

Niveles de descripción del computador

Escuela Superior de Ingeniería y Tecnología
Universidad de La Laguna

Abril, 2016

Esquema de la lección

Introducción

El nivel de
lógica digital

El nivel de la
microprogra-
mación

Nivel máquina
convencional

El nivel del
lenguaje
ensamblador

Lenguajes de
alto nivel

Transitando
los niveles de
descripción:
compiladores e
intérpretes

- 1 Introducción
- 2 El nivel de lógica digital
- 3 El nivel de la microprogramación
- 4 Nivel máquina convencional
- 5 El nivel del lenguaje ensamblador
- 6 Lenguajes de alto nivel
- 7 Transitando los niveles de descripción: compiladores e intérpretes

Introducción (I)

Introducción

El nivel de
lógica digital

El nivel de la
microprogra-
mación

Nivel máquina
convencional

El nivel del
lenguaje
ensamblador

Lenguajes de
alto nivel

Transitando
los niveles de
descripción:
compiladores e
intérpretes

- En esta lección trataremos sobre conceptos y herramientas directamente relacionados con los niveles de descripción del computador.
- **Nivel de descripción:** Todo objeto admite diferentes niveles de descripción. Por ejemplo: podemos describir un libro de diferentes modos:
 - Un libro con portada de color azul, de 20cm por 17cm por 6 cm y que pesa 340 gramos. La primera página está vacía. La segunda página también. En la tercer hay una letra “G” en la posición de tamaño ...
 - El libro se titula “Guía del viajero de las Islas Canarias”. Tiene un Prólogo y 7 capítulos. El prólogo se titula “Prólogo” y contiene el texto El primer capítulo se titula “La Gomera” y contiene el texto ...

Introducción (II)

Introducción

El nivel de
lógica digital

El nivel de la
microprogra-
mación

Nivel máquina
convencional

El nivel del
lenguaje
ensamblador

Lenguajes de
alto nivel

Transitando
los niveles de
descripción:
compiladores e
intérpretes

- Ambas descripciones permite a un usuario leer el libro si usa la información para reconstruirlo. **Pero en realidad no son equivalentes.**
- La primera es una descripción en un nivel más bajo que la segunda. Se dice que es de más bajo nivel por su mayor grado de concreción.
- Es decir, mientras de la descripción de mayor nivel pueden derivarse diferentes formas de la de menor nivel, de la especificación exacta de todos los aspectos del libro solo puede derivarse un texto.

En informática:

Una clase es una descripción de más alto nivel que un objeto (que ya tiene fijados los parámetros).

Los lenguajes basados en objetos son de más alto nivel que los programas imperativos (basados en procedimientos)

Introducción

El nivel de
lógica digital

El nivel de la
microprogra-
mación

Nivel máquina
convencional

El nivel del
lenguaje
ensamblador

Lenguajes de
alto nivel

Transitando
los niveles de
descripción:
compiladores e
intérpretes

Introducción (III)

- La materialización del libro requiere del mayor grado de concreción posible. Al final, una descripción lo más exacta posible del objeto físico requiere un alto nivel de concreción.
- Sin embargo, la cantidad de información que requiere esta descripción es enorme. La descripción de alto nivel requiere mucha menos información y aunque por sí sola no permite la reproducción exacta del objeto físico, sí permite generar otros con la misma utilidad.
- Por otra parte, esta separación en niveles permite generar diferentes productos físicos, con características operativas adecuadas para cada situación: por ejemplo, un libro de bolsillo, un libro de tapa dura en una lujosa edición o un libro electrónico.

Introducción (IV)

Introducción

El nivel de
lógica digital

El nivel de la
microprogra-
mación

Nivel máquina
convencional

El nivel del
lenguaje
ensamblador

Lenguajes de
alto nivel

Transitando
los niveles de
descripción:
compiladores e
intérpretes

- En esta lección no hablaremos de libros, sino de la ejecución de programas informáticos, es decir de la **actividad que realiza un computador.**
- Esta actividad admite diferentes descripciones: algunas serán mucho más concretas y requerirán mayor cantidad de información para expresarlas y otras serán mucho más compactas.
- Así pues, cuando hablamos de los diferentes niveles de descripción del computador nos referimos a **diferentes tipos de descripciones de las posibles ejecuciones de un programa en un computador.**

El nivel de lógica digital (I)

Introducción

El nivel de lógica digital

El nivel de la microprogra- mación

Nivel máquina convencional

El nivel del lenguaje ensamblador

Lenguajes de alto nivel

Transitando los niveles de descripción: compiladores e intérpretes

- Comenzaremos por un nivel de descripción con un alto grado de concreción. Nos referimos aquí a la especificación del computador en términos de su lógica digital y de la ejecución del programa en términos de los estados que estos circuitos lógicos transitan.
- En la tecnología actual estos estados se describen como asignaciones de 1 y 0 a variables binarias.
- Las posibles transiciones entre estados están acotadas por las reglas de los circuitos digitales.

El nivel de lógica digital (II)

Introducción

El nivel de lógica digital

El nivel de la microprogra- mación

Nivel máquina convencional

El nivel del lenguaje ensamblador

Lenguajes de alto nivel

Transitando los niveles de descripción: compiladores e intérpretes

- Así pues, conociendo el estado inicial de las variables digitales y los circuitos digitales involucrados podemos describir las posibles ejecuciones del programa.
- Observen que en este nivel el concepto de programa o instrucción del programa se diluye.
- Si a esta descripción añadimos la de los dispositivos electrónicos incluidos en el computador obtenemos un nivel de concreción aún mayor.
- La cantidad de información necesaria para recrear la actividad de un programa de complejidad media en este nivel es enorme, lo que hace necesario un diseño especial de los sistemas digitales de modo que sea posible un tipo de estado más abstracto y unas reglas de transición más sencillas.

El nivel de microprogramación

- En el computador actual se crean subsistemas digitales con diferentes responsabilidades. La funcionalidad de estos subsistemas se puede describir en términos de entradas, estados internos y salidas.
- Estos elementos admiten un número de posibles valores discretos relacionados por funciones. Estos valores discretos ya no son bits individuales y tienen un significado relacionado con la funcionalidad del subsistema.
- El conjunto de subsistemas es capaz de ejecutar microinstrucciones pertenecientes a un repertorio.
- No debemos confundir estas microinstrucciones con las instrucciones del procesador. Una microinstrucción representa una orden a uno o varios subsistemas digitales concretos.
- Por ejemplo, una microinstrucción puede consistir simplemente en activar una conexión entre dos registros compuestos por elementos de memoria digital.

El nivel máquina convencional (I)

Nivel de lenguaje máquina

- Al igual que hicimos con los sistemas digitales vamos a agrupar microinstrucciones para construir funciones más complejas: tenemos así **microprogramas**.
- Estos microprogramas en realidad son la especificación de una **instrucción del procesador**.
- Cuando se diseña un procesador, uno de los elementos fundamentales es su repertorio de instrucciones.

El nivel máquina convencional (II)

- El repertorio establece qué debe hacer la instrucción y cómo debe codificarse esta.
- La funcionalidad se implementa mediante microprogramas. Por tanto un mismo repertorio de instrucciones admite diferentes implementaciones físicas (diferentes microprogramas, con diferentes microinstrucciones y circuitería asociada).
- **Sin embargo, aunque la implementación de las instrucciones cambie, los programas del computador podrán seguir ejecutándose.**
- Esta es una idea muy importante, con grandes repercusiones en el abaratamiento de costes, la reutilización de la tecnología y la escalabilidad de los sistemas.

Nivel del lenguaje ensamblador (I)

- Las instrucciones del procesador se codifican mediante cadenas binarias cuya instrucción viene especificada por el fabricante.
- El programador necesita otro tipo de codificación y en algunos casos nuevas funcionalidades que hagan menos tediosa y compleja la elaboración de un programa.
- La solución pasa por un lenguaje que permita expresar órdenes similares a las instrucciones del procesador en texto. Es el **lenguaje ensamblador**

Nivel del lenguaje ensamblador (II)

- No se puede comprender el lenguaje ensamblador sin un conjunto de programas denominado “toolchain”
- La misión de una “toolchain” es construir una imagen binaria del programa en ensamblador que se pueda cargar en la memoria del computador para ser ejecutada.
- Un elemento fundamental de la “toolchain” es el denominado “compilador”.
- En el caso concreto del ensamblador, el compilador realiza las siguientes funciones:
 - Traduce las instrucciones ensamblador a instrucciones máquina. (Ensamblador)
 - Utiliza las **directivas** escritas en el programa ensamblador para organizar la imagen binaria (área de datos, programa, alineación, ...). (Ensamblador) + (Linker)
 - Procesa adecuadamente las referencias a memoria (saltos, bifurcaciones, variables) para que estas sean traducidas a direcciones de memoria en la imagen binaria o en el proceso de carga de la misma. (Linker)

Problemática del lenguaje ensamblador

- Aunque la funcionalidad ofrecida por las instrucciones del lenguaje ensamblador es un salto cualitativo importantísimo respecto al nivel de lógica digital, aún representa un nivel de concreción importante, lo que ocasiona varios problemas:
 - Se requiere un trabajo importante para especificar funciones más complejas, como tratar con estructuras de datos o realizar operaciones de entrada / salida con los dispositivos.
 - Por la misma razón, la construcción de programas sin fallos o que reaccionen adecuadamente ante posibles errores, es difícil.
 - Finalmente, cada programa estaría escrito para un repertorio de instrucciones concreto. La migración a otro repertorio supone un trabajo ingente.

Lenguajes de alto nivel (I)

- Los denominados lenguajes de alto nivel representan una solución a estos problema.
- Estos lenguajes permiten describir la funcionalidad de un programa de forma precisa, permitiendo al programador usar fácilmente operaciones y estructuras de datos complejas desde el punto de vista del lenguaje ensamblador.
- Existen una gran variedad de lenguajes de programación de alto nivel (C, C++, Java, JavaScript, Python, Haskell y un largo etcétera).

Lenguajes de alto nivel (II)

- El motivo de que existan tantos lenguajes es que cuando se plantea la construcción de un programa se hace para resolver un problema en un ámbito y hay muchos y variados ámbitos. También tenemos diferentes tipos de requerimientos impuestos por las características de los dispositivos de cómputo, tiempo de programación, robustez exigida al programa, etcétera.
- Sin embargo, en última instancia el programa debe ser ejecutado en un procesador por lo que se hace necesario el uso de herramientas que permitan esta objetivo.

Compiladores

- Ya hemos comentado la función del compilador en el caso del lenguaje ensamblador. Recordemos que nos permite obtener una imagen binaria compuesta entre otros elementos por instrucciones máquina.
- Es decir, pasamos de un nivel más abstracto a otro más concreto de forma automática.
- El compilador se caracteriza por que realiza una traducción del programa a un nivel de concreción mayor.
- Los compiladores no siempre traducen a instrucciones máquina del procesador. Existen compiladores que traducen lenguajes de alto nivel a otros lenguajes de alto nivel. Normalmente el lenguaje destino suele ser de mayor concreción.
- Un ejemplo: compilador Glasgow para el lenguaje funcional Haskell: el resultado es un programa equivalente en C que acaba siendo procesado por una toolchain para obtener un programa ejecutable.

Intérpretes

- Una solución alternativa a la del compilador (traducción) es la **interpretación** del programa.
- La interpretación no es una traducción del programa, sino la **ejecución progresiva del mismo, operando comando a comando.**
- La ejecución de un programa de instrucciones máquinas por el procesador es similar: **cada instrucción es traducida a un microprograma que es ejecutado.** Es un proceso de interpretación
- La interpretación de lenguajes de alto nivel es mucho más costosa computacionalmente, que la compilación. Sin embargo, las mejoras en el diseño de los lenguajes y sobre todo en las prestaciones de los computadores ha popularizado enormemente esta forma de ejecutar programas.
- El propio navegador web tiene incorporado un intérprete de Javascript. El lenguaje Java es traducido por un compilador a un lenguaje intermedio que es interpretado por una “máquina virtual”. Lo mismo sucede con los programas Android. Python es otro ejemplo de lenguaje popular interpretado.

El sistema operativo y las librerías

- Los compiladores e intérpretes nos permite aumentar el nivel de concreción de la representación hasta obtener la ejecución del programa.
- Sin embargo, también necesitamos recorrer el camino inverso cuando construimos un programa.
- Este aumento en el nivel de abstracción lo obtenemos incluyendo en nuestro programa “llamadas” a otros programas, concretamente a librerías o partes del sistema operativo que nos proporcionan funcionalidades complejas.
- Esto facilita enormemente la construcción de programas en cualquier nivel. En particular el uso de llamadas a partes del sistema operativo es algo normalmente intrínseco a casi cualquier programa desde el nivel del lenguaje ensamblador.