Uso de funções e pilha em Assembly

- Vamos investigar o funcionamento de passagem de parâmetros em assembly.
- Vamos usar o Evans Debugger (EDB), Debugador gráfico para ambiente Gnu Linux, para analisar o programa ao lado.
- Para cada execução, usamos F7 (step into), para executar cada linha.
- Observe os valores dos registradores RIP, RBP, RSP. Observe que o resultado da soma de 4 e 5 ficará no acumulador RAX

```
Seja o seguinte programa:
section .text

global _start

_start:

push 4              ;empilha 1o argumento, -8 bytes
push 5              ;empilha 2o argumento, -8 bytes
call soma           ;empilha IP de retorno e  chama a função soma
mov rcx,rax
;exit(0)
mov eax,1
mov ebx,0
int 0x80

soma:
    ;ao chegar aqui já guardou o RIP na pilha, -8 bytes
    push rbp        ;salva rpb,  -8 bytes
    ;usa-se o bp para guardar o valor de sp no stack frame
   ;o bp é a base para aceder os argumentos da pilha
    mov rbp,rsp
    mov rax,[rbp+24]    ;1o arg
    add rax,[rbp+16]    ;2o arg
  ;caso precisasse o sp pode ser modificado á vontade
   mov rsp,rbp
    pop rbp             ;recupera bp antigo
    ret                 ;salta para o
```

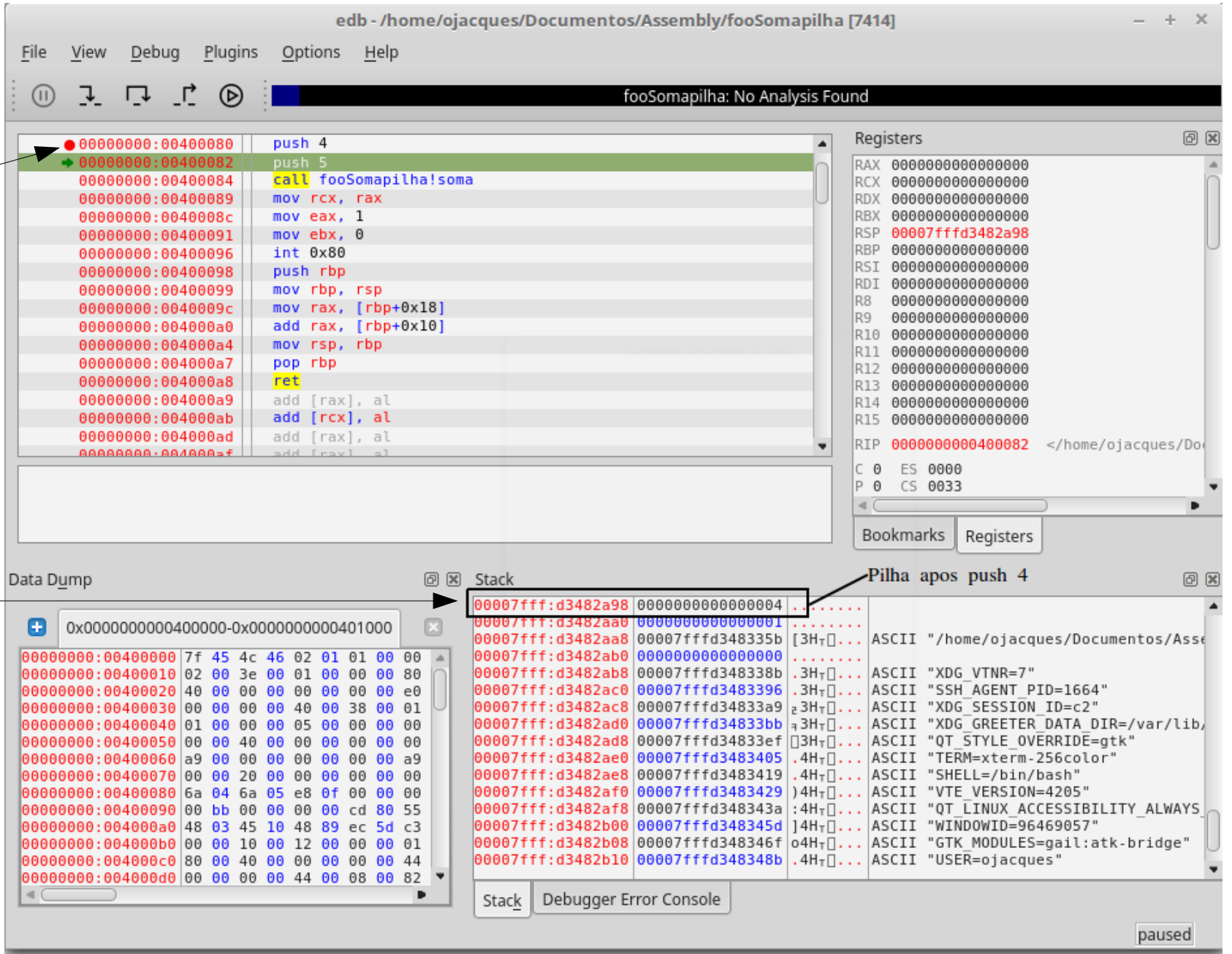
Uso de funções e pilha

Posição da pilha, após empilhar 4 (push 4).

RSP, que apontava para o endereço 0xd3482aa0 é diminuido de 8 bytes. Ou seja, RSP=RSP-8

Veja em Stack, o valor 4 no endereço

0xd3482a98, atual endereço de RSP.



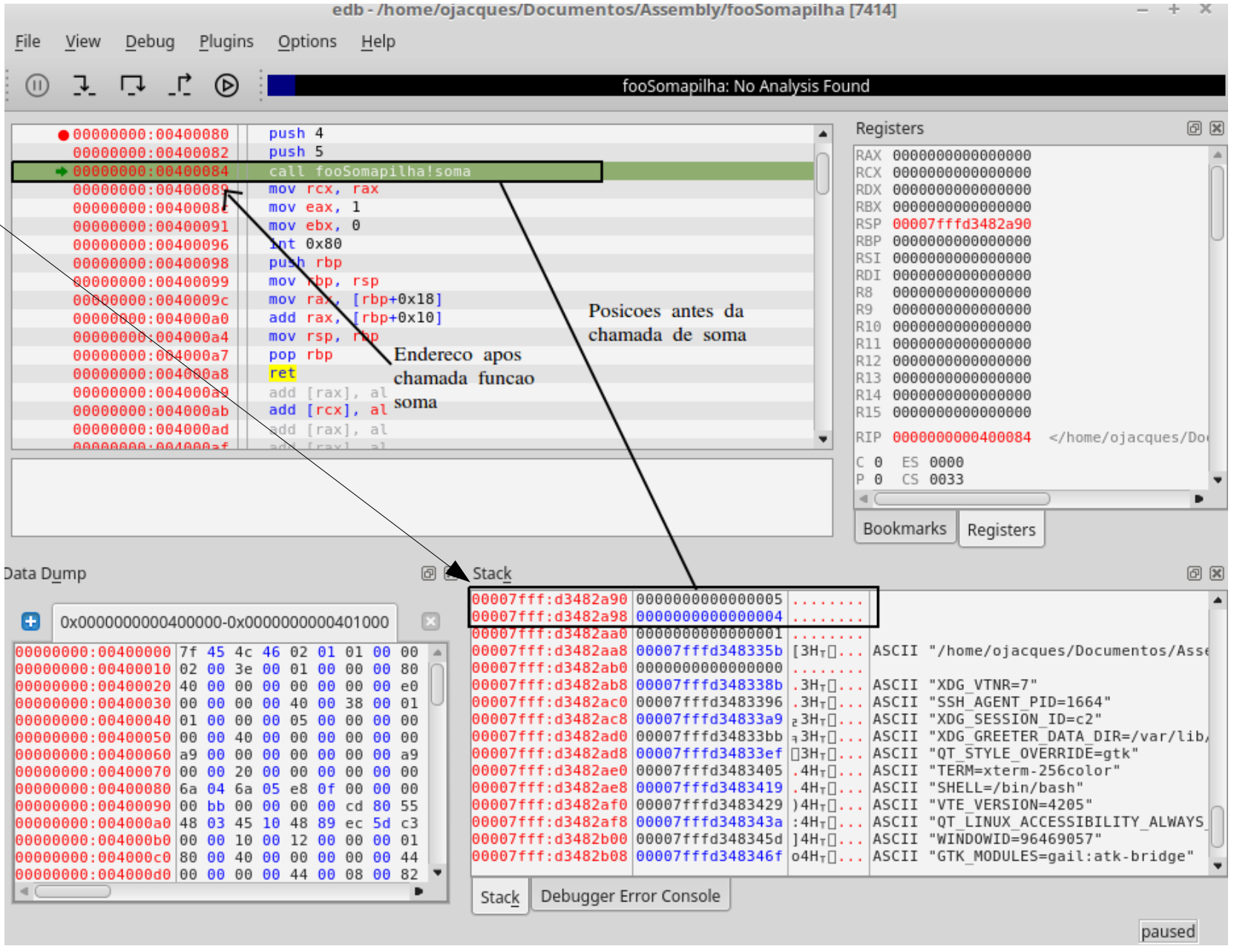
Uso de funções e pilha

Posição da pilha, após empilhar 5.

Veja em Stack, que 4 esta no endereço 0xd3482a98,e 5 está na posição 0xd3482a90.

Ou seja, o endereço decresceu 8 bytes (64 bits).

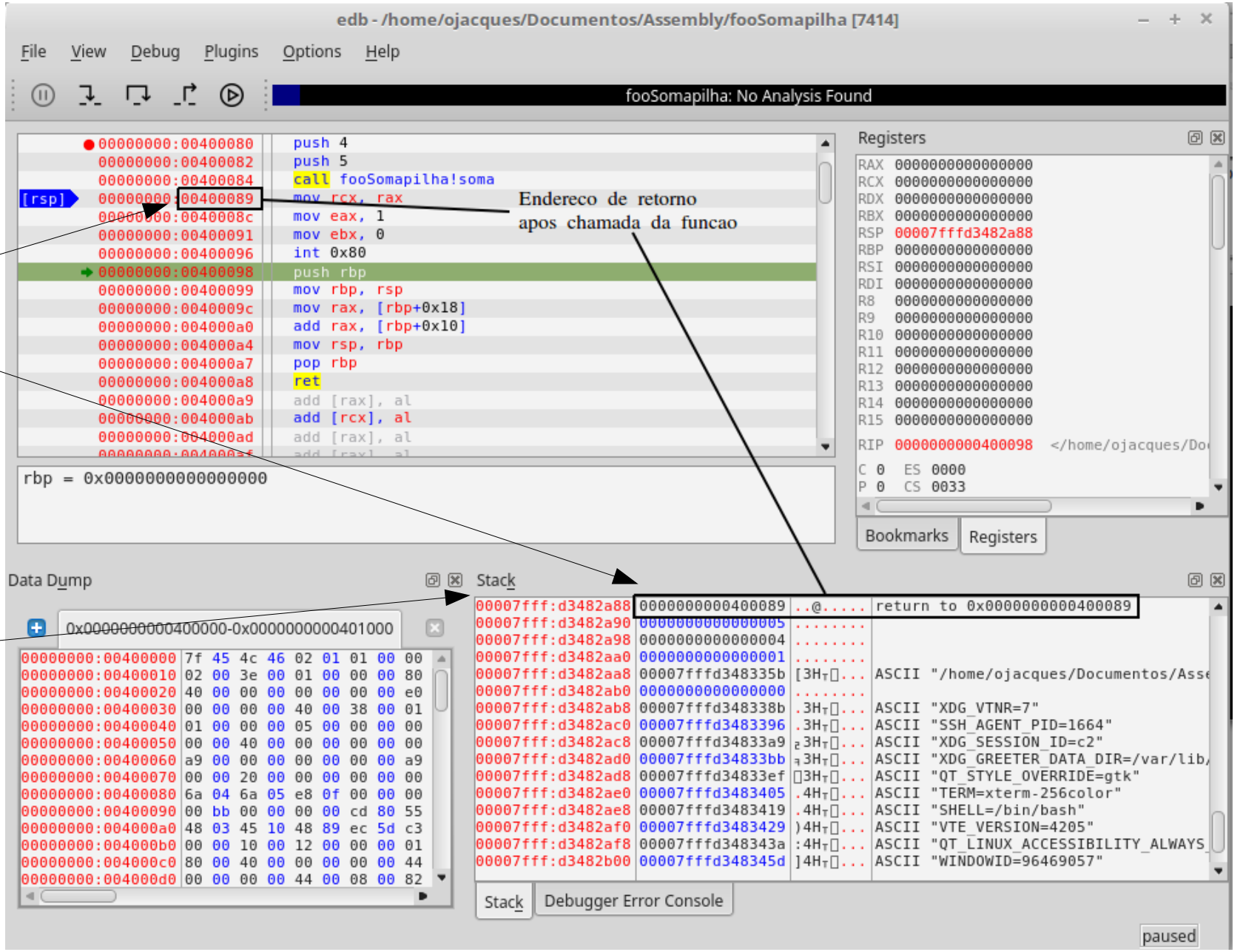
A pilha decresce ou cresce para baixo.



Uso de funções e pilha

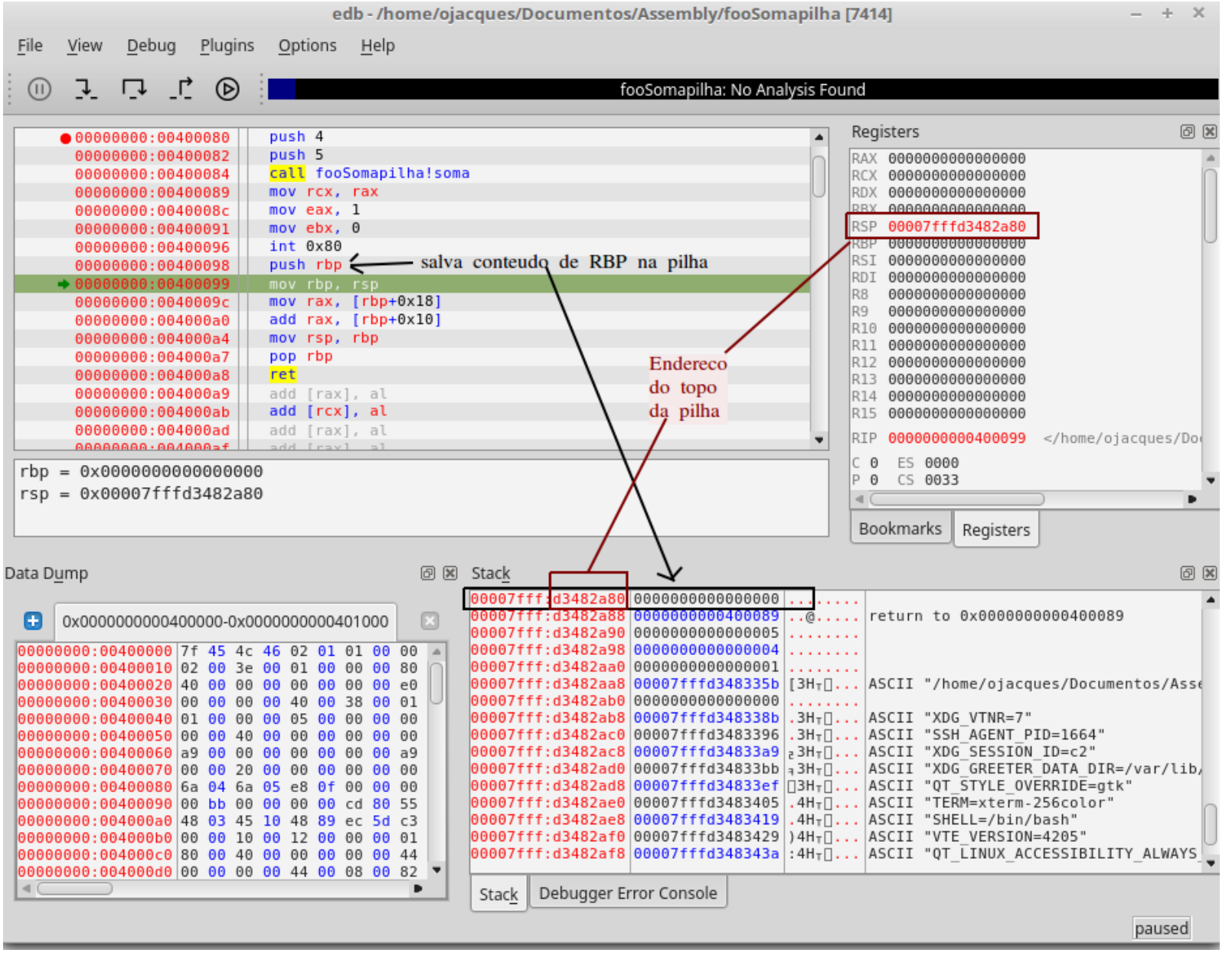
Posição após a chamada da função soma. Repare que fica guardado no topo da pilha o valor 0x400089. Esse valor é o endereço da próxima instrução após a execução da função soma.

RSP aponta para o topo da pilha (TOS). O topo é o endereço 0xd3482a88. Observe que 0xd3482a90 – 0x8 =0xd3482a88, pois estamos em notação hexadecimal.



Uso de funções e pilha

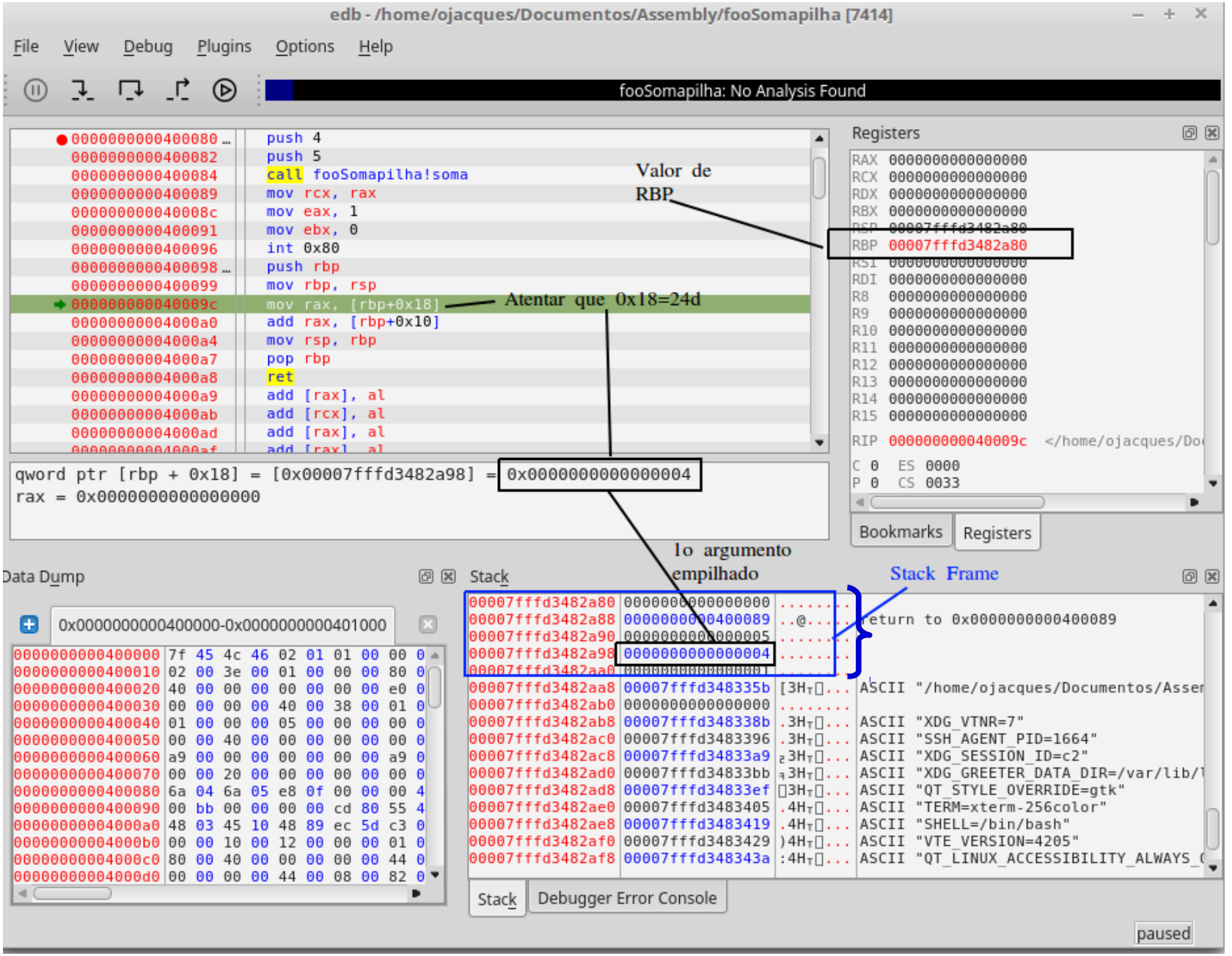
Dentro da função, a pilha ainda decresce com o salvamento de RBP, registrador de base. RSP agora aponta para 0xd3482a80, diminuindo mais 8 bytes. Na linha 0x400099 será atribuido a RBP o valor do topo da pilha atual (TOS), pela instrução mov rbp, rsp. O valor anterior de RBP foi salvo com push rbp.



Uso de funções e pilha

A instrução atual, mov rax, [rbp+0x18] é o mesmo que mov rax, [rbp+24]. Essa instrução acessa o primeiro valor empilhado, 4. Para somar 4 ao valor 5, fazemos add rax, [rbp+0x10], que é o mesmo que add rax, [rbp+16].

A região da pilha desde os parâmetros, endereço de retorno, valor de RBP salvo e varíaveis que possam ser empilhadas é denominada Stack Frame.



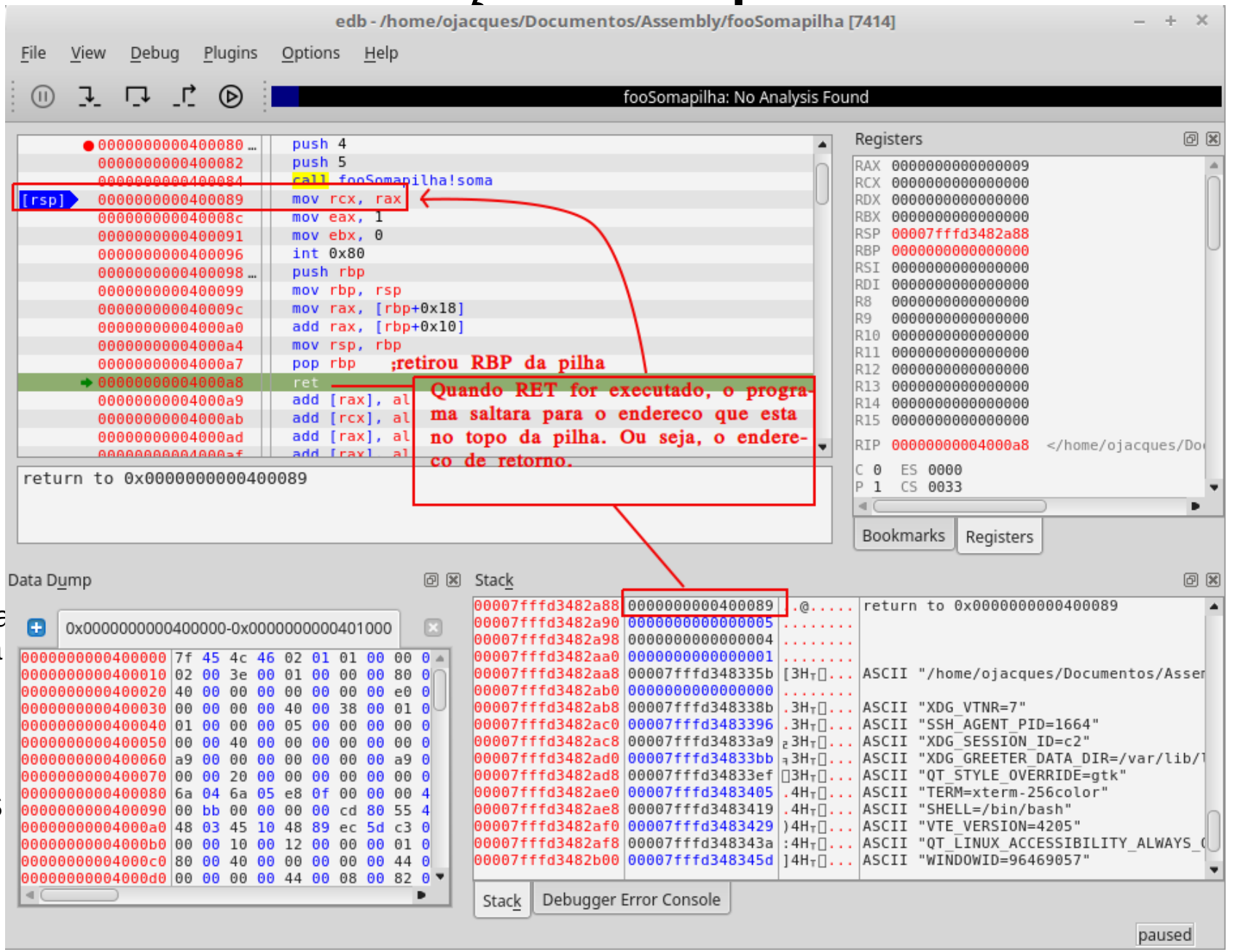
Uso de funções e pilha

A instrução mov rsp,rbp faz com que RSP volte a apontar para o local onde estava armazenado o valor antigo de RBP.

Após isso, pop rbp recupera o valor antigo de RBP

Quando a instrução ret for executada, ela RETornará para a próxima linha após a chamada da função soma.

Essa linha está armazenada no TOS apontada por RSP (0x400089).



Uso de funções e pilha

Após voltar da função soma, executa-se mov rcx,rax, salvando o resultado da soma em RCX. É sempre recomendado que os resultados venham através do acumulador (RAX).

Veja que o valor de RAX é 9.