

Manual Técnico CH-Máquina Final

Un ch-máquina es un sistema operativo que corre en la web, donde se realiza una simulación gráfica de un computador con funcionamiento básico. En este sistema se podrán correr programas con extensión .ch, estos programas pueden cargarse desde cualquier directorio, y tiene como bases fundamentales como la administración de memoria del sistema, la ejecución y administración de programas.

A continuación, se mostrará cuáles son las instrucciones permitidas y cada una de ellas tendrá su descripción.

Operación	Descripción
Cargue	Cárguese/copie en el acumulador el valor almacenado en la variable indicada por el operando
Almacene	Guarde/copie el valor que hay en el Acumulador en la posición de memoria que corresponda a la variable indicada por el operando
Nueva	Crea una nueva variable cuyo nombre es el especificado en el primer operando, en el segundo operando definirá el tipo de variable (C Cadena/alfanumérico, I Entero, R Real/decimal, L lógico o booleano (1 Verdadero o 0 Falso), un tercer operando establecerá un valor de inicialización. A cada variable se le asignará automáticamente una posición en la memoria. Las variables deberán estar definidas antes de ser utilizadas. Las variables no inicializadas tendrán por defecto el valor cero para reales y enteros, espacio para cadenas, 0 para lógicos. El separador de decimales es el punto.
Lea	Lee por teclado/pantalla el valor a ser asignado a la variable indicado por la variable referida en el operando
Sume	Incrementa/sume al valor del acumulador el valor indicado por la variable señalada por el operando.
Reste	Decrementa/reste del valor del acumulador el valor indicado por la variable que señala el operando.
Multiplique	Multiplique el valor del acumulador por el valor indicado por la variable señalada por el operando.
Divida	Divida el valor del acumulador por el valor indicado por la variable señalada por el operando. El divisor deberá ser una cantidad diferente de cero.

Potencia	Eleve el acumulador a la potencia señalada por el operando (los exponentes pueden ser valores enteros, positivos o negativos)
Modulo	Obtenga el modulo al dividir el valor del acumulador por el valor indicado por la variable señalada por el operando.
Concatene	Genere una cadena que una la cadena dada por el operando a la cadena que hay en el acumulador (Operando alfanumérico). El contenido del acumulador deberá tratarse como cadena en caso de ser numérico.
Elimine	Genere una subcadena que elimine cualquier aparición del conjunto de caracteres dados por el operando de la cadena que se encuentra en el acumulador (operando alfanumérico).
Extraiga	Genere una subcadena que extraiga los primeros caracteres (dados por el operando con valor numérico) de la cadena que se encuentra en el acumulador.
Y	Produce una operación lógica Y (AND) entre el primer operando y el segundo operando que son variables lógicas y la almacena en el tercer operando.
O	Produce una operación lógica O (OR) entre el primer operando y el segundo operando que son variables lógicas y la almacena en la variable del tercer operando.
NO	Produce una operación de negación lógica para el primer operando que es una variable lógica y el resultado se almacena en la variable del segundo operando.
Muestre	Presente por pantalla el valor que hay en la variable indicada por el operando, si el operando es acumulador muestre el valor del acumulador.
Imprima	Presente por la impresora el valor que hay en la variable indicada por el operando, si el operando es acumulador muestre el valor del acumulador
Retorne	El programa termina; debe ser la última instrucción del programa y tiene opcionalmente un operando numérico entero.
Vaya*	Salte a la instrucción que corresponde a la etiqueta indicada por el operando y siga la

	ejecución a partir de allí.
Vayasi*	Si el valor del acumulador es mayor a cero salte a la instrucción que corresponde a la etiqueta indicada por el primer operando y continúe la ejecución a partir de allí. Si el valor del acumulador es menor a cero salte a la instrucción que corresponde a la etiqueta indicada por el segundo operando y continúe la ejecución a partir de allí. o Si el acumulador es cero salte a la siguiente instrucción adyacente a la instrucción vayasi* y siga la ejecución a partir de allí.
Etiqueta	La etiqueta es un nombre que opcionalmente se le puede asignar a una instrucción en el programa para evitar trabajar con las posiciones en memoria de las instrucciones y poder utilizar un nombre simbólico independiente de su ubicación. Crea una nueva etiqueta cuyo nombre es el especificado en el primer operando y a la cual le asignará automáticamente la posición indicada en el segundo operando (esta será la posición relativa de la instrucción a la que se le asigna este nombre con respecto a la primera instrucción del programa). Las instrucciones que definen etiquetas podrán definirse en cualquier posición del programa, pero en todo caso antes de la instrucción retorne.

Reglas:

Se asume que el sistema operativo ocupa las primeras posiciones de la memoria, su contenido no es importante y su tamaño se puede variar solo al iniciar el programa. El programa utilizará un acumulador para registrar los valores de los cálculos y recibirá como nombre reservado "acumulador". Toda posición de memoria que se utilice para almacenar datos, deberá ser creada antes de ser usada y tendrá un nombre bajo las reglas ya establecidas. Las instrucciones constan de 2 partes; el código de la operación y el(los) operando(s) dependiendo el tipo de instrucción.

Administración de Memoria:

La administración de memoria se basa en crear un vector de 99999 espacios, y se debe mostrar su funcionamiento en la pantalla los aspectos como por ejemplo que contiene en el momento y que se está cargando, se requiere simular el sistema operativo en la memoria, el cual va a requerir un espacio en ella, con esto entonces se reservará la cantidad $10 \cdot Z + 3$ posiciones, donde Z es el último dígito de la cédula y se ubicará en los primeros espacios de memoria.

Ejecución y administración de programas:

La manera de ejecución y administración de programas es como se representa la acción que hace el sistema para mostrar los programas como normalmente lo conocemos. El sistema operativo no contiene la misma cantidad de herramientas y opciones de las convencionales, solo acepta un lenguaje de programación que se conoce como ch-código, este es un lenguaje de programación muy sencillo.

En este proceso los programas se partirán en sub-procesos con los que se entenderá un poco más como realizar dichos procesos como verificación de sintaxis de los .ch que son los archivos que se tienen el lenguaje ch-código.

Verificación de sintaxis de los archivos con extensión .ch:

En todo lenguaje de programación y más cuando es manejado por el ser humano existe gran posibilidad de que haya errores como en declaración de variables, hasta un ciclo mal estandarizado, etc. Por eso se requiere que todos esos errores de mecanografía sean corregidos y eventualmente se puedan verificar cada una de las líneas de código para asegurar que se utilice de manera correcta las palabras reservadas y se declare de manera óptima cada una de las variables.

Cargar los .ch a la memoria:

Se ha realizado anteriormente la verificación de los archivos .ch y se continúa con la carga de cada uno de los archivos con sintaxis del código a la memoria, las cuales se cargarán a la memoria dejando el espacio libre de los que se reservaron para el sistema operativo.

Interpretación y ejecución del ch-código:

La interpretación es el manejo de cada una de las líneas en ch-Código y que sean traducidas a un lenguaje conocido, de esta manera se simulará el proceso normal que lleva a cabo cada uno de los programas del sistema operativo a la hora de llevarlo a lenguaje máquina.

La ejecución del programa es bastante sencilla si se han seguido todos los pasos anteriores ya que se puede asegurar que todo lo que se corre por el ch-Maquina está debidamente estandarizado y si podrá realizar todas las labores para lo que ha sido realizado el ch-Programas.

ESPECIFICACIONES TÉCNICAS Lenguaje Utilizado

- JavaScript :

JavaScript es un lenguaje de programación interpretado. Es orientado a objetos y basado en prototipos imperativos, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas, en bases de datos locales al navegador. Su uso en aplicaciones externas a la web, por ejemplo, en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo.

- CSS:

Cascading Style Sheets, hojas de estilo en cascada de ahí sus siglas CSS.

CSS es un lenguaje que se utiliza para definir la presentación de un documento estructurado escrito en HTML o XML2 (y por extensión en XHTML). Es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores.

La información de estilo puede ser adjuntada como un documento separado o en el mismo documento HTML. En este último caso podrían definirse estilos generales en la cabecera del documento o en cada etiqueta particular mediante el atributo "<style>"

- HTML:

Siglas de HyperText Markup Language («lenguaje de marcado de hipertexto»). Este lenguaje es predominante para la realización de páginas web, se usa para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.

El HTML se escribe en forma de

«etiquetas», rodeadas por corchetes angulares (<,>). HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo, JavaScript), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML.