



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

Estructura de Datos y Algoritmos I

Actividad #01 Viernes

Acordeón Lenguaje C

Acordeón de lenguaje de programación con mi inicial
(EUPHORIA)

Alumno:

Flores Ramírez Eduardo Amilcar



28 de febrero del 2021

Acordeón Lenguaje C

La computación, el nuevo medio del mundo.

El uso de un equipo de cómputo se vuelve fundamental para el desarrollo de muchas de las actividades y tareas cotidianas que se realizan día con día, no importando el giro al creando nuevas y versátiles soluciones que apoyen y beneficien directamente a la sociedad al realizar dichas actividades; es por ello, que comprender cómo funciona y cómo poder mejorar dicho funcionamiento se vuelve un tema importante durante la formación del profesionista en ingeniería. Es por lo anterior, que en el desarrollo de proyectos se realizan varias actividades donde la computación es un elemento muy útil. De las actividades que se realizan en la elaboración de proyectos o trabajos podemos mencionar:

- Registro de planes, programas y cualquier documento con información del proyecto en su desarrollo y en producción.
- Almacenamiento de la información en repositorios que sean accesibles, seguros y que la disponibilidad de la información sea las 24 hrs de los 360 días del año.
- Búsqueda avanzada o especializada de información en Internet.

Sistemas operativos.

El Sistema Operativo es el conjunto de programas y datos que administra los recursos tanto de hardware (dispositivos) como de software (programas y datos) de un sistema de cómputo y/o comunicación. Además, funciona como interfaz entre la computadora y el usuario o aplicaciones.

En la actualidad existen diversos sistemas operativos; por ejemplo, para equipos de cómputo están Windows, Linux, Mac OS entre otros. Para el caso de dispositivos móviles se encuentran Android, IOS, Windows Phone entre otros. Cada uno de ellos tiene diferentes versiones y distribuciones que se ajustan a los diversos equipos de cómputo y comunicación en los que trabajan.

Los componentes de un sistema operativo, de forma general, son:

- Gestor de memoria.
- Administrador y planificador de procesos.
- Sistema de archivos.
- Administración de E/S.

Comúnmente, estos componentes se encuentran en el kernel o núcleo del sistema operativo.

En cuanto a la Interfaz con el usuario, las hay de tipo texto y de tipo gráfico. En la actualidad, es común trabajar con la interfaz gráfica ya que facilita mucho seleccionar la aplicación a utilizar; inclusive esta selección se hace “tocando la pantalla” (técnica touch).

Sin embargo, cuando se desarrollan proyectos donde se elaborarán documentos y programas es necesario el uso de dispositivos de entrada y salida (hardware) y aplicaciones en modo texto (software).

Problemas informáticos.

Un problema informático se puede definir como el conjunto de instancias al cual corresponde un conjunto de soluciones, junto con una relación que asocia para cada instancia del problema un subconjunto de soluciones (posiblemente vacío).

Para poder solucionar un problema nos apoyamos en la Ingeniería de Software que de acuerdo con la IEEE se define como “La aplicación de un enfoque sistemático, disciplinado y cuantificable hacia el desarrollo, operación y mantenimiento del software”. Por lo que el uso y establecimiento de principios de ingeniería sólidos, son básicos para obtener un software que sea económicamente fiable y funcione eficientemente.

La Ingeniería de Software provee métodos que indican cómo generar software. Estos métodos abarcan una amplia gama de tareas:

- Planeación y estimación del proyecto.
- Análisis de requerimientos del sistema y software.
- Diseño de la estructura de datos, la arquitectura del programa y el procedimiento algorítmico.
- Codificación.
- Pruebas y mantenimiento (validación y verificación).

Diagramas de flujo.

Un diagrama de flujo es la representación gráfica de un proceso, es decir, muestra gráficamente el flujo de acciones a seguir para cumplir con una tarea específica.

Dentro de las ciencias de la computación, un diagrama de flujo es la representación gráfica de un algoritmo. La correcta construcción de estos diagramas es fundamental para la etapa de codificación, ya que, a partir del diagrama de flujo es posible codificar un programa en algún lenguaje de programación.

Pseudocódigo.

Una vez que un problema dado ha sido analizado (se obtiene el conjunto de datos de entrada y el conjunto de datos de salida esperado) y se ha diseñado un algoritmo que lo resuelva de manera eficiente (procesamiento de datos), se debe proceder a la etapa de codificación del algoritmo.

Para que la solución de un problema (algoritmo) pueda ser codificada, se debe generar una representación de este. Una representación algorítmica elemental es el pseudocódigo.

Un pseudocódigo es la representación escrita de un algoritmo, es decir, muestra en forma de texto los pasos a seguir para solucionar un problema. El pseudocódigo posee una sintaxis propia para poder realizar la representación del algoritmo (solución de un problema).

Sintaxis de pseudocódigo

El lenguaje pseudocódigo tiene diversas reglas semánticas y sintácticas. A continuación, se describen las más importantes:

1. Alcance del programa: Todo pseudocódigo está limitado por las etiquetas de INICIO y FIN. Dentro de estas etiquetas se deben escribir todas las instrucciones del programa.
2. Palabras reservadas con mayúsculas: Todas las palabras propias del pseudocódigo deben de ser escritas en mayúsculas.
3. Sangría o tabulación: El pseudocódigo debe tener diversas alineaciones para que el código sea más fácil de entender y depurar.
4. Lectura / escritura: Para indicar lectura de datos se utiliza la etiqueta LEER. Para indicar escritura de datos se utiliza la etiqueta ESCRIBIR. La lectura de datos se realiza, por defecto, desde el teclado, que es la entrada estándar del sistema. La escritura de datos se realiza, por defecto, en la pantalla, que es la salida estándar del sistema.
5. Declaración de variables: la declaración de variables la definen un identificador (nombre), seguido de dos puntos, seguido del tipo de dato.
6. Operadores aritméticos: Se tiene la posibilidad de utilizar operadores aritméticos y lógicos.
7. Notación de camello. Para nombrar variables y nombres de funciones se debe hacer uso de la notación de camello.

Lenguaje de programación. (Lenguaje C)

Un lenguaje de programación permite expresar una serie de instrucciones que podrán ser realizadas por una computadora. Unos de los lenguajes de programación mayormente difundidos es el lenguaje C.

Éste es muy utilizado ya que la forma de dar instrucciones es muy cercana a lo que un humano podría abstraer, es decir, las instrucciones no son tal cual las que una computadora podría entender, para ello se necesitaría conocer a fondo el microprocesador, el sistema operativo entre otros aspectos. Por esta razón, C es conocido como un lenguaje de alto nivel, esto significa a que las instrucciones podrían ser entendidas fácilmente por un humano. En contraparte, un lenguaje de bajo nivel, son instrucciones que son cercanas a lo que la máquina puede entender y difícilmente pueden ser comprendidas por una persona que no tenga conocimientos de la máquina en que operarán. Algunos autores consideran al lenguaje C

como un lenguaje de mediano nivel, ya que no es totalmente transparente sino tiene elementos que tienen que ver con la arquitectura de la máquina a la hora de programar.

Para realizar un programa usando el lenguaje C, es necesario pensar primero en el sistema operativo que corre sobre la máquina y posteriormente, si este sistema cuenta con interfaz gráfica o sólo posee línea de comandos. A veces, se puede pensar siempre en sólo usar sistemas operativos con interfaz gráfica dado a que su manejo es más sencillo, sin embargo, esta se encuentra limitada para operar toda la funcionalidad del sistema operativo además de que consume recursos de cómputo que pueden ser indispensables para equipos donde el rendimiento es imprescindible. Una vez que se han seleccionado estos elementos, se necesita buscar qué opciones de editores y compiladores están disponibles.

Fundamentos del lenguaje C.

Una vez que un problema dado ha sido analizado (se identifican los datos de entrada y la salida deseada), que se ha diseñado un algoritmo que lo resuelva de manera eficiente (procesamiento de datos), y que se ha representado el algoritmo de manera gráfica o escrita (diagrama de flujo o pseudocódigo) se puede proceder a la etapa de codificación.

La codificación se puede realizar en cualquier lenguaje de programación estructurada, como lo son Pascal, Python, Fortran o PHP. En este curso se aprenderá el uso del lenguaje de programación C.

Estructuras de control.

Las estructuras de control de flujo en un lenguaje especifican el orden en que se realiza el procesamiento de datos.

Las estructuras de selección (o condicionales) permiten realizar una u otra acción con base en una expresión lógica. Las acciones posibles a realizar son mutuamente excluyentes, es decir, solo se puede ejecutar una a la vez dentro de toda la estructura.

Lenguaje C posee 3 estructuras de selección: la estructura if-else, la estructura switch y la estructura condicional o ternaria.

Estructuras de repetición.

Las estructuras de repetición son las llamadas estructuras cíclicas, iterativas o de bucles. Permiten ejecutar un conjunto de instrucciones de manera repetida (o cíclica) mientras que la expresión lógica a evaluar se cumpla (sea verdadera).

En lenguaje C existen tres estructuras de repetición: while, do-while y for. Las estructuras while y do-while son estructuras repetitivas de propósito general.

Depuración de un programa.

Depurar un programa significa someterlo a un ambiente de ejecución controlado por medio de herramientas dedicadas a ello. Este ambiente permite conocer exactamente el flujo de ejecución del programa, el valor que las variables adquieren, la pila de llamadas a funciones, entre otros aspectos. Es importante poder compilar el programa sin errores antes de depurarlo.

Arreglos.

Un arreglo es un conjunto de datos contiguos del mismo tipo con un tamaño fijo definido al momento de crearse.

A cada elemento (dato) del arreglo se le asocia una posición particular, el cual se requiere indicar para acceder a un elemento en específico. Esto se logra a través del uso de índices. Los arreglos pueden ser unidimensionales o multidimensionales.

Los arreglos se utilizan para hacer más eficiente el código de un programa.

Funciones.

Como ya se mencionó, un programa en lenguaje C consiste en una o más funciones. C permite tener dentro de un archivo fuente varias funciones, esto con el fin de dividir las tareas y que sea más fácil la depuración, la mejora y el entendimiento del código.

En lenguaje C la función principal se llama main. Cuando se ordena la ejecución del programa, se inicia con la ejecución de las instrucciones que se encuentran dentro de la función main, y ésta puede llamar a ejecutar otras funciones, que a su vez éstas pueden llamar a ejecutar a otras funciones, y así sucesivamente.

Lectura y estructura de datos.

Un archivo es un conjunto de datos estructurados en una colección de entidades elementales o básicas denominadas registros que son del mismo tipo, pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. Lenguaje C permite manejar la entrada y la salida de datos desde o hacia un archivo, respectivamente, a través del uso de la biblioteca de funciones de la cabecera stdio.h.

Me llamo Eduardo y el lenguaje de programación que yo elegí fue EUPHORIA.

EUPHORIA

EUPHORIA (acrónimo del inglés: "*End User Programming With Hierarchical Objects for Robust Interpreted Applications*") que quiere decir Programación del usuario final con

objetos jerárquicos para aplicaciones robustas interpretadas, es un lenguaje de programación interpretado, de alto nivel, estructurado, imperativo, y motor de juego, simultáneamente, creado por Robert Craig en 1993 como su tesis doctoral, y luego comercializado por la firma RDS (Rapid Deployment Software). EUPHORIA es de dominio público y de código abierto. El código fuente del interprete EUPHORIA está escrito completamente en lenguaje EUPHORIA.

El lenguaje posee soporte a la programación multitarea, que posibilita la ejecución de varias tareas simultáneamente, con una velocidad razonable, y puede ser útil en la creación de juegos que precisan ejecutar múltiples tareas.

Es un lenguaje simple y potente, que, comparado con otros lenguajes interpretados, ofrece una bastante mayor velocidad de ejecución, principalmente en entornos de texto. Aunque no es orientado a objeto su particular flexibilidad en el manejo de datos, rendimiento y sencillez lo convierten en una opción muy atractiva para desarrollos en las plataformas más utilizadas de 32 bits.

En cuanto a la programación en entornos gráficos, el intérprete ofrece una gama tal vez algo limitada de instrucciones, pero se mejora notablemente con el uso de librerías externas y con un entorno de desarrollo IDE/RAD disponible en forma gratuita.

Existe un traductor gratuito de EUPHORIA a C que permite generar a partir de código EUPHORIA un programa en lenguaje C para diversos compiladores disponibles, en DOS, Windows y Linux/BSD, con lo cual su velocidad de ejecución se puede incrementar al máximo.

Características - EUPHORIA fue desarrollado para que cumpliera con las siguientes premisas:

- **Simplicidad:** Facilidad para su aprendizaje y usos rápidos.
- **Poder:** Capacidades de programación en bajo nivel necesarias para acceder a rutinas del sistema operativo, memoria RAM, dispositivos, puertos y BIOS. Enlace simple con rutinas C.
- **Seguridad:** Extenso soporte de depuración, chequeo de tipo de datos y gestión de memoria automática.
- **Flexibilidad:** Soporte para tipos implícitos sencillos, potentes, dinámicos y también otros definidos por el usuario.
- **Velocidad:** Velocidad suficiente para rivalizar con los lenguajes compilados puros.
- **Eficiencia:** Desarrollo rápido de aplicaciones eficientes y robustas, con muy escaso consumo de recursos hardware.
- **Gestión de Memoria:** Gestión automática de memoria virtual, independientemente del sistema operativo.

Tipos de archivos - Los archivos de código más comunes de EUPHORIA corresponden con los siguientes tipos y extensiones:

- **.E** (Biblioteca DOS - texto)
- **.EU** (Biblioteca Linux/FreeBSD - texto)
- **.EW** (Biblioteca Windows - texto)
- **.EX** (Ejecutable con Intérprete externo DOS32 - texto)
- **.EXU** (Ejecutable con Intérprete externo Linux/FreeBSD - texto)
- **.EXW** (Ejecutable con Intérprete externo Windows32 - texto)
- **.IL** (Programa multiplataforma - binario)
- **.EXE** (Ejecutable que incluye un pequeño intérprete - binario)
- **.PRJ** (Proyecto armado mediante la IDE EUPHORIA - texto)

Tipos de datos - EUPHORIA tiene un tipo de dato básico:

- **object:** Un tipo de dato genérico que puede contener cualquier tipo de elemento, y puede ser cambiado durante la ejecución. Eso significa que, si el programador tiene un objeto llamado X que posee el valor 3.172, después el programador puede darle a X el valor "ABC".

Adicionalmente, Euphoria tiene tres tipos de dato "especializados":

- **atom:** Números implementados como enteros o fraccionarios de 32 bit o 64 bit, dependiendo del valor actual. Almacena los valores enteros en 4 bytes.
- **integer:** Una forma especial de atom, restringida a valores enteros de 31 bits. Ese tipo de dato es más eficiente que el atom, pero no puede contener la misma extensión. Los caracteres son almacenados como enteros.
- **sequence:** Vector que puede contener cero o más elementos; cada elemento es un objeto. La cantidad de elementos a contener no es fija; el codificador puede agregar o quitar elementos cuando es requerido durante la ejecución.

Referencias.

Wikipedia. (2019, 5 agosto). EUPHORIA (lenguaje de programación). Recuperado 28 de febrero de 2021, de [https://es.wikipedia.org/wiki/EUPHORIA_\(lenguaje_de_programaci%C3%B3n\)](https://es.wikipedia.org/wiki/EUPHORIA_(lenguaje_de_programaci%C3%B3n))