

UNIVERSIDAD DEL VALLE DE GUATEMALA

Construcción de compiladores CC3032

Sección 10

Bidkar Pojoy



Laboratorio 01

Guiado

José Daniel Gómez Cabrera 21429

Github

<https://github.com/JDgomez2002/compilers-lab01.git>

Imagen de Docker

```
lab-1 — root@865ce0f2efab: /home — docker run --rm -ti -v ~/Documents/UVG/compilers/labs/compilers-2024-students/lab-1:/home lab1-image — 172x41
[jdgom...@MacbookPro lab-1 % docker build --rm . -t lab1-image && docker run --rm -ti -v "$(pwd)":/home lab1-image
[+] Building 0.7s (10/10) FINISHED
⇒ [internal] load build definition from Dockerfile
⇒ ⇒ transferring dockerfile: 24kB
⇒ [internal] load metadata for docker.io/library/ubuntu:latest
⇒ [auth] library/ubuntu:pull token for registry-1.docker.io
⇒ [internal] load .dockerrignore
⇒ ⇒ transferring context: 2B
⇒ [1/4] FROM docker.io/library/ubuntu:latest@sha256:2e863c44b718f27c860746568e1d54afdf13b2fa71b160f5cd9058fc436217b30
⇒ [internal] load build context
⇒ ⇒ transferring context: 113B
⇒ → CACHED [2/4] RUN apt-get update -y && apt-get install -y --no-install-recommends build-essential bison flex && rm -rf /var/lib/apt/lists/*
⇒ → CACHED [3/4] COPY files /home/files
⇒ → CACHED [4/4] WORKDIR /home
⇒ exporting to image
⇒ ⇒ exporting layers
⇒ ⇒ writing image sha256:c8d4195244a1532f41929626bbd9e86cae2d531acc3a02fde66ca7cc134a4ecc
⇒ ⇒ naming to docker.io/library/lab1-image
root@865ce0f2efab:/home#
```

Compilación

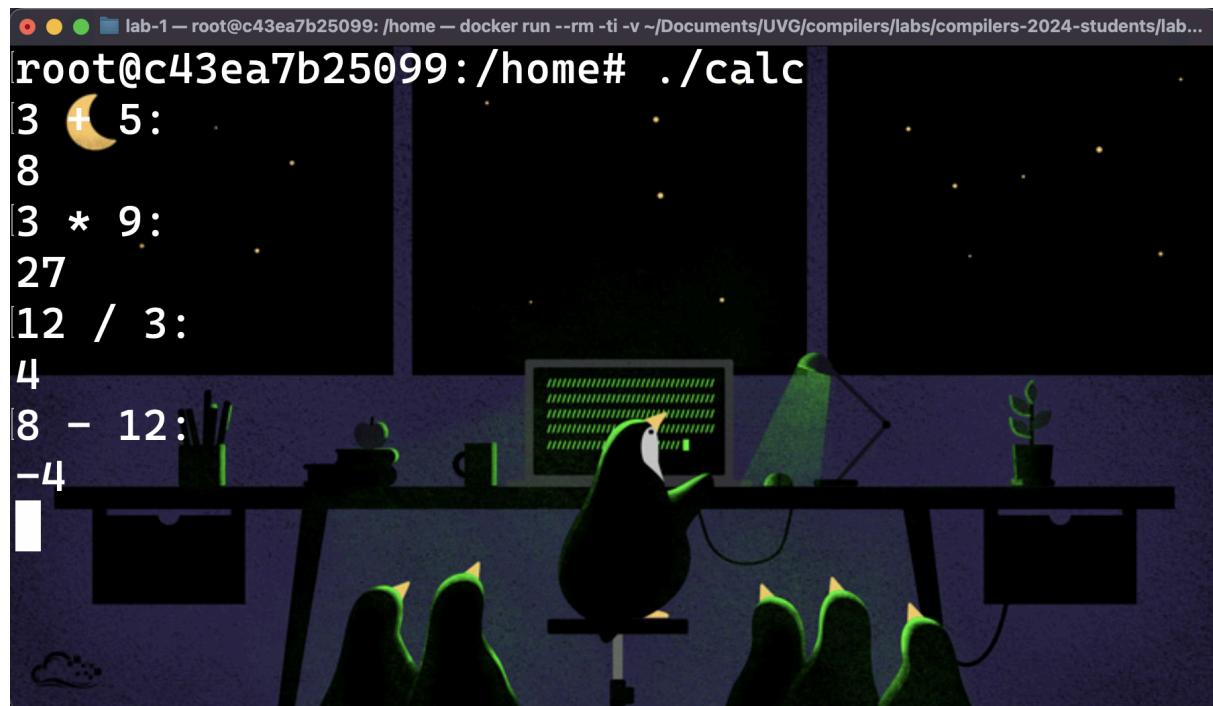
```
⇒ exporting to image
⇒ ⇒ exporting layers
⇒ ⇒ writing image sha256:c8d4195244a1532f41929626bbd9e86cae2d531acc3a02fde66ca7cc134a4ecc
⇒ ⇒ naming to docker.io/library/lab1-image
root@c43ea7b25099:/home# sh buildLanguage.sh
root@c43ea7b25099:/home#
```

Asignación de una variable

```
lab-1 — root@c43ea7b25099: /home — docker run --rm -ti -v ~/Documents/UVG/compilers/labs/compilers-2024-students/lab-1:/home lab1-image — 172x41
root@c43ea7b25099:/home# ./calc
y = 3
y:
Assign y = 3
3
```

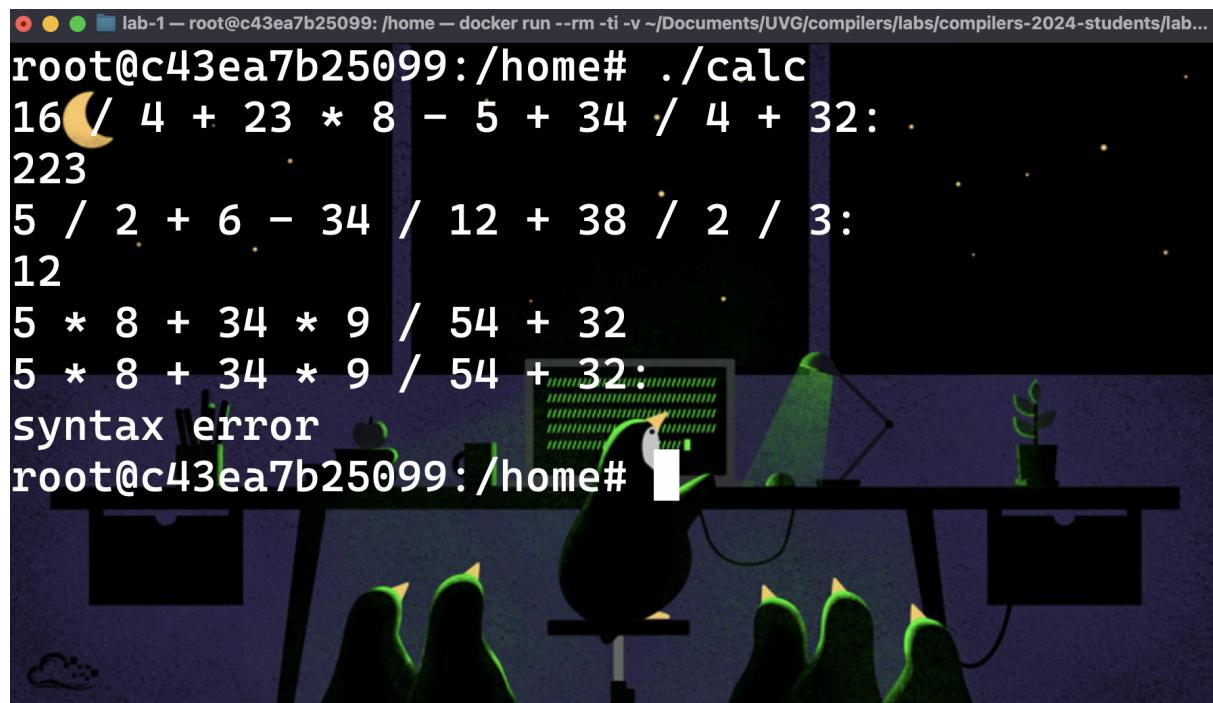
Operaciones básicas

```
root@c43ea7b25099:/home# ./calc  
3 + 5:  
8  
3 * 9:  
27  
12 / 3:  
4  
8 - 12:  
-4
```



Operaciones complejas

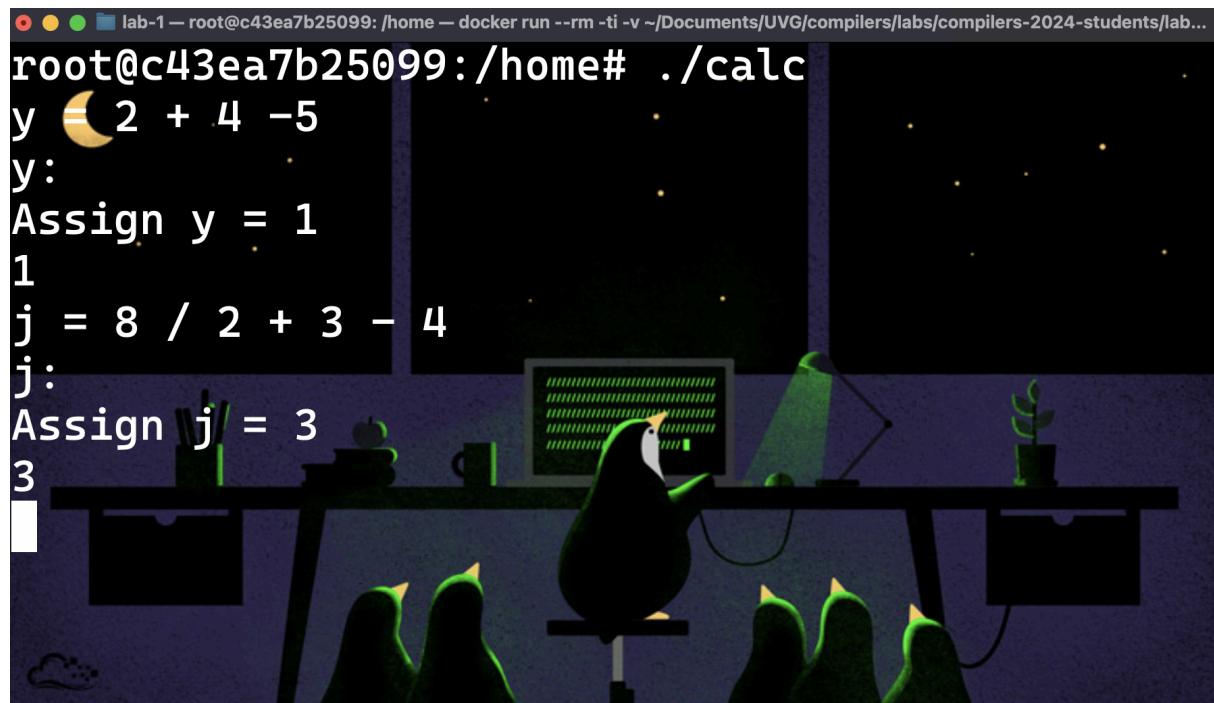
```
root@c43ea7b25099:/home# ./calc  
16 / 4 + 23 * 8 - 5 + 34 / 4 + 32:  
223  
5 / 2 + 6 - 34 / 12 + 38 / 2 / 3:  
12  
5 * 8 + 34 * 9 / 54 + 32  
5 * 8 + 34 * 9 / 54 + 32:  
syntax error  
root@c43ea7b25099:/home#
```



Las operaciones complejas fueron realizadas correctamente, sin embargo para algunas operaciones con divisiones más sofisticadas, da errores.

Asignación de expresiones aritméticas

```
root@c43ea7b25099:/home# ./calc
y = 2 + 4 - 5
y:
Assign y = 1
1
j = 8 / 2 + 3 - 4
j:
Assign j = 3
3
```



No hay problemas con asignarle a una variable el resultado de una operación aritmética.

Manejo de errores para tokens inválidos

```
root@37731d6f6b38:/home# ./calc
y @ 3
Error: Undefined variable: y
```

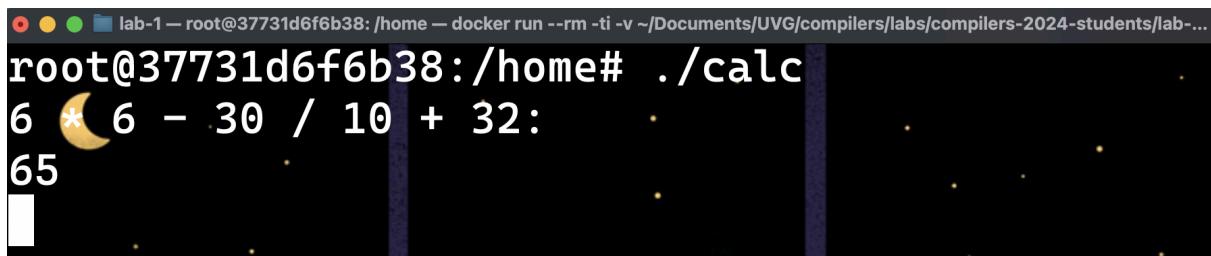
Pruebas de procedencia

No. 1

```
root@37731d6f6b38:/home# ./calc
4 + 7 * 3 - 1 / 2:
25
```

Se puede observar que la respuesta es 24.5 sin embargo, supongo que redondea la respuesta hacia el número entero más cercano.

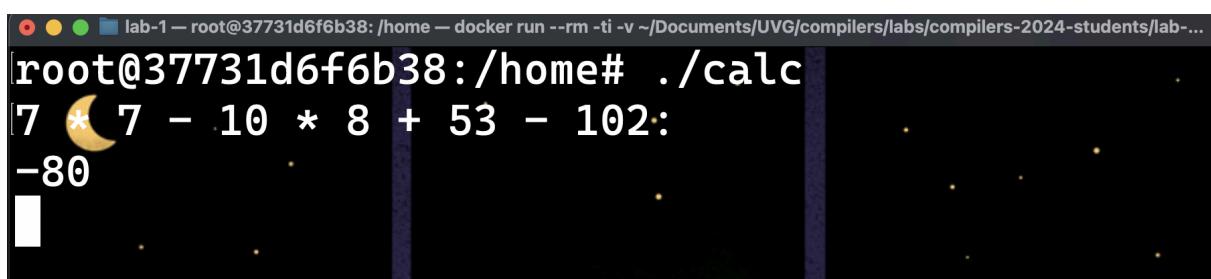
No. 2



```
root@37731d6f6b38:/home# ./calc
6 * 6 - 30 / 10 + 32:
65
```

Se puede observar que actúa de manera esperada, realizando primero las operaciones de mayor procedencia.

No. 3



```
root@37731d6f6b38:/home# ./calc
7 * 7 - 10 * 8 + 53 - 102:
-80
```

Se puede ver que también ejecuta las operaciones de manera esperada.

El árbol sintáctico no debería verse afectado por lo menos con estas operaciones o casos similares.