**Laboratorio 1**

Construcción de la imagen del contenedor

A screenshot of a computer

Description automatically generated



Creación de archivo ScriptShell

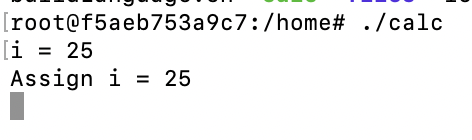


Ejecición de buildLanguage.sh

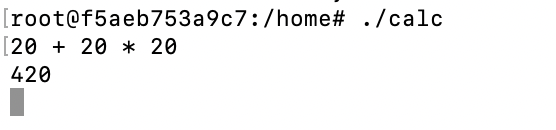


*Ejercicios*

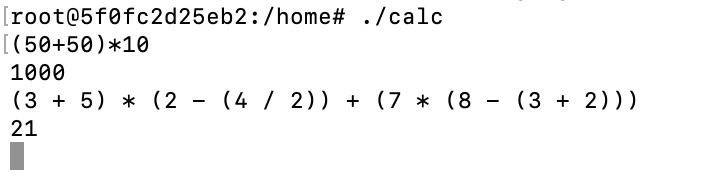
1. *Cree un programa que asigne un valor a una variable.*

**

1. *Cree un programa que realice una operación aritmética simple.*

**

1. *Experimente con expresiones más complejas y verifique que el compilador las procese correctamente.*

**

1. *Modifique el lenguaje para incluir la asignación de variables con expresiones aritméticas.*

*A white background with black text

Description automatically generated*

1. *Agregue manejo de errores al compilador para detectar tokens invÅLalidos en el programa fuente.*

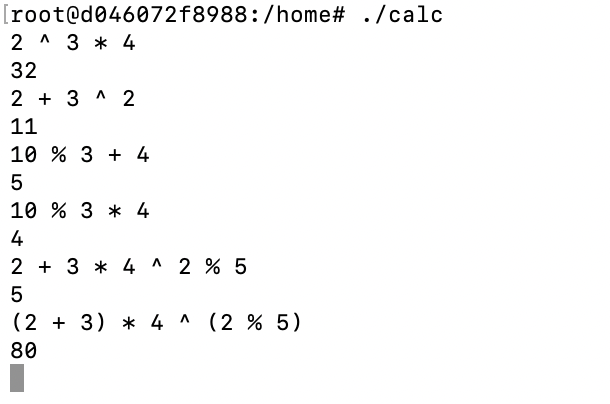
*A screenshot of a computer

Description automatically generated*

*A close up of a number

Description automatically generated*

1. *Experimente con la precedencia de operadores en el lenguaje y observe cómo afecta la generación del arbol sintactico*

**

**Resultados de las Expresiones**

* Expresión: 2 ^ 3 \* 4

Evaluación:

2 ^ 3 se evalúa primero debido a la mayor precedencia de la potenciación, resultando en 8. Luego, 8 se multiplica por 4.

Resultado:

8 \* 4 = 32

Explicación:

La potenciación tiene mayor precedencia que la multiplicación.

* Expresión: 2 + 3 ^ 2

Evaluación:

3 ^ 2 se evalúa primero debido a la mayor precedencia de la potenciación, resultando en 9. Luego, 2 se suma a 9.

Resultado:

2 + 9 = 11

Explicación:

La potenciación tiene mayor precedencia que la suma.

* Expresión: 10 % 3 + 4

Evaluación:

10 % 3 se evalúa primero debido a la mayor precedencia del módulo, resultando en 1. Luego, 1 se suma a 4.

Resultado:

1 + 4 = 5

Explicación:

El módulo tiene mayor precedencia que la suma.

* Expresión: 10 % 3 \* 4

Evaluación:

10 % 3 se evalúa primero debido a la mayor precedencia del módulo, resultando en 1. Luego, 1 se multiplica por 4.

Resultado:

1 \* 4 = 4

Explicación:

El módulo tiene mayor precedencia que la multiplicación.

* Expresión: 2 + 3 \* 4 ^ 2 % 5

Evaluación:

Primero, 4 ^ 2 = 16 debido a la mayor precedencia de la potenciación.

Luego, 3 \* 16 = 48 debido a la multiplicación.

Luego, 48 % 5 = 3 debido a la mayor precedencia del módulo.

Finalmente, 2 + 3 = 5.

Resultado:

5

Explicación:

La potenciación tiene mayor precedencia, seguida de la multiplicación y el módulo, y finalmente la suma.

* Expresión: (2 + 3) \* 4 ^ (2 % 5)

Evaluación:

Primero, 2 % 5 = 2 debido a la mayor precedencia del módulo dentro de los paréntesis.

Luego, 4 ^ 2 = 16 debido a la mayor precedencia de la potenciación dentro de los paréntesis.

Luego, 2 + 3 = 5 debido a la suma dentro de los paréntesis.

Finalmente, 5 \* 16 = 80.

Resultado:

80

Explicación:

Los paréntesis cambian la precedencia normal, forzando la evaluación de las operaciones dentro de ellos primero.

*Sin precedencia*

*A screen shot of a computer code

Description automatically generated*

**Resultados de las Expresiones**

A screenshot of a math problem

Description automatically generated

En estos ejemplos se puede observar que no se sigue la precedencia de operaciones, lo que resulta en cálculos incorrectos y, en algunos casos, inconsistentes, incluso cuando la expresión es la misma. Por eso, es importante declarar las directivas %right y %left, que se utilizan para establecer la precedencia y asociatividad de los operadores, resolviendo ambigüedades en las expresiones matemáticas. En Yacc, la precedencia de los operadores se establece en orden ascendente, es decir, los operadores con menor precedencia se declaran primero, mientras que los de mayor precedencia se declaran al final. Además, Yacc tiene reglas predeterminadas para resolver conflictos shift-reduce, como preferir shift sobre reduce. Esto puede hacer que los conflictos no sean evidentes de inmediato, pero garantizará una evaluación coherente y correcta de las expresiones (IBM, 2023).

Link a repositorio:

<https://github.com/isaackeitor/CC-2024.git>

Referencias

IBM. (2023). yacc grammar file declarations. Obtenido de https://www.ibm.com/docs/en/aix/7.2?topic=information-yacc-grammar-file-declarations