de servicio de interrupciones
Organizar y administrar ciertas instrucciones para el CPU relacionadas con el uso de memoria.
- **Controladores de dispositivos:** Un controlador tiene ciertos registros de dispositivos que se utilizan para darle comandos al dispositivo, para leer su estado, o ambas. Cada dispositivo que se conecta a nuestra computadora necesita de al menos un driver para poder ser controlado. Esta capa del sistema operativo tiene la función de administrar estos drivers.
Entre las funciones de administración se encuentran: Administrar peticiones abstractas de lectura y escritura del software, administrar requerimientos y eventos, administrar la transferencia de datos para evitar daños en la estructura de datos, inicializar el dispositivo, buscar y reportar errores.
- **Software de sistema operativo independiente del dispositivo:** La función básica del software independiente del dispositivo es realizar las funciones de E/S que son comunes para todos los dispositivos y proveer una interfaz uniforme para el software a nivel de usuario. Entre estas funciones comunes se encuentran:
- **Generar una interfaz uniforme para controladores de dispositivos:** No es conveniente tener que modificar el sistema operativo para cada nuevo dispositivo. Por lo tanto esta capa se encarga de administrar un conjunto de funciones comunes que el controlador debe proporcionar para la gran mayoría de dispositivos.
Administrar el uso de búfer: Administra estrategias para aprovechar el máximo rendimiento del uso del búfer.
- **Asignar y liberar dispositivos dedicados:** Algunos dispositivos sólo pueden ser utilizados por un solo proceso en un momento dado. Es responsabilidad del sistema operativo examinar las peticiones de uso de los dispositivos y aceptarlas o rechazarlas.
- **Proporcionar un tamaño de bloque independiente del dispositivo:** Los distintos discos de almacenamiento pueden tener diferentes tamaños de sectores. Es responsabilidad del software independiente del dispositivo ocultar este hecho y proporcionar un tamaño de bloque uniforme a los niveles superiores.
- **Software de E/S de capa de usuario:** Aunque la mayor parte del software de E/S está dentro del sistema operativo, una pequeña porción de éste consiste en bibliotecas (conjuntos de archivos asociados a memoria) vinculadas entre sí con programas de usuario. Su función es administrar los procedimientos que involucran operaciones de E/S y que se ejecutan como parte de los programas de usuario. También administra el uso de colas para lidiar con los dispositivos de E/S dedicados en un sistema de multiprogramación.