

Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

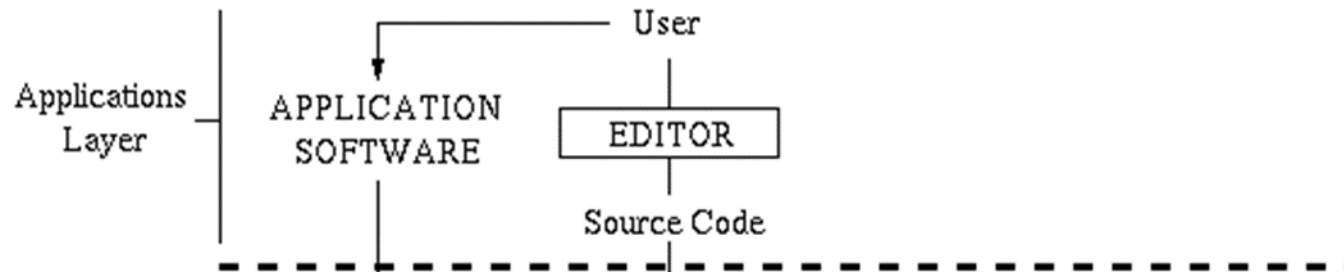


IIC2343 – Arquitectura de Computadores

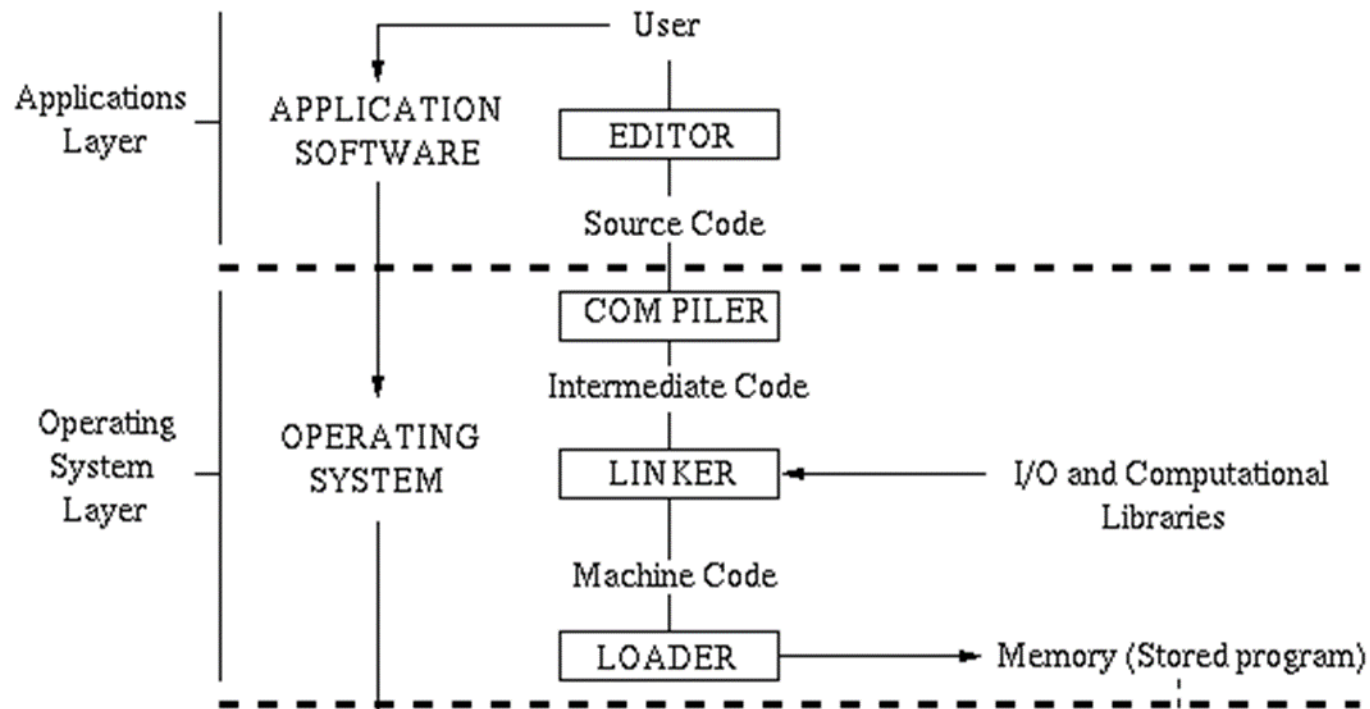
Representaciones Numéricas Parte 1: Números Enteros

Profesor: Hans Löbel

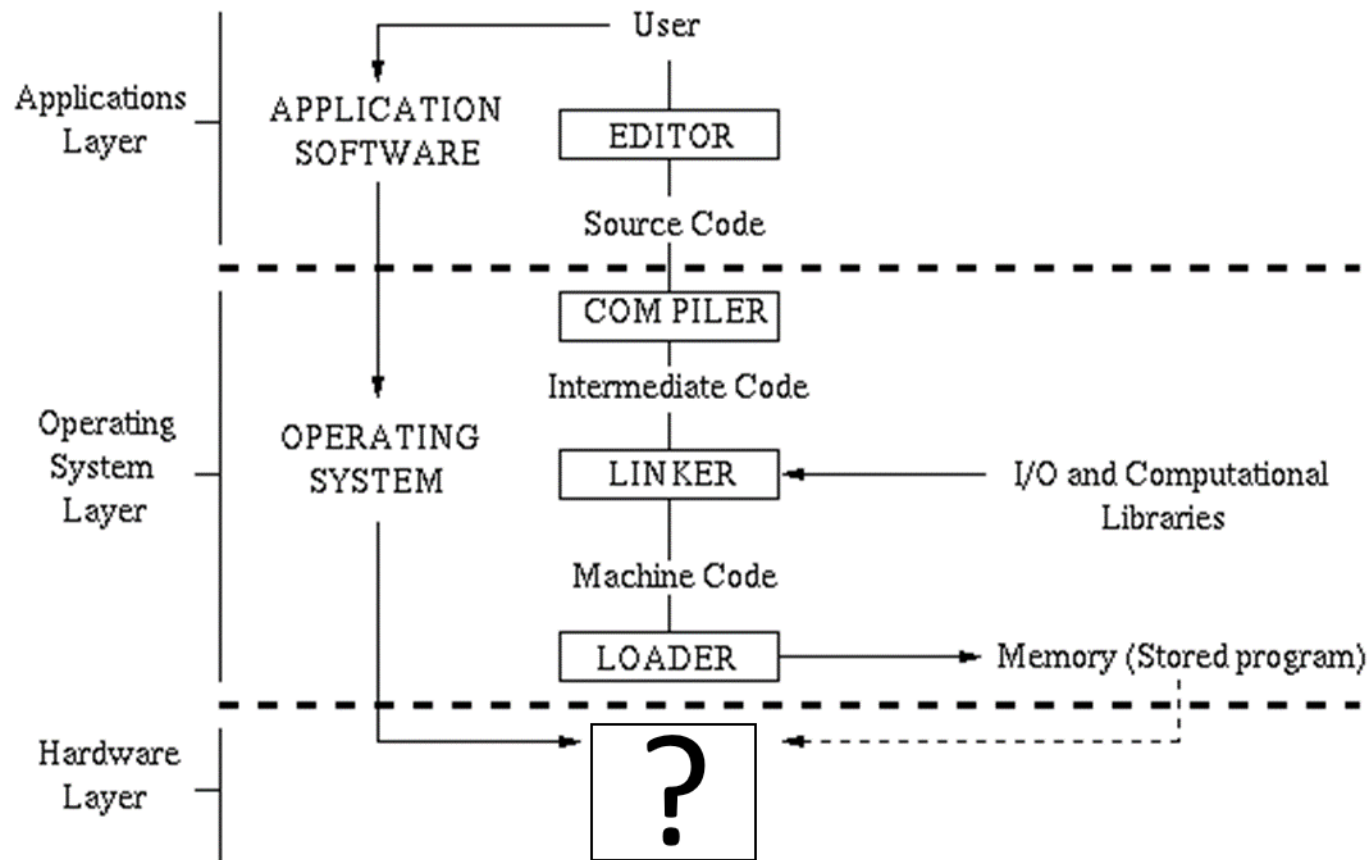
¿Cómo se estructura un sistema computacional (software+hardware)?



¿Cómo se estructura un sistema computacional (software+hardware)?



¿Cómo se estructura un sistema computacional (software+hardware)?

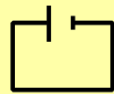




usuario



Sistemas
Digitales



Arquitectura de
Computadores

```
010110
101100
111101
```

Sistemas Operativos

```
010110
101100
111101
```

```
010110
101100
111101
```

Compiladores

```
class A
{
  int a;
}
```

Introducción a
la Programación

Programación
Avanzada

```
class A
{
  int a;
}
```

```
class A
{
  int a;
}
```



desarrollador

¿QUÉ ES UN COMPUTADOR?



Respuesta depende del punto de vista

- En este curso, primará la visión de un computador como una máquina programable (que obviamente también ejecuta los programas).
- La pregunta es entonces: ¿cómo construimos una máquina programable que ejecuta programas?

```
public static void promedio()
{
    int[] arreglo = new int[]{6,4,2,3,5};
    int n = 5;
    int i = 0;
    float promedio = 0;

    while(i < n)
    {
        promedio += arreglo[i];
        i++;
    }
    promedio /= n;
    System.out.println(promedio);
}
```


En resumen

- ¿Qué es un computador?
 - Máquina **programable** que ejecuta programas
- Elementos necesarios para un computador
 - Datos: números enteros y reales, texto, imágenes, etc.
 - Operaciones
 - Variables: simples, arreglos
 - Control: comparación, ciclos

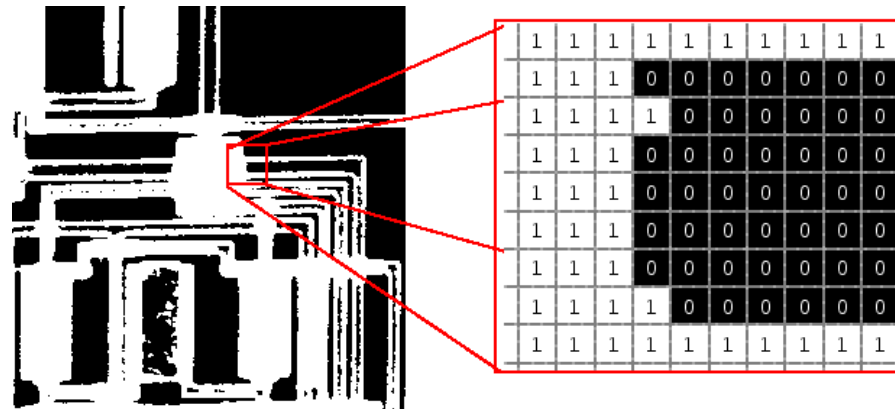
En resumen

- ¿Qué es un computador?
 - Máquina **programable** que ejecuta programas
- Elementos necesarios para un computador
 - **Datos: números enteros y reales, texto, imágenes, etc.**
 - Operaciones
 - Variables: simples, arreglos
 - Control: comparación, ciclos

¿Cómo podemos representar **datos**?

Números enteros son operables, infinitos y enumerables, por lo que pueden representar casi cualquier cosa:

- Cantidades: obvio, ¿o no?
- Caracteres: cada carácter se asocia con un número (ASCII)
PERRO = 80 69 82 82 79
- Imágenes: el color de cada pixel es uno o varios números



Necesitamos representaciones numéricas eficientes

Partamos con los números naturales

- Asumamos que tenemos una cantidad ilimitada de piedras.

¿Cómo podríamos representar con ellas un número **natural** arbitrario?

- Asumamos ahora que tenemos una cantidad limitada de piedras.

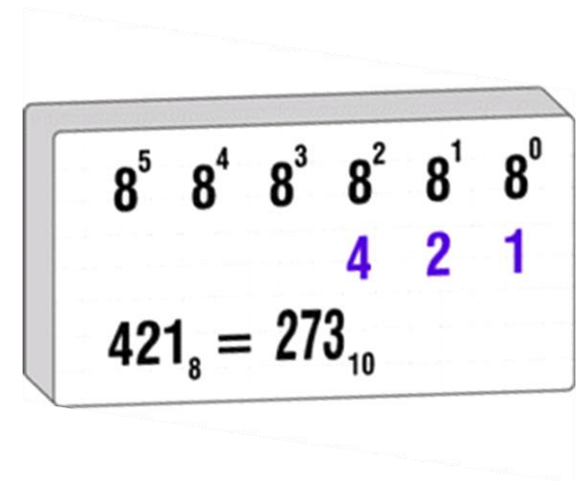
¿Cómo podemos representar en este caso, **de manera eficiente**, un número natural arbitrario?



Representaciones posicionales

- Se basan en 2 elementos:
 - La cantidad de símbolos disponibles (base).
 - La posición de estos en la secuencia.
- El valor de un número se obtiene con la fórmula:

$$\sum_{k=0}^{n-1} s_k \times b^k$$



- Ejemplos: decimal, binario, octal, hexa, etc.
- Aritmética simple y conocida: suma y multiplicación.

Diferenciación de números en distintas bases

- Decimal $\Rightarrow 17$
- Octal $\Rightarrow 21 = (21)_8$
- Binario $\Rightarrow 10001 = (10001)_2 = 10001b$
- Hexa $\Rightarrow 11 = (11)_{16} = 11h = 0x11$

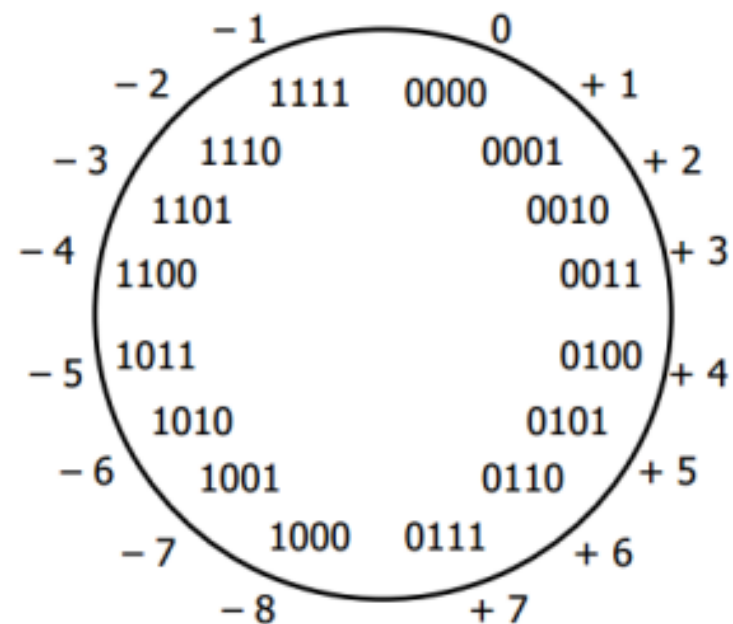
Conversión binario-octal-hexa

- Todos tienen bases que son **potencias de 2**
- Conversión es **trivial** en este caso
- Cuando se baja de base, se expande cada símbolo de acuerdo a la base destino
- Cuando se sube de base, basta con formar grupos de acuerdo al valor de la base y luego transformarlos independientemente

$$\begin{array}{ccccccc} 9F2_{16} & = & & 9 & & F & & 2 \\ & & & \downarrow & & \downarrow & & \downarrow \\ & & & 1001 & & 1111 & & 0010 \\ & & & \text{= } 100111110010_2 & & & & \end{array}$$

Números negativos (enteros)

- Usar signo “-” implica un nuevo símbolo
- 3 opciones para binarios, usando sólo números:
 - Bit de signo: bit extra a la izq. 0 = pos., 1=neg.
 - Complemento a 1: se niega cada dígito.
 - **Complemento a 2:** se niega cada dígito y luego se suma 1. Algebraicamente consistente.



I1 2011/2

- a) Describa una representación y/o mecanismo para obtener el inverso aditivo de un número en representación hexadecimal. Este proceso no debe incluir conversiones a otras representaciones numéricas, como binario o decimal. **(3 ptos.)**
- b) Usando el proceso descrito anteriormente, calcule el resultado de $0x43 - 0x64$. Indique claramente cada una de las etapas del proceso. **(1,5 ptos.)**
- c) ¿Usando este esquema, cuál es el número hexadecimal con mayor valor absoluto que puede representarse usando sólo 2 dígitos?, ¿Por qué? **(1,5 ptos.)**

Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación



IIC2343 – Arquitectura de Computadores

Representaciones Numéricas Parte 1: Números Enteros

Profesor: Hans Löbel