

Lenguajes de Programación

Métodos de implementación

Sandra Roger

Facultad de Informática
Universidad Nacional del Comahue

Primer Cuatrimestre

¿Qué vimos?

- ▶ Definición de un lenguaje de programación
- ▶ Criterio de evaluación
- ▶ Paradigmas

Ahora: Métodos de Implementación

¿Qué deberíamos tener en cuenta cuando diseñamos un LdP?

- ▶ **La computadora subyacente** en la cual se ejecutarán los programas
- ▶ **El modelo de ejecución**, o computadora virtual que soporta el lenguaje en el hardware real
- ▶ **Metodologías de programación**: El modelo computacional que implementa el lenguaje

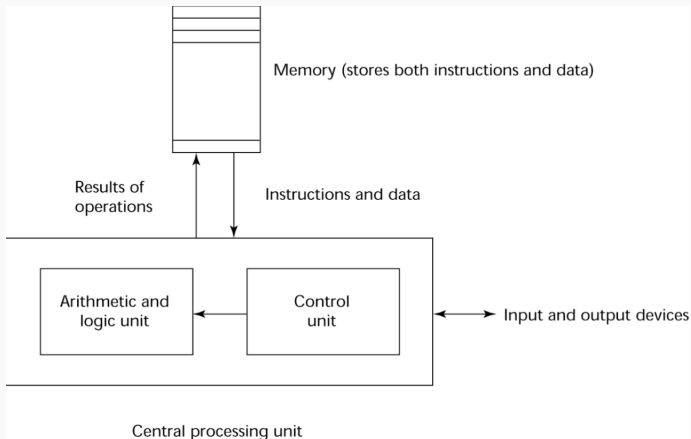
¿Qué deberíamos tener en cuenta cuando diseñamos un LdP?

- ▶ **La computadora subyacente** en la cual se ejecutarán los programas
- ▶ **El modelo de ejecución**, o computadora virtual que soporta el lenguaje en el hardware real
- ▶ **Metodologías de programación**: El modelo computacional que implementa el lenguaje

Influencia de arquitectura de computadoras

- ▶ Los lenguajes imperativos son los más dominantes
 - ▶ Datos y programas almacenados en la memoria
 - ▶ La memoria está separada del CPU
 - ▶ Instrucciones y datos se conectan de la memoria al CPU
 - ▶ Bases para lenguajes imperativos
 - ▶ Celdas de memoria - variables
 - ▶ Sentencias de asignación
 - ▶ Iteración es eficiente

Computadora



Influencias de metodologías de programación

- ▶ **1950s y principios de 1960s:** Aplicaciones simples. Eficiencia de la máquina
- ▶ **Finales de 1960s:** Eficiencia de las personas.
 - ▶ Programación estructurada
 - ▶ Diseño top-down y perfeccionamiento paso-a-paso
- ▶ **Finales de 1970s:** de orientación-a-procesos a orientación-a-datos (abstracción de datos)
- ▶ **Mediados de 1980s:** POO (Abstracción de datos + herencia + polimorfismo)

Computadora

Definición



Una computadora es un *conjunto integrado de algoritmos y estructuras de datos* capaz de *almacenar programas y ejecutar computaciones*:

- ▶ Computadora HW
- ▶ Computadora Virtual: computadora simulada por SW

computadora = algoritmos + ED + levantar un programa y ejecutarlo

Definición

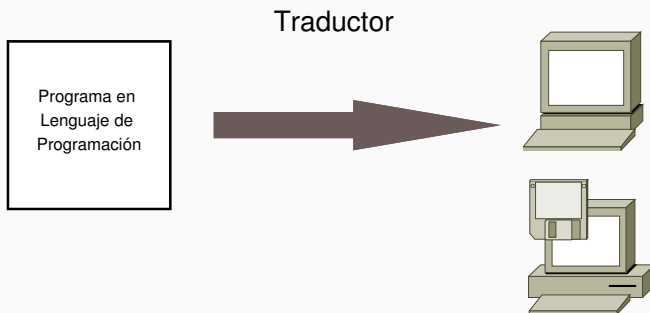


Una computadora es un *conjunto integrado de algoritmos y estructuras de datos* capaz de *almacenar programas y ejecutar computaciones*:

- ▶ Computadora HW
- ▶ Computadora Virtual: computadora simulada por SW

computadora = algoritmos + ED + levantar un programa y ejecutarlo

Implementación de un Lenguaje de Programación



Computadora: Componentes

Componentes principales de una computadora que corresponden a los aspectos principales de un lenguaje de programación (desde el punto de vista del diseñador)

1. **Datos (estructuras)**: Distintos tipos de datos elementales y tipos de datos estructurados
2. Operaciones primitivas
3. Control de Secuencias: Se controla cuál es la secuencia / orden de ejecución de las operaciones primitivas.
4. Acceso a los Datos: Se controla qué datos se les entrega a una operación para cada ejecución
5. Gestión de Almacenamiento: Se controla la disposición del almacenamiento para los programas y los datos.
6. Entorno operativo: Se proveen mecanismos para la comunicación con el entorno externo que contiene datos y programas.

Computadora: Componentes

Componentes principales de una computadora que corresponden a los aspectos principales de un lenguaje de programación (desde el punto de vista del diseñador)

1. **Datos (estructuras)**: Distintos tipos de datos elementales y tipos de datos estructurados
2. **Operaciones primitivas**
3. **Control de Secuencias**: Se controla cuál es la secuencia / orden de ejecución de las operaciones primitivas.
4. **Acceso a los Datos**: Se controla qué datos se les entrega a una operación para cada ejecución
5. **Gestión de Almacenamiento**: Se controla la disposición del almacenamiento para los programas y los datos.
6. **Entorno operativo**: Se proveen mecanismos para la comunicación con el entorno externo que contiene datos y programas.

Computadora: Componentes

Componentes principales de una computadora que corresponden a los aspectos principales de un lenguaje de programación (desde el punto de vista del diseñador)

1. **Datos (estructuras)**: Distintos tipos de datos elementales y tipos de datos estructurados
2. **Operaciones primitivas**
3. **Control de Secuencias**: Se controla cuál es la secuencia / orden de ejecución de las operaciones primitivas.
4. **Acceso a los Datos**: Se controla qué datos se les entrega a una operación para cada ejecución
5. **Gestión de Almacenamiento**: Se controla la disposición del almacenamiento para los programas y los datos.
6. **Entorno operativo**: Se proveen mecanismos para la comunicación con el entorno externo que contiene datos y programas.

Computadora: Componentes

Componentes principales de una computadora que corresponden a los aspectos principales de un lenguaje de programación (desde el punto de vista del diseñador)

1. **Datos (estructuras)**: Distintos tipos de datos elementales y tipos de datos estructurados
2. **Operaciones primitivas**
3. **Control de Secuencias**: Se controla cuál es la secuencia / orden de ejecución de las operaciones primitivas.
4. **Acceso a los Datos**: Se controla qué datos se les entrega a una operación para cada ejecución
5. **Gestión de Almacenamiento**: Se controla la disposición del almacenamiento para los programas y los datos.
6. **Entorno operativo**: Se proveen mecanismos para la comunicación con el entorno externo que contiene datos y programas.

Computadora: Componentes

Componentes principales de una computadora que corresponden a los aspectos principales de un lenguaje de programación (desde el punto de vista del diseñador)

1. **Datos (estructuras)**: Distintos tipos de datos elementales y tipos de datos estructurados
2. **Operaciones primitivas**
3. **Control de Secuencias**: Se controla cuál es la secuencia / orden de ejecución de las operaciones primitivas.
4. **Acceso a los Datos**: Se controla qué datos se les entrega a una operación para cada ejecución
5. **Gestión de Almacenamiento**: Se controla la disposición del almacenamiento para los programas y los datos.
6. **Entorno operativo**: Se proveen mecanismos para la comunicación con el entorno externo que contiene datos y programas.

Computadora: Componentes

Componentes principales de una computadora que corresponden a los aspectos principales de un lenguaje de programación (desde el punto de vista del diseñador)

1. **Datos (estructuras)**: Distintos tipos de datos elementales y tipos de datos estructurados
2. **Operaciones primitivas**
3. **Control de Secuencias**: Se controla cuál es la secuencia / orden de ejecución de las operaciones primitivas.
4. **Acceso a los Datos**: Se controla qué datos se les entrega a una operación para cada ejecución
5. **Gestión de Almacenamiento**: Se controla la disposición del almacenamiento para los programas y los datos.
6. **Entorno operativo**: Se proveen mecanismos para la comunicación con el entorno externo que contiene datos y programas.

Computadora: Elementos

Datos

- ▶ Memoria principal
- ▶ Registros del procesador (alta velocidad de acceso)
- ▶ Archivos externos
- ▶ Memoria cache (alta velocidad)

Computadora: Elementos

Operaciones

- ▶ Un conjunto incluido de operaciones primitivas
- ▶ Operaciones aritméticas en cada tipo de datos incorporado (+, −, ×, /)
- ▶ Controles/test de algunas propiedades de los ítems de datos (verificar si el número es cero, negativo o positivo)
- ▶ Permitir el acceso y modificación de partes de un ítem de dato
- ▶ Controlar los dispositivos de entrada-salida
- ▶ Control de Secuencias (ej, los saltos (jumps))

Computadora: Elementos

Control de secuencia

Utilizando los registros de direcciones

- ▶ Existe un intérprete (¿siempre?):
- ▶ Interpretación y ejecución de programas
 - ▶ Buscar una instrucción
 - ▶ Decodificar la instrucción
 - ▶ Buscar los operandos asignados
 - ▶ Bifurcar a la operación asignada (si corr.)
 - ▶ Ejecutar las operaciones primitivas

Computadora: Elementos

Acceso a datos

- ▶ Acceder a los operandos de la operación
- ▶ Entero para indicar una celda de memoria con algún tipo de dato determinado

Computadora: Elementos

Gestión de almacenamiento

- ▶ Gestionar todos los recursos de la computadora, y que operen lo máximo posible
 - ▶ Memoria
 - ▶ Procesador central
 - ▶ Dispositivos de datos externos

Multiprogramación
Uso de memoria caché

Computadora: Elementos

Entorno operativo

- ▶ El mundo externo a la computadora
conjunto de periféricos y dispositivos de entrada/salida

Visiones de una computadora

Organización de una computadora

- ▶ Visión estática: Estructura
- ▶ Visión dinámica: Estado - dinámica de las operaciones en la estructura estática en la secuencia de transiciones de estados realizadas por la computadora

Computadora Virtual

- ▶ **En un principio:** Construir los programas para usar eficientemente el HW de las computadoras
- ▶ **Porsteriormente:**
 - ▶ No se escribe directamente en código máquina
 - ▶ Se usan “capas” para definir el software
 - ▶ Máquina virtual: Cada capa es una máquina que provee funciones a la próxima capa

Computadora Virtual: Ejecución

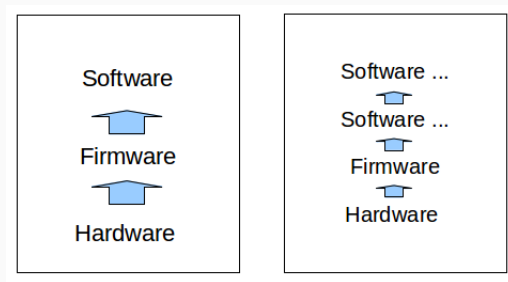
- ▶ Programas:
 - ▶ Datos
 - ▶ Operaciones sobre esos datos
- ▶ Computadora
 - ▶ Implementación de las estructuras de datos
 - ▶ Implementación de las operaciones

Computadora Virtual

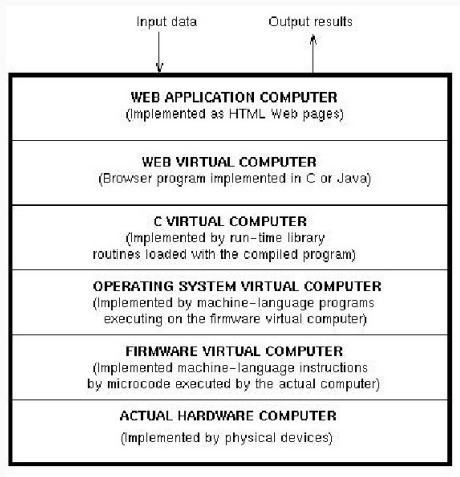
- ▶ **Computadora “Dura” (Hardware):** Datos elementales, operaciones muy simples
- ▶ **Computadora “Semiblanda” (Firmware):**
 - ▶ Datos elementales
 - ▶ Instrucciones en lenguaje máquina (operaciones más complejas y procesadores especializados -MPU-)
- ▶ **Computadora “Blanda” (Software):** Cada entorno de programación establece una computadora “blanda” (de software) específica

Computadora Virtual

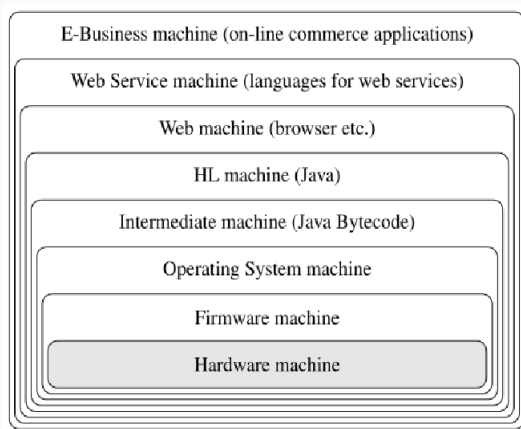
Jerarquía de Máquinas:



Computadora Virtual Ejemplo



Computadora Virtual Ejemplo



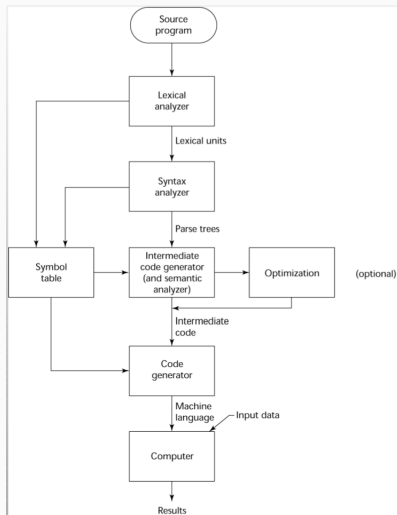
¿Cuál es la idea?

- ▶ **Compilación:** Los programas son traducidos al código de la máquina subyacente.
- ▶ **Interpretación pura:** Los programas son interpretados por otro programa conocido como intérprete.
- ▶ **Sistemas híbridos:**
 - ▶ Compromiso entre los compiladores y los intérpretes puros.
 - ▶ Un programa en lenguaje de alto nivel se traduce a un lenguaje intermedio que permite una fácil interpretación
 - ▶ ej, PERL, JAVA

Compilación

- ▶ Traducir Un programa en lenguaje de alto nivel (lenguaje fuente/origen) en código máquina (lenguaje máquina)
- ▶ Traducción lenta, ejecución rápida
- ▶ El proceso de compilación tiene varias fases:
 - ▶ Análisis léxico: convierte los caracteres del programa fuente en unidades léxicas
 - ▶ Análisis sintáctico: transforma las unidades léxicas en árboles sintácticos que representan la estructura sintáctica de un programa
 - ▶ Análisis semántico: árbol etiquetado
 - ▶ Generación de código intermedio
 - ▶ Generación de código: traducción a código máquina

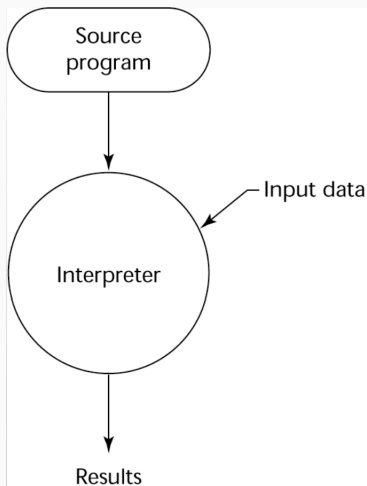
Compilación



Interpretación pura

- ▶ No hay traducción
- ▶ Implementación más sencilla de los programas (errores de tiempo de ejecución, se muestran de forma simple e inmediatamente)
- ▶ Ejecución más lenta (10 a 100 veces más lento que los lenguajes compilados)
- ▶ A veces requieren más espacio
- ▶ No es habitual en los lenguajes de alto nivel
- ▶ Moda en algunos lenguajes script de web (JavaScript, etc.)

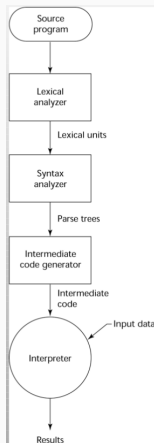
Interpretación



Sistemas Híbridos

- ▶ Compromiso entre los compiladores y los intérpretes puros
- ▶ Un programa en lenguaje de alto nivel se traduce a un lenguaje intermedio que permite una fácil interpretación
- ▶ Muchos más rápido que la interpretación pura
- ▶ Ejemplos
 - ▶ Programas Perl: compilados parcialmente para detectar errores antes de la interpretación
 - ▶ Implementaciones iniciales de java, a forma intermedia (bytecode), permite la portabilidad a cualquier máquina que tenga un intérprete de código y un sistema de ejecución (o Java Virtual Machine - JVM)

Sistemas Híbridos



Pre-Procesadores

- ▶ Las macros de preprocesamiento se usan para especificar que código de otro archivo será incluido, condiciones del código destino, etc.
- ▶ Un pre-procesador procesa el programa antes de que el programa se compile para “expandir” las macros embebidas en el código
- ▶ Ejemplos:
 - ▶ Preprocesador C: `#include` `#define`

Entornos de programación

- ▶ Conjunto de herramientas utilizadas para el desarrollo de software
- ▶ UNIX
 - ▶ Un sistema operativo antiguo con una gran colección de herramientas de desarrollo
 - ▶ Desarrollo usando GUI (CDE, KDE, Gnome)
- ▶ Ejemplos IDE
 - ▶ Borland Jbuilder, NetBeans, Eclipse, Anjuta, MonoDevelop, Allegra, Jdeveloper, etc.
 - ▶ Microsoft Visual Studio.NET
 - ▶ Entornos grande y complejo (de desarrollo)
 - ▶ C#, Visual BASIC.NET, Jscript, J#, or C++

Desafíos

- ▶ Explique el concepto de firmware. ¿Cuál es la ventaja del firmware?
- ▶ ¿Cuál es la idea de simulación de software?. Discuta las ventajas y desventajas.
- ▶ En su computadora, determine la estructura de la computadora virtual que define el sistema operativo ¿En qué difiere esta computadora virtual de la computadora de hw real? ¿Existen características del hw que están restringidas por el sistema operativo, es decir, instrucciones de hw que no se permiten en programas de usuarios en la computadora virtual definida por el sistema operativo? ¿Qué características nuevas se suministran directamente en la computadora virtual del sistema operativo que sólo se podrían simular mediante software complejo en la computadora de hw básica, por ejemplo entrada/salida?

Bibliografía

- ▶ Pratt, Terrance W., Programming Languages: Design and Implementation, 3rd Edition Capítulo 1 y 2.
- ▶ Sebesta, Robert, Concepts of Programming Languages, 9th Edition. 2009. Capítulo 1 y 2.
- ▶ Material de apoyo en pedco.