

Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Computación

Bachillerato en Ingeniería en Computación



IC-3101 Arquitectura de Computadoras

Profesor: Ing. Eduardo A. Canessa Montero, M.Sc.

Tarea Programada #2

Integrantes:

Sara Castro Sáenz 2014085332

Ericka Céspedes Moya 2017239557

José David Martínez Garay 2018160104

Fecha de Entrega: 26 de noviembre.

II Semestre, 2018

I. Descripción de diseño y arquitectura.

El algoritmo está implementado en una arquitectura de x64, este se compone de 5 módulos principales; ingreso de datos, reemplazo de variables, la validación de la ecuación, el procedimiento que realiza la ecuación y la validación de la salida (verificar que sea representable en 64 bits).

Se ha elegido la utilización de la arquitectura x64 por su facilidad para realizar llamadas del sistema, métodos de retorno y la utilización de 64 bytes para cada registro de propósito general y específico.

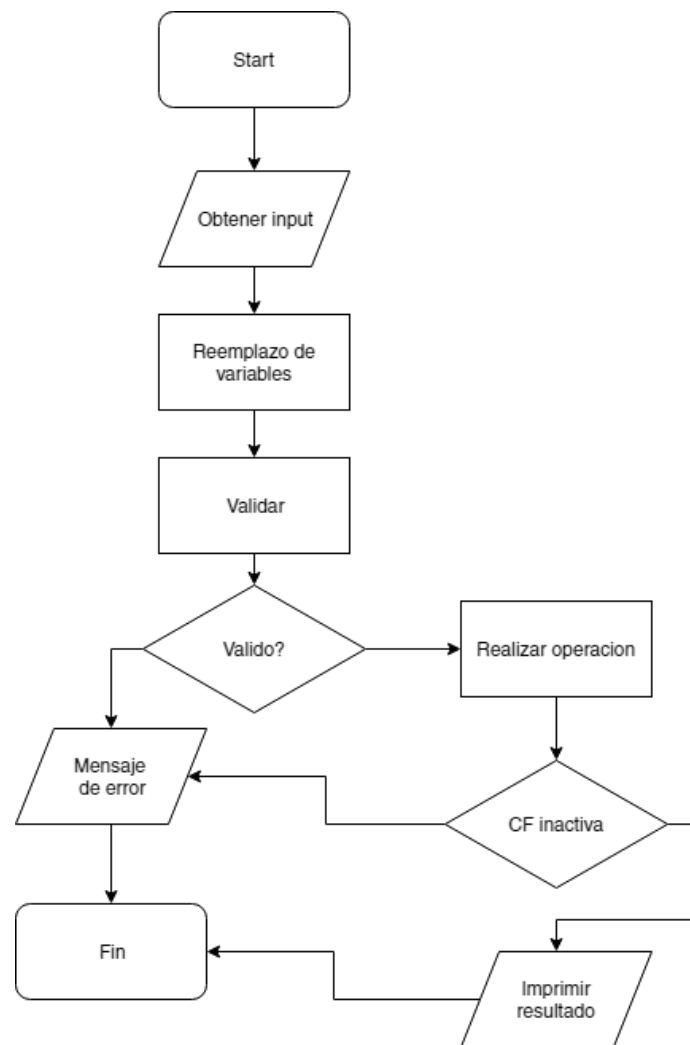


Diagrama general del programa

Bloques del programa

Ingreso de datos

Para obtener la ecuación, solamente se realiza la llamada de lectura al sistema.

Reemplazo de variables

En este procedimiento se reemplazan todas las variables por su valor numérico correspondiente, si no tiene variables, solo se copia la ecuación tal cual como la ingreso el usuario.

Validación de la ecuación

Una vez que se han reemplazado las letras por los valores numéricos, se llama el procedimiento para validar. Para llevar a cabo la validación, se lee la tira de caracteres y se suma o se resta en un registro dependiendo del carácter que sea, si al final el registro no está en cero se despliega un mensaje de error.

Solución de la ecuación

Después de validar la tira de caracteres, se procede a resolver la ecuación.

Validación de la salida

Cuando se realizan las operaciones, se realiza un jump si la bandera de carry se ha activado para lanzar un mensaje de error en caso de que el número no se pueda representar.

II. Descripción técnica

Ingreso de datos

Para obtener la ecuación, solamente se realiza la llamada de lectura al sistema utilizando un macro que realiza la llamada.

Reemplazo de variables

Después de obtener la ecuación, se llama a *change* procedimiento que se encarga de reemplazar las variables por valores numéricos en una ecuación, si esta posee variables, si no, solamente copia la tira obtenida por SYSTEM_READ en la otra tira que es la que se utiliza para realizar la ecuación.

Para reemplazar las variables de la tira, esta se lee caracter por caracter, copiandolos en una tira nueva, hasta que encuentra una letra, cuando encuentra una letra se llama a otro procedimiento que busca el valor numérico de esa variable y lo agrega en la nueva tira de caracteres, después de agregar el numero, agrega cualquier carácter diferente de una letra a la nueva tira hasta que encuentra una letra o una coma, si es una letra, vuelve a repetir el procedimiento, si es una coma termina ya que ha llegado al final de la ecuación.

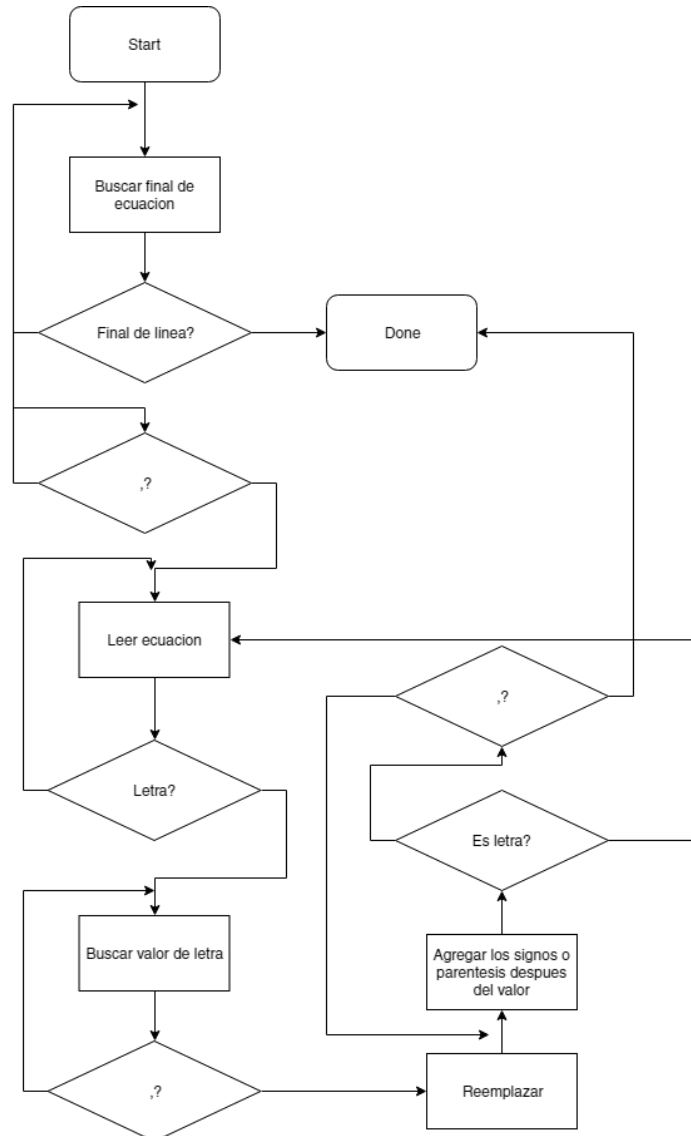
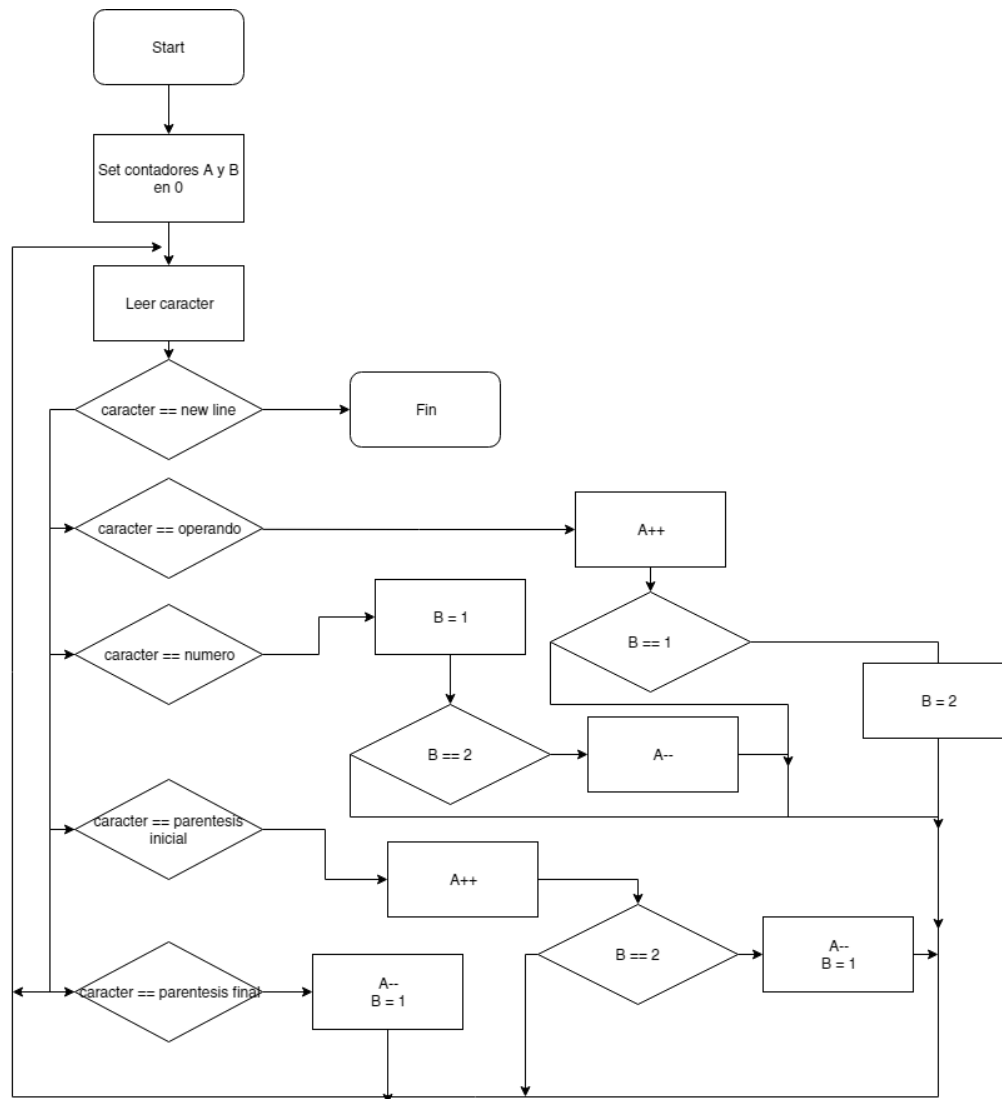


Diagrama de reemplazo de variables

Validación de la ecuación

Una vez que se han reemplazado las letras por los valores numéricos la ecuación está lista para validarse por lo que se llama a a *validate* procedimiento que lee la tira de caracteres y corrobora que sea una ecuación válida. Para llevar a cabo esto, se utiliza un registro (rax) en el que se setea un número que permite saber el tipo de carácter del cual se proviene y así poder realizar un incremento o decremento en el registro rdx, el cual se utiliza para comprobar si es una ecuación válida o no cuando se termina de leer toda la línea de caracteres. Para comprobar esto se compara rdx con cero, si es diferente a cero significa que la ecuación no es válida.



Solución de la ecuación

Después de validar la tira de caracteres, se procede a resolver la ecuación.

Validación de la salida

Cuando se realizan las operaciones, se realiza un jump si la bandera de carry se ha activado para lanzar un mensaje de error en caso de que el número no se pueda representar.

Macros utilizados

<pre>%macro print 2 mov rdx, %1 mov rsi, %2 mov rdi, 1 mov rax, SYS_WRITE syscall %endmacro</pre>	<p>Realiza una llamada al sistema para imprimir en consola.</p> <p>Recibe dos parámetros, el largo de la tira de caracteres (%1) y la tira de caracteres (%2)</p>
<pre>%macro exit 0 mov rax, SYS_EXIT syscall %endmacro</pre>	<p>Realiza una llamada al sistema para salir del programa.</p> <p>No recibe parametros.</p>
<pre>%macro if_something 1 xor rdx, rdx call %1 cmp rdx, 1 %endmacro</pre>	<p>Setea el registro rdx antes de llamar a un procedimiento y hace una comparación del resultado.</p> <p>Recibe como parámetro el procedimiento que se desea llamar.</p>
<pre>%macro read 2 mov rdx, %1 mov rsi, %2 mov rdi, 0 mov rax, 0 syscall %endmacro</pre>	<p>Realiza una llamada al sistema para leer el input del usuario en la consola.</p> <p>Recibe dos parámetros, el largo de la tira de caracteres (%1) y la tira de caracteres (%2)</p>

III. Manual de usuario

Modos de ingresar ecuación

Con variables:

Ingresa una ecuación con variables, estas deben ser letras minúsculas de la a a la z. Una vez ingresada la ecuación, escriba una coma para marcar el final de esta y escriba el valor de cada una de las variables de la forma *variable=valor*,

Ejemplos:

$(a+b)+a+c*d+(a+c)$,a=23,b=34,c=253,

$(3+b)*z/9$,b=123,

a-b+c/d+e*f+g-h/i+j+k*l+m+n*o*p/q+r-s+t,a=12,b=123,c=43,d=9,e=45,f=5432,g=90,h=2,i=142,j=654,k=9,l=342,m=543,n=654,o=753,p=543,q=533,r=264,s=64,t=543,

Nota: Siempre se debe colocar una coma después de cada número

Sin variables

Ingresa la ecuación sin las variables, no escriba la coma al final

Ejemplos:

2+2

2+(6*(3+5))