Llenguatges de Programació
Curs 03-04

Gonzalo Besuievsky
IMA - UdG

Organización

- Horario
  - Miércoles de 9:30 a 13:00
- Tutorías
  - Después de clase
  - por mail: gonzalo@ima.udg.es
- Página del curso
  - http://ima.udg.es/Docencia/03-04/3157RA0008/index.html

Organización

- Teoría
  - Problemas
- Prácticas
  - Lenguaje Pascal
  - Lenguaje Lingo
- Evaluación
  - 10% Problemas + 20% Práctica + 60% Examen +10% Asistencia

Introducción: Contenido

- Conceptos básicos de Programación
  - Algoritmo, Programa, Proceso, Instrucción, Acción
  - Acción simple y acción compuesta
  - Proceso de resolución de un problema
  - Concepto de Análisis descendiente
- Lenguajes de Programación
  - Clasificación por niveles
  - Lenguajes específicos

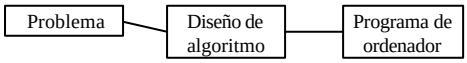
Conceptos básicos

- Algoritmo
  - Método sistemático para resolver un problema
  - Describe la secuencia de pasos ordenada (y no ambigua) que conducen a la solución del problema
- Ejemplo 1: Procesar el pedido de un cliente
  - Inicio
  - Leer pedido
  - Examinar ficha de cliente
  - Si el cliente es solvente, aceptar pedido
    - caso contrario, rechazar pedido
  - Fin
- Ejemplo 2: Calcular la suma de los primeros 100 números pares

Conceptos básicos (2)

- Procesamiento
  - Tratamiento de resolución del problema mediante maquinas automáticas: procesadores
- Programa
  - Codificación del algoritmo de forma que pueda ser interpretado por el procesador
- Lenguaje de Programación
  - Definición de reglas para expresar la codificación

Resolución de un problema



- Pasos a seguir:
  - Diseño del algoritmo
    - análisis del problema y desarrollo del algoritmo
  - Programa
    - Codificación en un lenguaje de programación adecuado
  - Ejecución y validación por ordenador

Definiciones

- Acción
  - Suceso finito en el tiempo que tiene un estado inicial y final
- Acción elemental (o simple)
  - Acción que entiende directamente el procesador (por ej. operación sumar)
- Acción compuesta
  - Acción que debe ser descompuesta en elementales para ser procesada (por ej. Operación multiplicar).
  - El tipo de acción depende de las características del procesador

Especificación del problema

- Objetos
  - Elementos que intervienen en la resolución
- Estado inicial
  - Situación inicial
- Estado final
  - Situación final, resultados esperados
- Partes del algoritmo
  - Entradas → Proceso → Salida

Estructura de datos

- Objetos que se utilizan para almacenar la información necesaria
- Se diseñan junto con el algoritmo
- Ejemplos:
  - números naturales
  - una cadena de texto
  - un plato de cocina (ingredientes, cantidades …)
  - un alumno (dni, nombre, notas)

Lenguajes de Programación

- Conjunto de reglas sintácticas para codificar el algoritmo
- El procesador tiene que ser capaz de comprender cada paso y realizar las operaciones
- Niveles de Lenguajes
  - lenguajes de máquina
  - lenguajes de bajo nivel
  - lenguajes de alto nivel

Lenguajes de Programación (2)

- Acciones (o instrucciones) típicas de un Lenguaje
  - Entrada y Salida
    - Transferencia de información desde periféricos a la memoria central
  - Aritmético - lógica
    - Operaciones aritméticas (+, *, \, pot …) y operaciones lógicas (and, or, not)
  - Instrucciones alternativas
    - Selecciona tareas en función de condiciones
  - Instrucciones repetitivas
    - Permite repetir acciones un número determinado de veces

Lenguajes de máquina

- Expresados en código binario (0 y 1)
- Dependiente del hardware
- Ventaja
  - Velocidad
- Inconveniente
  - Dificultad de codificación y verificación
  - Dependiente del procesador
- Normalmente se utilizan otros lenguajes y traductores

Lenguajes de bajo nivel

- Lenguaje ensamblador (assembly)
- Provee instrucciones básicas
- Opera directamente sobre las posiciones de memoria de los objetos
- Necesita un traductor para convertir a lenguaje de máquina
  - Programa fuente → Assembler → Programa en leng. de máquina
- Inconvenientes
  - Depende de la máquina
  - Poca comodidad para desarrollar programas

Lenguajes de Alto Nivel

- Diseñados para que las personas puedan escribir programas de forma fácil
- Independientes de la máquina
  - Portabilidad entre máquinas
- Ventajas
  - Tiempo de aprendizaje corto
  - Reglas sintácticas similares a lenguajes humanos
  - Verificación y modificaciones más fáciles
- Inconvenientes
  - Necesidad de traductores
  - No se aprovechan todos los recursos internos de las máquinas
  - Aumenta la ocupación de memoria
  - Tiempo de ejecución puede ser mayor

Lenguajes de Alto Nivel (2)

- Ejemplos:
  - C, C++, Pascal, BASIC, Visual BASIC, Fortran
  - Clipper, LISP, Smalltalk
  - Delphi, Eiffel
  - Java, Javascript, html, Perl
  - Lingo
  - ...

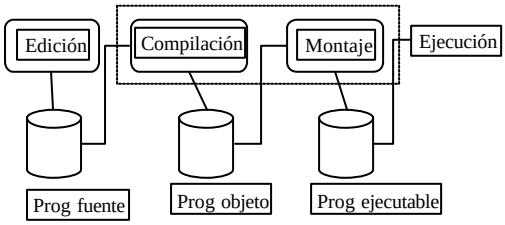
Traductores de Lenguajes

- Son programas que traducen programas fuentes en lenguaje de alto nivel a código de máquina
- Interpretes
  - Traducen línea a línea del código y lo ejecutan
  - Programa fuente → Intérprete → Traducción y ejecución línea a línea
  - Ejemplos:
    - BASIC, LISP, LINGO, Javascript, Perl

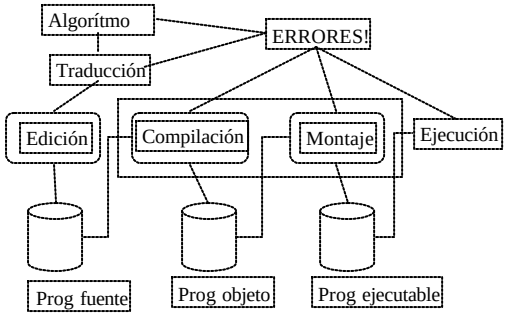
Traductores de Lenguajes (2)

- Compiladores
  - Traducen todo el programa fuente a un programa objeto
  - El programa objeto se utiliza para crear un programa ejecutable
  - Programa fuente → Compilador → Programa Objeto
  - Ejemplos:
    - C, C++, Pascal, Fortran

Fases de la compilación



Fases de creación de un programa



Curso GRAM

- Lenguaje Algoritmico (o pseudocódigo)
  - Lenguaje en notación estándar en catalán que permite escribir algoritmos de forma práctica
  - No se pueden ejecutar
- Lenguaje Pascal
  - Lenguaje estructurada y de sintaxis sencilla
  - Recomendado para iniciación a la programación
  - Utilizaremos fpc (free pascal):
    - http://www.freepascal.org/
- Lenguaje Lingo
  - Lenguaje de “script” de propiedad de Macromedia
  - Funciona con el programa Director
  - Utilizaremos para desarrollar aplicaciones multimedia

Primer programa

```
program hola;

  begin
    writeln('Hola Juancito');
  end.
```

Ejercicio:

Editar, Compilar y Ejecutar.