Introducción

José Otero

[1]Departamento de informática
Universidad de Oviedo

1 de octubre de 2008

Índice

Casi cualquier tarea puede describirse como una secuencia de acciones.
La secuenciación implica un orden en la ejecución de dichas tareas.
Dos secuencias de acciones que se distingan sólo en el orden pueden representar tareas distintas.
En una tarea pueden distinguirse tres tipos de elementos:

- **Procesador:** entidad capaz de entender un enunciado y de ejecutar el trabajo que en el se indica.
- **Entorno:** conjunto de los materiales necesarios para la ejecución del trabajo.
- **Acción:** suceso que modifica el entorno. Tiene lugar en un tiempo finito y produce un resultado bien definido y previsto.

- Un computador es un procesador diseñado para ejecutar listas de acciones relacionadas con el cálculo.
- Entorno= datos + resultados

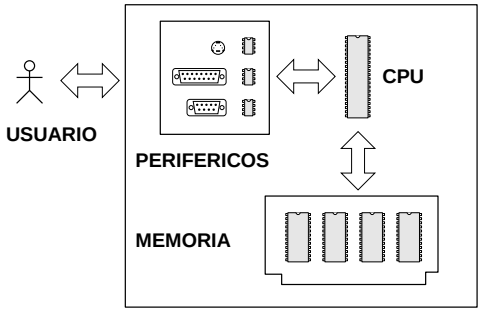
Ejemplo: resolver la ecuación de segundo grado $ax^2 + bx + c = 0$

- Si $d = b^2 - 4ac$ es mayor o igual que cero las soluciones son reales y las da la fórmula $x = \frac{-b \pm \sqrt{d}}{2a}$
- En caso contrario las soluciones son complejas y las da la fórmula $\frac{-b}{2a} \pm \frac{\sqrt{-d}}{2a} i$

- Para *redactar* la descripción anterior, una persona *necesita*:
  - Conocer el proceso de cálculo.
  - Conocer el lenguaje en el que se expresa.
  - Saber describir el proceso en ese lenguaje.
- Para *aplicarla* una persona *necesita*
  - Conocer el lenguaje.
  - Conocer las operaciones involucradas
- *Informalmente*, una descripción de un cálculo es un algoritmo.
- *Informalmente*, un programa es un algoritmo escrito en un lenguaje de programación, ejecutándose en un ordenador.
- Vamos a aprender a escribir programas.

- **Léxico del procesador:** conjunto de acciones e informaciones elementales conocidas por el procesador.
- **Acción primitiva:** acción cuyo enunciado es suficiente para ejecutarla sin información complementaria.
- **Indicador:** el entorno de una acción esta compuesto por indicadores. Puede tomar distintos valores. Cada uno tiene al menos un nombre.
- **Tipo de dato:** cada indicador pertenece a un tipo de dato, entero, real, etc.
- **Estado:** conjunto de los valores de los diferentes indicadores en un instante dado.
- **Algoritmo:** de un proceso es el enunciado de una lista de acciones primitivas que debería realizar el procesador para que, partiendo de un estado inicial, se llegue al estado final que se desea alcanzar.

- El circuito electrónico capaz de ejecutar una secuencia de órdenes se denomina microprocesador.
- Por si mismo es poco útil. Necesita de una serie de componentes para almacenar datos, resultados, programas y para interactuar con el exterior.
- Arquitectura Von Neuman:



- **Unidad central de proceso (cpu):** interpreta y ejecuta las acciones.
- **Memoria:** almacena la lista de acciones y los valores da los indicadores.
- **Periféricos:** permiten introducir y extraer datos en y desde la memoria del computador. Ejemplos: teclado y pantalla.

El microprocesador es gobernado por un reloj.

- Cada instrucción tarda en ejecutarse un número determinado de ciclos de reloj.
- Entre ciclos de reloj no se puede ejecutar ninguna instrucción (más o menos).

Cuando los parámetros de los procesos en los que se utiliza el ordenador varían rápidamente hay que tener en cuenta:

- Los cálculos tardan en realizarse cierto tiempo.
- Cualquier suceso que,como mínimo, dure menos que un ciclo de reloj puede pasar inadvertido.

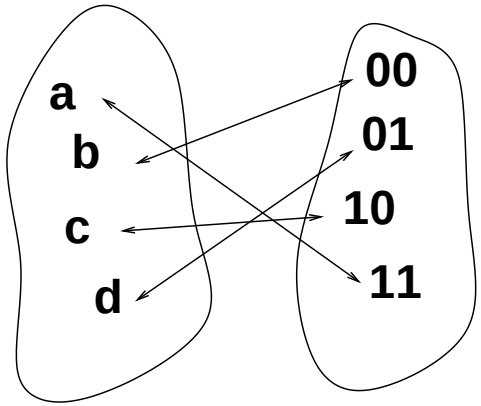
Es muy sencillo, consta de instrucciones para:

- Realizar operaciones aritméticas.
- Realizar operaciones lógicas.
- Decidir si se realiza o no una operación en función del valor de un indicador.
- Repetir varias veces una acción, en función del valor de un indicador.

- Todos los elementos del computador presentan *dos* estados *estables diferenciables*.
- La memoria también.
- El sistema binario permite representar números naturales mediante 1 y 0.
- Se puede representar el 1 con un estado (alto/cerrado) y el cero con otro (bajo/abierto).
- Cualquier información tiene que representarse utilizando este sistema.

- Una cadena tamaño 1 es un bit (*B*inary Dig*it*).
- Es la cantidad mínima de información.
  - Ocho bits se denominan *Byte*.
  - $2^{10} = 1024$ es un Kilo (K).
  - $2^{20} = 1048576$ es un Mega (M).
  - $2^{30} = 1099511627776$ es un Giga (G).
  - $2^{40} = 1208925819614629174706176$ es un Tera (T).

- Una codificación puede definirse como una biyección entre el conjunto de informaciones a representar y un conjunto de cadenas de 1 y 0.
- La biyección podría ser arbitraria.



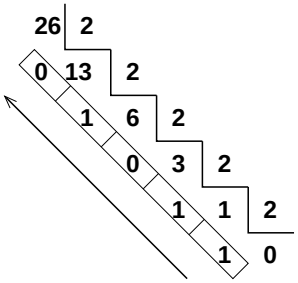
Cuanto mayor es el cardinal del conjunto de la izquierda, mayor debe de ser la longitud de las cadenas.
Con cadenas de longitud n se representan 2^n informaciones diferentes.
Como la memoria es finita, cuando el cardinal del conjunto de la izquierda es infinito:

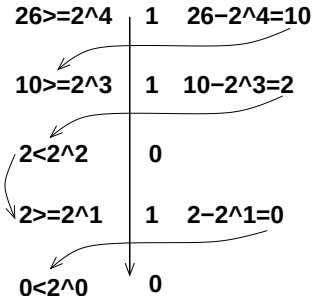
- Hay que acotar el rango de valores.
  - Con 16 bits se pueden representar los enteros del 0 al 65535 o de -32768 a 32767.
- Hay que discretizar el conjunto.
  - No todos los números reales tienen representación en un ordenador.

- Para algunos tipos de información existe un algoritmo para deducir la codificación.
  - Números Naturales: binario natural.
  - Números Enteros: complemento a 2.
  - Números Reales: representación en coma flotante.
- Para otros la codificación es arbitraria.
  - Caracteres: ASCII.

- En un sistema de numeración posicional los dígitos de un número tienen un peso igual a la base del sistema elevado a la posición del dígito dentro del número.
  Ejemplos:
  $1436_{10} = 1 \times 10^3 + 4 \times 10^2 + 3 \times 10^1 + 6 \times 10^0$
  $11011_2 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 =$
  $(16 + 8 + 2 + 1)_{10} = 27_{10}$
- En general:
  $(d_n d_{n-1} d_{n-2} \ldots d_2 d_1 d_0)_b = \sum_{i=0}^{i=n} d_i b^i$

Dos formas comunes de convertir números naturales a binario:





No solo se codifican en binario los datos y resultados.

- Los programas también son información y deben almacenarse en la memoria.
- Por lo tanto también deben codificarse en binario.
- Es difícil para un humano escribirlos directamente en binario.
- Es más fácil escribirlos utilizando un *lenguaje de programación*.

En el caso de algunos lenguajes, después de la codificación, los programas se ejecutan en otro programa denominado *intérprete*.

- Por ejemplo Matlab.

En otros casos, es necesario compilarlos antes:

- El proceso lo realiza otro programa (compilador) a partir del fichero o ficheros *fuente* del programa.
- Si no existen errores en el *uso* del lenguaje, se produce un *ejecutable*.

- Si el programa se ejecuta en un intérprete, puede correr en ordenadores de distinta arquitectura (PC con XP, PC con Linux, Mac,...).
  - Siempre que exista intérprete para esa arquitectura.
- Si el programa se ha escrito en un lenguaje que necesita ser compilado:
  - Los ejecutables sólo corren en la arquitectura para la que se han compilado.
    - La máquina en la que se ejecuten puede ser virtual (Ejemplo:Java).
  - Los fuentes pueden compilarse para cualquier arquitectura, si se dispone del compilador adecuado. Esta característica se denomina portabilidad.

Especificación de requisitos

Texto en el que se detalle de forma clara y no ambigua la tarea que se desea que realice el programa, indicando:

- La finalidad del proceso
- Cual es el conjunto al que pertenecen los datos que procesará el algoritmo.
- Cual es el conjunto al que pertenecen los de los datos que se obtendrán como resultado.

Diseño

Consiste en construir un algoritmo que solucione el problema que se ha especificado.

- Seleccionar varias acciones elementales.
  - No necesariamente definidas en un lenguaje de programación.
  - Equivalentes a composiciones de estas.
- Organizarlas en el tiempo para conseguir el resultado.

Análisis

Estudiar lo eficaz que éste es resolviendo el problema.

- Existen muchos algoritmos distintos que resuelven el mismo problema
- No todos ellos tardan el mismo tiempo.

Codificación

Convertir al lenguaje de programación que se desee el algoritmo.

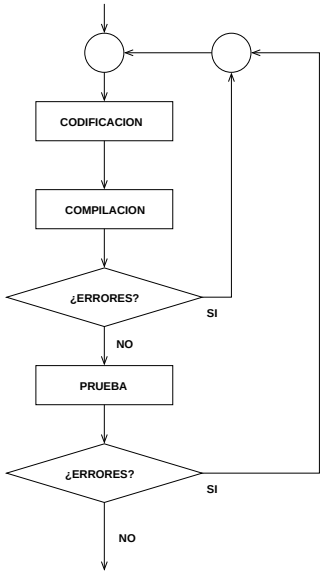
- Se elige una representación para los datos que el programa manejará.
- Se reemplazan las acciones del algoritmo por acciones primitivas válidas en el lenguaje de programación.
- El programa se guarda en uno o varios ficheros de texto. Es indiferente el editor que se utilice.

Verificación

Consta de tres aspectos:

- Demostración de la corrección del algoritmo o verificación formal.
  - Puede hacerse antes de la codificación.
- Prueba del programa.
  - Elegir un conjunto representativo de datos.
  - Comprobar el resultado.
    - Sólo es una prueba si se usan todos las combinaciones de datos posibles.
    - Si algún resultado es incorrecto el programa es incorrecto.
    - En caso contrario no sabemos nada.
- Depuración
  - Detectar y corregir los errores cometidos en la fase de codificación.
  - Existen herramientas para ejecutar el programa paso a paso y examinar los resultados intermedios.

Gráficamente



Más Gráficamente