



Ingeniería de Software
Arquitectura de Computadoras
Periodo: 2024-1

Tarea Número	3		
Fecha:	20/10/2023		
Nombres:	Jordy Ortiz, Johnny palaguachi		
Nombre de la Tarea:	Lógica de Programación		
Enlaces a productos elaborados	Agregar o eliminar las filas que necesite		
	Nro	Nombre	Enlace
	1		
	2		

Resumen

En el procesamiento de datos, existen dos tipos de procesos: procesamiento por lotes y procesamiento en línea. El procesamiento por lotes implica almacenar datos y procesarlos más adelante, por ejemplo, en el caso de la nómina quincenal, cualquier actualización sobre horas extras u otros asuntos se recopila semanalmente y se almacena hasta que llegue el momento de procesarla. El procesamiento por lotes no se usa comúnmente hoy en día, mientras que el procesamiento en línea se usa como por ejemplo en escenarios como reservas de aerolíneas donde los datos no se pueden almacenar para su procesamiento posterior.

Objetivos

- Comprender los conceptos básicos de la informática, como el procesamiento compartido y la comunicación de datos.
- Familiarizarse con los principios de programación y los diferentes tipos de lenguajes de programación.
- Reconocer la importancia de la seguridad de datos en la comunicación entre computadoras y dispositivos.

Desarrollo

El avance en los requisitos de procesamiento condujo a la implementación del tiempo compartido, un método en el que el procesador divide su tiempo disponible entre varios usuarios o tareas. Esto indica que el sistema realiza cálculos para dos usuarios simultáneamente mientras produce resultados para un usuario. La asignación de tiempo de procesamiento a cada tarea da la apariencia de procesamiento paralelo.

Las telecomunicaciones y los avances en sistemas de comunicación han facilitado el procesamiento en red, actualmente, un ordenador puede recibir solicitudes de procesamiento de otros ordenadores, ya sea localmente o a grandes distancias, gracias a los avances en las redes de comunicación y las telecomunicaciones.

Cuando se necesita establecer comunicación entre dos o más computadoras o dispositivos similares, se generan códigos en el transmisor. Los códigos se generan en el transmisor cuando es necesaria la comunicación entre dos o más ordenadores o dispositivos similares. Estos códigos suelen pasar por un sistema de codificación de datos que los prepara para ser transmitidos a través del canal de comunicación. Otro mecanismo en el receptor se encarga de descodificar los datos para que la unidad central pueda procesarlos.

Definiciones básicas

A continuación, se presentan una serie de definiciones necesarias para desarrollar el contenido del libro:

- **Soporte físico (hardware):** Son todas las partes tangibles de un computador que se pueden tocar esto comprende la parte eléctrica, electrónica, mecánica, electromecánica, de cableado, periférica, tarjetas ,etc
- **Dirección de memoria:** Toda computadora tiene una cantidad determinada de almacenamiento interno llamada memoria principal. Esta es direccionable, lo que quiere decir que a cada variable o instrucción se la asigna una única dirección de memoria, y esta corresponde a un pequeño espacio en la memoria RAM.
- **Campo:** Es un grupo de caracteres que se pueden tratar como una unidad de información simple y a los que se les asigna una dirección de memoria única. Los campos pueden ser:

CAMPOS CONSTANTES	CAMPOS VARIABLES
<ul style="list-style-type: none"> • Espacio en memoria que almacena un valor que NO cambia. • El valor que almacena corresponde a un tipo básico de dato. • Su valor se asigna con = cuando se declara. • Se utiliza la palabra reservada final en su declaración para indicar que es una constante • Campos constantes numéricos enteros. Formados por dígitos sin el punto decimal. Ejemplos: 251, 1951, 32675, 675432198654 • Campos constantes numéricos reales. Formado por dígitos incluido el punto decimal. Ejemplos: 3.1416, 9.8, 65433.8765 	<ul style="list-style-type: none"> • Espacio en memoria que almacena un valor que puede cambiar. • El valor que almacena corresponde a un tipo básico de dato. • Los valores se le asignan con = cada vez que se quieran cambiar. • Ejemplos: a, ab12, s899, salario, nombres, k, i, nota, salario_bruto, salarioNeto, etcétera.

- **Instrucción:** Son campos (constantes y/o variables) unidos a través de operadores. Por ejemplo: $a=b+c-20$.
- **Programa fuente:** Es el conjunto de determinadas instrucciones relacionadas y escritas en un lenguaje de programación que realizan una tarea específica. El computador no entiende el programa fuente; por lo tanto, hay que traducirlo a lenguaje de máquina (unos y ceros). Para esto se necesita de un traductor, es decir, de un ensamblador, un compilador o un interpretador. Como por ejemplos un programa fuente codificado (representado) en visual C++ :

```
#include<iostream>
using namespace std;
class suma{};
int main(){
    int x, y, z;
    cout<<"DIGITE UN VALOR ENTERO: ";
    cin>>x;
    cout<<"DIGITE OTRO VALOR ENTERO: ";
    cin>>y;
    z= x +y;
    cout<<"LA SUMA DE: "<<x<<"+"<<y<<" ES: "<<z<<endl;
    system("pause");
}
```

- **Ensamblador:** Traduce las instrucciones simbólicas de un lenguaje de programación en las instrucciones necesarias de máquina para que sean ejecutadas. En los lenguajes ensambladores se define un código especial (llamado nemónico) para cada una de las operaciones de la máquina y se introduce una notación especial para especificar el dato con el cual debe realizarse la operación.
- **Compilador:** Son traductores más complejos, usados con lenguajes de alto nivel. Traducen los programas que realiza el usuario al lenguaje de máquina necesario para que sean ejecutados. Además de hacer dicha traducción, el compilador chequea que el programa fuente esté correcto en cuanto a sintaxis e instrucciones propias de cada lenguaje.
- **Interpretador.** Es un programa de computador que traduce cada instrucción o sentencia del lenguaje original a una secuencia de instrucciones del lenguaje de la máquina.
- **Programa objeto.** Es un programa fuente que ha sido traducido a lenguaje de máquina. Es el programa que ejecuta el computador. Se convierte, entonces, en un conjunto de instrucciones de lenguaje de máquina para la solución de un problema, obtenido como el resultado final de un proceso de traducción.
- **Lenguaje de computador:** Conjunto definido de caracteres que sirven para formar símbolos, palabras, instrucciones, etcétera, así como de las reglas para combinar estos caracteres, lo que da como resultado comunicaciones significativas entre algunos ejemplos tenemos FORTRAN, PASCAL, PROLOG, C, C++, Java, entre otros.
- **Lenguaje de máquina:** Es el conjunto final que todos los computadores deben utilizar (lenguaje binario). Todos los demás lenguajes de programación deben compilarse o traducirse, a la larga, a un código binario antes de introducirse en el procesador.
- **Lenguaje de bajo nivel:** Es el primer paso de la sofisticación de los lenguajes del computador. Permite una comunicación más fácil con la máquina. Las instrucciones escritas en este lenguaje deben ser traducidas por medio de un traductor a lenguaje de máquina.
- **Lenguaje de alto nivel:** Es el lenguaje simbólico más fácil de entender y de programar. Se traduce a lenguaje objeto mediante un compilador y permite preparar un programa para ser procesado por el computador. A diferencia de los anteriores, este es más flexible, fácil de aprender y más apropiado para corregir los errores de programación. Una instrucción de lenguaje de alto nivel es traducida a lenguaje de máquina en una o varias instrucciones
- **Archivo:** Es utilizado para almacenar información, relacionada con una determinada entidad, persona o cosas, en forma masiva. Podemos decir entonces que un archivo es un conjunto de registros relacionados. Por ejemplo, suponga que la compañía A tiene seis empleados y que se mantiene la siguiente información acerca de cada uno de ellos: cédula, nombre y salario. La información individual de cada cliente se denomina registro, y un conjunto de registros formará un archivo.
- **Base de datos:** Es un archivo amplio, sin información repetida, relacionado de una manera especial y que continuamente se está actualizando.
- **Representación de datos:** Define la forma en que el computador almacena y procesa diferentes tipos de datos tales como alfanuméricos, numéricos enteros y numéricos reales.
- **Sistema numérico:** Define los símbolos necesarios para representar r dígitos. La cantidad r se conoce como la base numérica y las más comunes son decimal (r=10), binaria (r=2), octal (r=8) y hexadecimal (r=16).

Conclusiones

- Los avances en el procesamiento de datos han llevado al desarrollo del tiempo compartido y el procesamiento en red, lo que ha permitido que múltiples usuarios o dispositivos accedan y compartan recursos de manera más eficiente.
- La codificación y decodificación de datos desempeñan un papel fundamental en la comunicación entre computadoras o dispositivos, facilitando la transmisión de información a través de canales de comunicación.

Recomendaciones

- Dado el papel central de la codificación y decodificación de datos en la comunicación entre dispositivos, es importante garantizar la seguridad y la integridad de los datos durante la transmisión. Se deben utilizar métodos de cifrado y protocolos seguros para proteger la información.
- Con la evolución de los lenguajes de programación y los sistemas informáticos, es crucial que los profesionales de TI y desarrolladores mantengan sus habilidades actualizadas y estén al tanto de las diferencias entre lenguajes de bajo y alto nivel, así como de las tendencias en el desarrollo de software y la administración de sistemas. Esto ayudará a garantizar una programación más efectiva y la adaptabilidad a entornos tecnológicos cambiantes.

Bibliografía

Oviedo Regino, E. M. (2015). Lógica de programación orientada a objetos.. Ecoe Ediciones.
<https://elibro.net/es/ereader/uisekecuador/70431?page=32>