## Curso de Assembly x86 64 – Bits

Prof. Ronaldo Luiz Alonso UFMT

## Instrução call

- call <endereço-da-função>
  - Coloca na pilha o endereço imediatamente após o endereço da função.
  - Desvia para o endereço da função (faz o ponteiro de instrução receber o valor do endereço da função).

- Retira da pilha o endereço que está no topo.
- Desvia para este endereço (faz o ponteiro de instrução receber esse endereço.

### Pilha

- A pilha possui dois ponteiros
  - SP: Aponta para o topo da pilha
  - BP: aponta para a base da pilha
- A instrução push coloca um valor na pilha (incrementa ou decrementa sp e coloca um valor na posição apontada por sp)
- A instrução pop retira um valor da pilha (incrementa ou decrementa sp e coloca um valor na posição apontada por sp)
- A instrução call coloca um endereço na pilha (endereço para onde a função chamada deve retornar).
- A instrução ret retira um endereço da pilha (endereço para onde a função chamada deve retornar).

## Convenção de chamada de funções

- Chamador empilhar os parâmetros
- Chamador chama função
- Função chamada cria as variáveis locais
- Função chamada destrói as variáveis locais
- Função chamada coloca o endereço de retorno na pilha
- Chamador retira o valor retornado da pilha colocado pela função chamada.

# Assembly gerado pelo gcc(AT&T) e correspondente no NASM (gnu)

#### AT&T:

- Constantes começam com \$
- · Registradores começam com %
- · <instrução> destino, fonte
- · Labels começam com .

#### NASM:

- · Constantes são definidas com equ (equate).
- · Registradores não precisam de %
- · <instrução> fonte, destino
- · Labels são identificados com :

```
main:
                            .LFB0:
                                .cfi startproc
                                                                main:
                                pushq
                                         %rbp
                                                                    push
                                                                             rbp
                                .cfi def cfa offset 16
                                                                     mov
                                                                             rbp, rsp
                                 .cfi offset 6, -16
                                                                    sub
                                                                             rsp, 16
int main(void) {
                                         %rsp, %rbp
                                movq
                                                                             [rbp-12], 2
                                                                     mov
  int a, b, c;
                                .cfi def cfa register 6
                                                                             [rbp-8], 3
                                                                     mov
  a = 2;
                                subg
                                         $16, %rsp
                                                                             edx, [rbp-8]
                                                                     mov
  b = 3;
                                movl
                                         $2, -12(%rbp)
                                                                             eax,[rbp-12]
                                                                    mov
  c = func1(a,b);
                                movl
                                         $3, -8(%rbp)
                                                                             esi, edx
                                                                    mov
  return c;
                                         -8(%rbp), %edx
                                movl
                                                                             edi, eax
                                                                    mov
                                movl
                                         -12(%rbp), %eax
                                                                     call
                                                                             func1
                                         %edx, %esi
                                movl
                                                                             [rbp-4], eax
                                                                     mov
                                movl
                                         %eax, %edi
                                                                             eax, [rbp-4]
                                                                     mov
                                call func1
                                                                     leave
                                movl
                                         %eax, -4(%rbp)
                                                                     ret
                                         -4(%rbp), %eax
                                movl
                                leave
                                .cfi def cfa 7, 8
                                ret
                                .cfi endproc
```

movq

movl

movl

movl

movl

movl

movl

call

main:

%rbp

push mov

sub

mov

mov

mov

mov

mov

mov

call

main:

rbp, rsp rsp, 16 [rbp-12], 2

[rbp-8], 3

edx, [rbp-8]

eax,[rbp-12]

rbp

a = 2; b = 3;c = func1(a,b);return c;

int main(void) {

int a, b, c;

subq movl movl

\$16, %rsp \$2, -12(%rbp) \$3, -8(%rbp) -8(%rbp), %edx -12(%rbp), %eax %edx, %esi

func1

%rsp, %rbp

-4(%rbp), %eax

%eax, %edi %eax, -4(%rbp)

mov mov leave ret

esi, edx

edi, eax

func1 [rbp-4], eax

eax, [rbp-4]

leave

movq

subq

main:

%rbp

%rsp, %rbp

push mov sub mov

mov

mov

main:

rbp, rsp rsp, 16 [rbp-12], 2 [rbp-8], 3 edx, [rbp-8] eax,[rbp-12]

esi, edx

edi, eax

rbp

movl c = func1(a,b);movl movl movl movl movl call

int main(void) {

int a, b, c;

a = 2;

b = 3;

return c;

func1

\$16, %rsp \$2, -12(%rbp) \$3, -8(%rbp) -8(%rbp), %edx -12(%rbp), %eax %edx, %esi %eax, %edi %eax, -4(%rbp) -4(%rbp), %eax

mov mov mov call mov mov leave ret

func1 [rbp-4], eax eax, [rbp-4]

leave ret

movl

movl

movq

subq

movl

movl

movl

int main(void) {

int a, b, c;

c = func1(a,b);

a = 2;

b = 3;

return c;

main:

%rbp

%rsp, %rbp

main:

push

mov

sub

mov

mov

mov

mov

mov

mov

leave

ret

rbp

rbp, rsp rsp, 16 [rbp-12], 2 [rbp-8], 3 edx, [rbp-8]

movl movl movl call movl movl leave

%edx, %esi %eax, %edi

func1

\$2, -12(%rbp) \$3, -8(%rbp) -8(%rbp), %edx -12(%rbp), %eax

%eax, -4(%rbp)

-4(%rbp), %eax

\$16, %rsp

call mov mov

esi, edx edi, eax func1

eax,[rbp-12]

[rbp-4], eax eax, [rbp-4]

movq

subq

movl

movl

main:

%rbp

%rsp, %rbp

\$16, %rsp

\$2, -12(%rbp)

rbp, rsp mov sub rsp, 16 mov mov

rbp

push

mov

mov

ret

[rbp-12], 2 [rbp-8], 3 edx, [rbp-8] eax,[rbp-12]

movl movl movl movl call movl

\$3, -8(%rbp) -8(%rbp), %edx -12(%rbp), %eax %edx, %esi %eax, %edi func1

%eax, -4(%rbp)

-4(%rbp), %eax

mov mov

main:

call mov mov leave

esi, edx edi, eax func1 [rbp-4], eax eax, [rbp-4]

movl leave

int main(void) {

int a, b, c;

c = func1(a,b);

a = 2;

b = 3;

return c;

main:

movq

subq

movl

call

movl

movl

leave

ret

%rbp

%rsp, %rbp

\$16, %rsp

func1

\$2, -12(%rbp)

mov sub

push

mov

mov

mov

mov

leave

main:

rbp, rsp rsp, 16

rbp

[rbp-12], 2 [rbp-8], 3 edx, [rbp-8]

eax,[rbp-12]

b = 3;c = func1(a,b);return c;

a = 2;

int main(void) {

int a, b, c;

movl movl movl movl movl

\$3, -8(%rbp) -8(%rbp), %edx -12(%rbp), %eax %edx, %esi %eax, %edi

mov mov call mov mov

esi, edx edi, eax func1 [rbp-4], eax

eax, [rbp-4]

%eax, -4(%rbp) -4(%rbp), %eax

main:

int main(void) {
 int a, b, c;

c = func1(a,b);

a = 2; b= 3;

return c;

	main:				
pushq	%rbp	push	rbp		
		mov	rbp, rsp		
mova	06ren 06rhn	sub	rsp, 16		
movq	%rsp, %rbp	mov	[rbp-12], 2		
subg	\$16, %rsp	mov	[rbp-8], 3		
movl	\$2, -12(%rbp)	mov	edx, [rbp-8]		
movl	\$3, -8(%rbp)	mov	eax,[rbp-12]		
movl	-8(%rbp), %edx	mov	esi, edx		
movl	-12(%rbp), %eax	mov call	edi, eax func1		
movl	%edx, %esi	mov	[rbp-4], eax		
movl	%eax, %edi	mov	eax, [rbp-4]		
call	func1	leave	cax, [lbp +]		
movl	%eax, -4(%rbp)	ret			
movl	-4(%rbp), %eax	100			
leave					
ret					

movq

subq

movl

movl

movl

movl

movl

movl

movl

movl

leave

ret

call

int main(void) {

int a, b, c;

c = func1(a,b);

a = 2;

b = 3;

return c;

main:

%rbp

%rsp, %rbp

\$16, %rsp

\$2, -12(%rbp)

-8(%rbp), %edx

%eax, -4(%rbp)

-4(%rbp), %eax

-12(%rbp), %eax

\$3, -8(%rbp)

%edx, %esi

%eax, %edi

func1

main:

push

mov sub

mov

mov

mov

mov

mov

mov

call

mov

mov

leave

ret

rsp, 16 [rbp-12], 2

[rbp-8], 3 edx, [rbp-8] eax,[rbp-12]

rbp

rbp, rsp

esi, edx

edi, eax func1

[rbp-4], eax

eax, [rbp-4]

movq

subq

movl

movl

movl

movl

movl

movl

movl

movl

leave

ret

call

int main(void) {

int a, b, c;

c = func1(a,b);

a = 2;

b = 3;

return c;

main:

%rbp

%rsp, %rbp

\$16, %rsp

\$2, -12(%rbp)

-8(%rbp), %edx

%eax, -4(%rbp)

-4(%rbp), %eax

-12(%rbp), %eax

\$3, -8(%rbp)

%edx, %esi

%eax, %edi

func1

push mov sub mov

mov

mov

mov

mov

mov

call

mov

mov

leave

ret

main:

rbp, rsp rsp, 16

rbp

[rbp-12], 2 [rbp-8], 3 edx, [rbp-8] eax,[rbp-12]

esi, edx edi, eax

func1

[rbp-4], eax

eax, [rbp-4]

int main(void) {

int a, b, c;

c = func1(a,b);

a = 2;

b = 3;

return c;

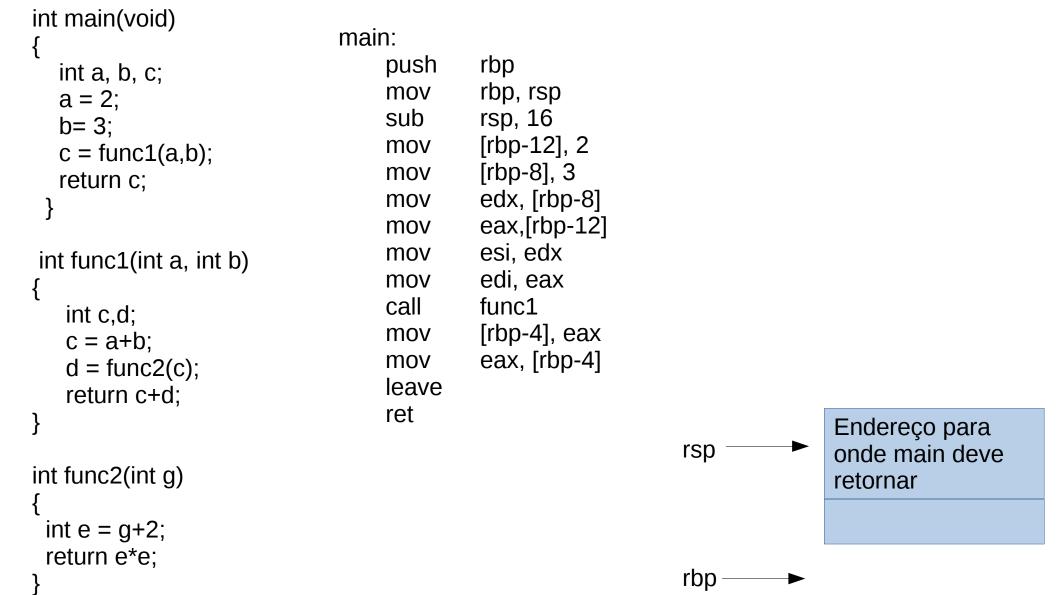
main:

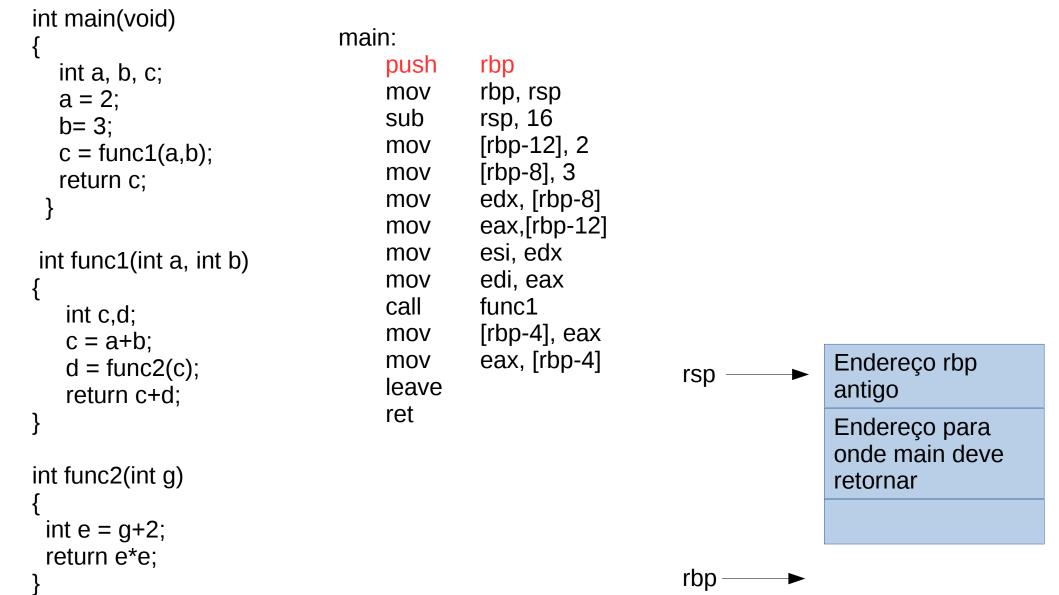
pushq	%rbp	mai
movq	%rsp, %rbp	
subq movl movl movl movl call movl	\$16, %rsp \$2, -12(%rbp) \$3, -8(%rbp) -8(%rbp), %edx -12(%rbp), %eax %edx, %esi %eax, %edi func1 %eax, -4(%rbp) -4(%rbp), %eax	
leave ret		

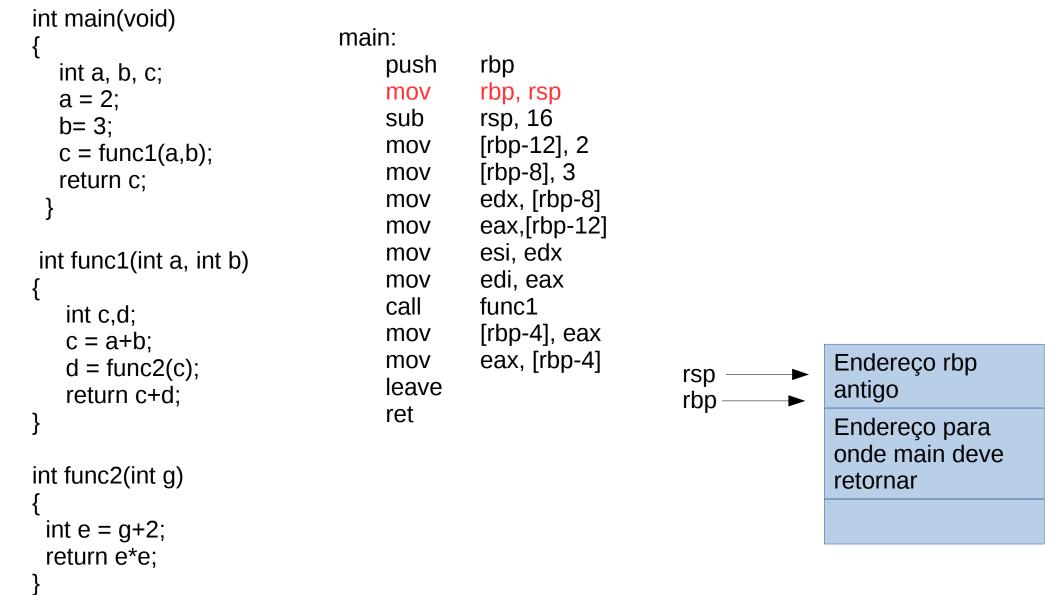
in: push rbp rbp, rsp mov rsp, 16 sub [rbp-12], 2 mov [rbp-8], 3 mov edx, [rbp-8] mov eax,[rbp-12] mov esi, edx mov edi, eax mov call func1 [rbp-4], eax mov eax, [rbp-4] mov leave ret

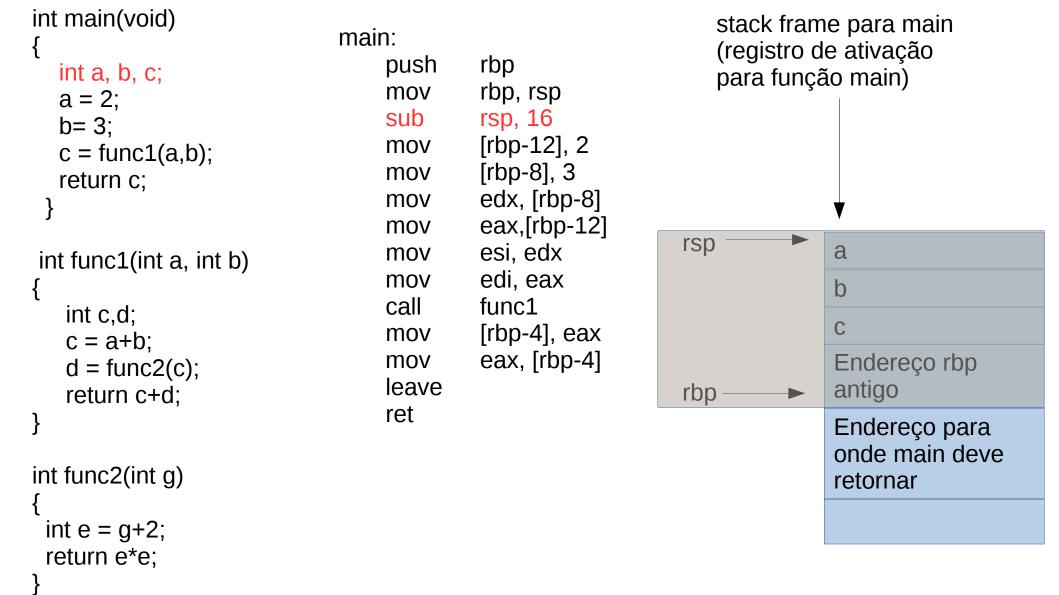
main:

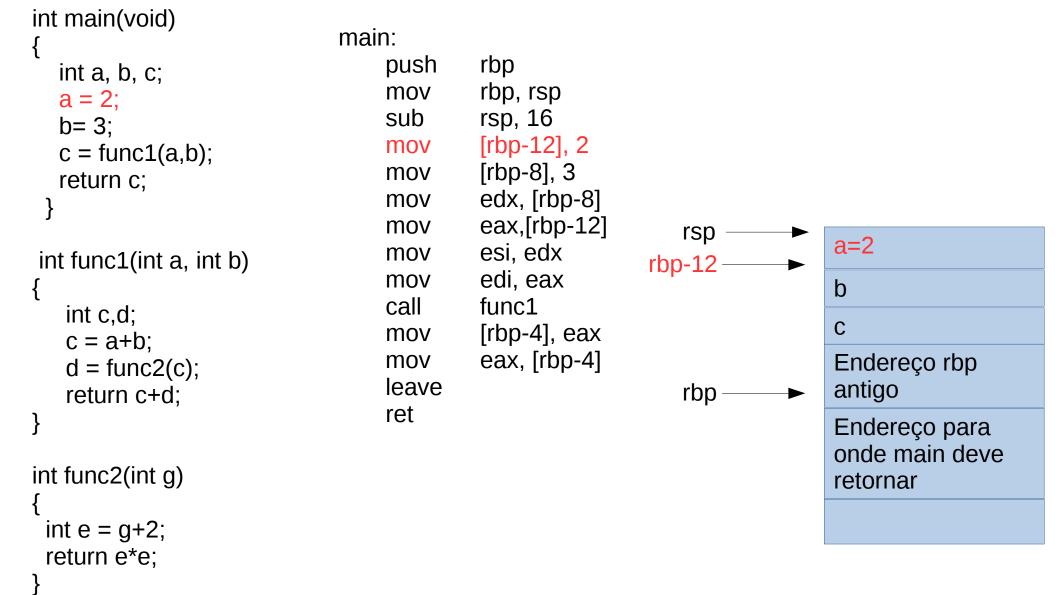
	pushq	%rbp	main: push mov	rbp rbp, rsp
<pre>int main(void) {    int a, b, c;    a = 2;    b= 3;    c = func1(a,b);    return c; }</pre>	movq subq movl movl movl movl call movl movl leave	%rsp, %rbp \$16, %rsp \$2, -12(%rbp) \$3, -8(%rbp) -8(%rbp), %edx -12(%rbp), %eax %edx, %esi %eax, %edi func1 %eax, -4(%rbp) -4(%rbp), %eax	sub mov mov mov mov call mov mov leave ret	rsp, 16 [rbp-12], 2 [rbp-8], 3 edx, [rbp-8] eax,[rbp-12] esi, edx edi, eax func1 [rbp-4], eax eax, [rbp-4]

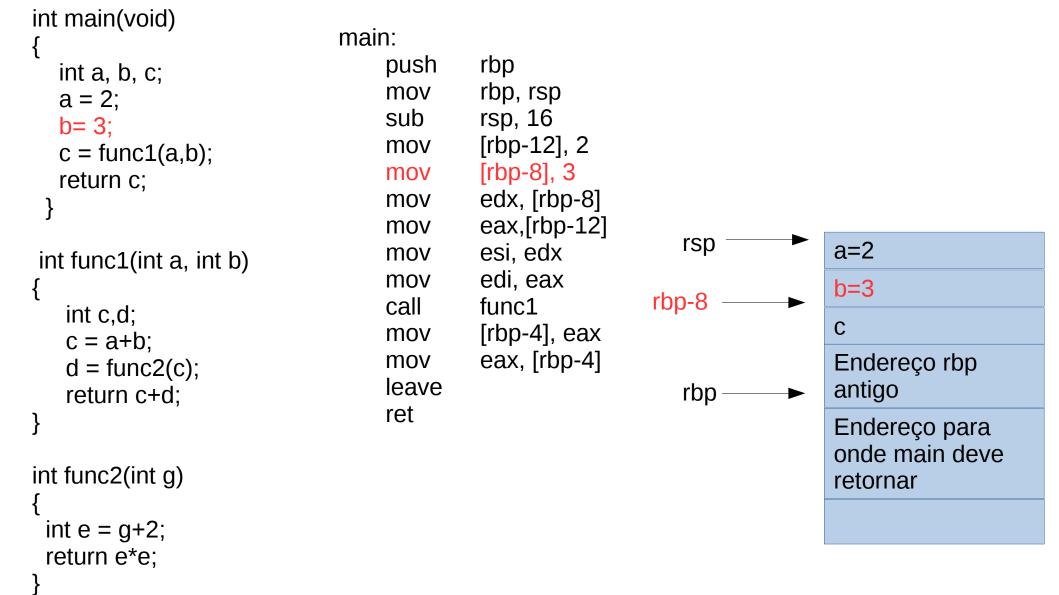


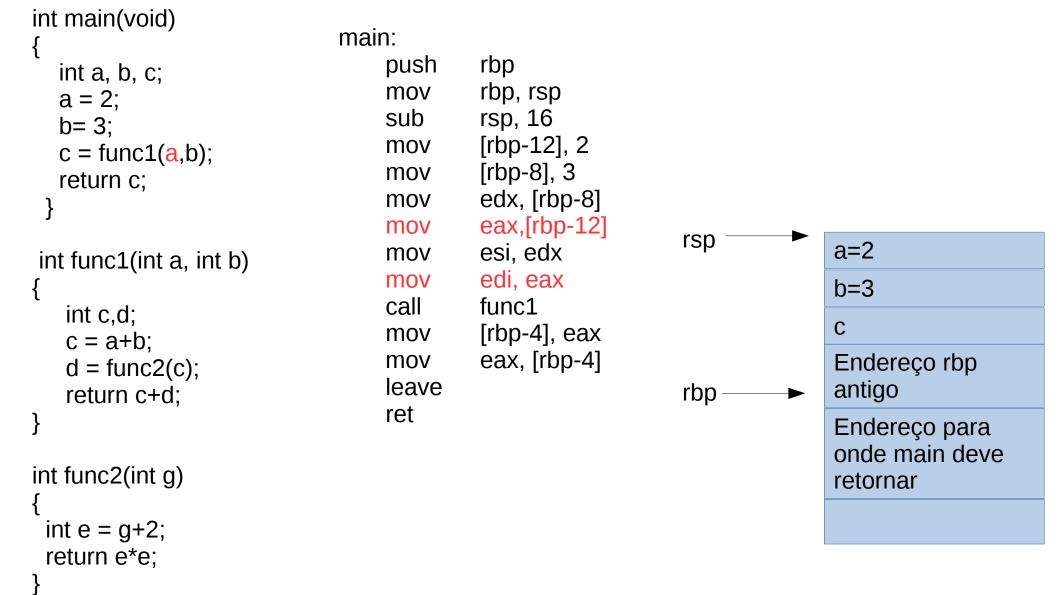


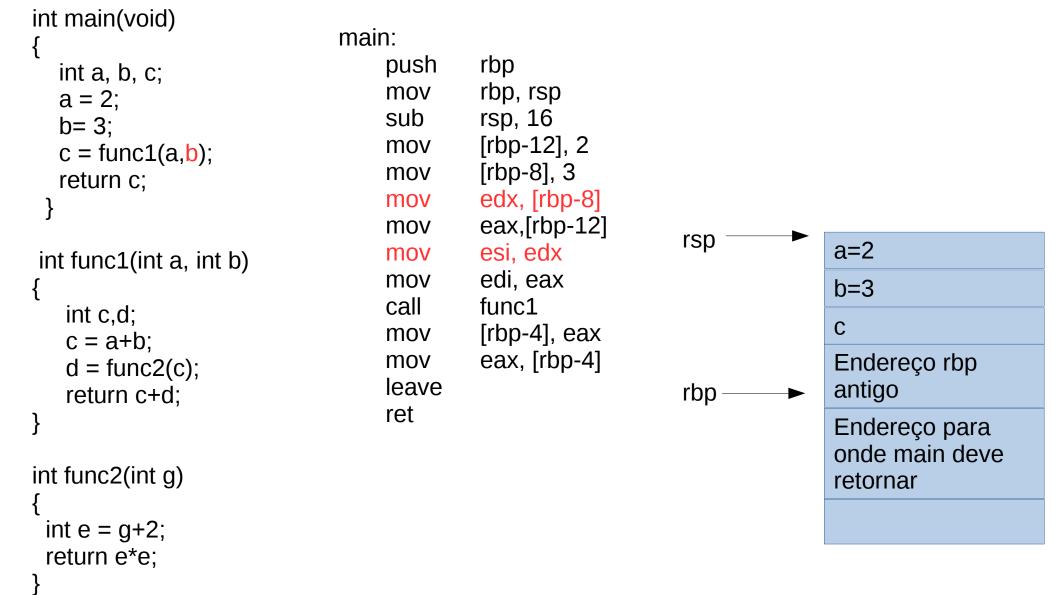


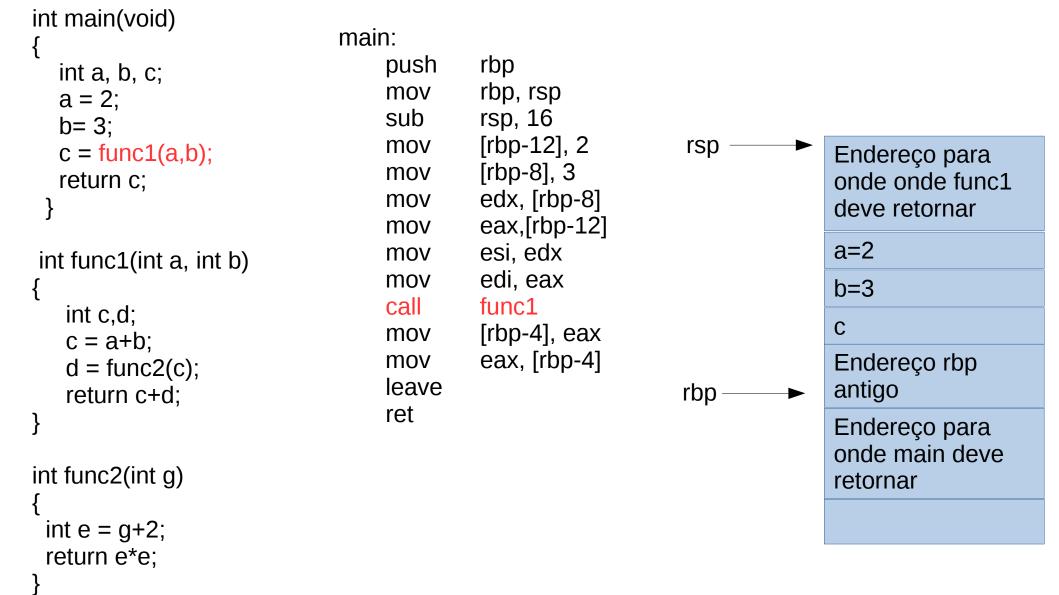












```
func1:
                             .LFB1:
                                  .cfi startproc
                                            %rbp
                                  pushq
                                  .cfi def cfa offset 16
                                  .cfi offset 6, -16
                                            %rsp, %rbp
                                  movq
int func1(int a, int b)
                                  .cfi def cfa register 6
                                  subg $32, %rsp
                                  movl %edi, -20(%rbp)
  int c,d;
                                  movl %esi, -24(%rbp)
  c = a+b:
                                  movl -20(%rbp), %edx
  d = func2(c);
                                  movl -24(%rbp), %eax
  return c+d;
                                  addl %edx, %eax
                                  movl %eax, -8(%rbp)
                                  movl -8(%rbp), %eax
                                  movl %eax, %edi
                                  call func2
                                  movl %eax, -4(%rbp)
                                  movI -8(%rbp), %edx
                                  movl -4(%rbp), %eax
                                  addl %edx, %eax
                                  leave
                                  .cfi def cfa 7, 8
                                  ret
                                  .cfi endproc
```

```
push
          rbp
mov
          rbp, rsp,
sub rsp,32
movl [rbp-20],edi
movl [rbp-24],esi
movl edx, [rbp-20]
movl eax, [rbp-24]
addl eax, edx
movl [rbp-8], eax
movl eax,[rbp-8]
movl edi, eax
call func2
movl [rbp-4], eax
movl edx, [rbp-8]
movl eax, [rbp-4]
addl eax, edx
leave
ret
```

func1:

```
func1:
                                                                        func1:
                                                                             push
                                                                                       rbp
                                                                             mov
                                                                                       rbp, rsp,
                                  pushq
                                            %rbp
                                                                             sub rsp,32
                                                                             movl [rbp-20],edi
                                            %rsp, %rbp
                                                                             movl [rbp-24],esi
                                  movq
int func1(int a, int b)
                                                                             movl edx, [rbp-20]
                                  subq $32, %rsp
                                                                             movl eax, [rbp-24]
                                  movl %edi, -20(%rbp)
                                                                             addl eax, edx
  int c,d;
                                  movl %esi, -24(%rbp)
                                                                             movl [rbp-8], eax
  c = a+b:
                                  movl -20(%rbp), %edx
                                                                             movl eax,[rbp-8]
  d = func2(c);
                                  movl -24(%rbp), %eax
                                                                             movl edi, eax
  return c+d;
                                  addl %edx, %eax
                                                                             call func2
                                  movl %eax, -8(%rbp)
                                                                             movl [rbp-4], eax
                                  movl -8(%rbp), %eax
                                                                             movl edx, [rbp-8]
                                  movl %eax, %edi
                                                                             movl eax, [rbp-4]
                                  call func2
                                                                             addl eax, edx
                                  movl %eax, -4(%rbp)
                                                                             leave
                                  movI -8(%rbp), %edx
                                  movl -4(%rbp), %eax
                                                                             ret
                                  addl %edx, %eax
                                  leave
                                  ret
```

```
func1:
                                                                        func1:
                                                                             push
                                                                                       rbp
                                                                             mov
                                                                                       rbp, rsp,
                                  pushq
                                            %rbp
                                                                             sub rsp,32
                                                                             movl [rbp-20],edi
                                            %rsp, %rbp
                                                                             movl [rbp-24],esi
                                  movq
int func1(int a, int b)
                                                                             movl edx, [rbp-20]
                                  subq $32, %rsp
                                                                             movl eax, [rbp-24]
                                  movl %edi, -20(%rbp)
                                                                             addl eax, edx
                                  movl %esi, -24(%rbp)
                                                                             movl [rbp-8], eax
                                  movl -20(%rbp), %edx
                                                                             movl eax,[rbp-8]
                                  movl -24(%rbp), %eax
                                                                             movl edi, eax
                                  addl %edx, %eax
                                                                             call func2
                                  movl %eax, -8(%rbp)
                                                                             movl [rbp-4], eax
                                  movl -8(%rbp), %eax
                                                                             movl edx, [rbp-8]
                                  movl %eax, %edi
                                                                             movl eax, [rbp-4]
                                  call func2
                                                                             addl eax, edx
                                  movl %eax, -4(%rbp)
                                                                             leave
                                  movI -8(%rbp), %edx
                                  movl -4(%rbp), %eax
                                                                             ret
                                  addl %edx, %eax
                                  leave
                                  ret
```

int c,d;

c = a+b:

d = func2(c);

return c+d;

```
func1:
                                           func1:
                                                push
                                                          rbp
                                               mov
                                                          rbp, rsp,
     pushq
               %rbp
                                                sub rsp,32
                                                movl [rbp-20],edi
                                                movl [rbp-24],esi
               %rsp, %rbp
     movq
                                                movl edx, [rbp-20]
     subq $32, %rsp
                                                movl eax, [rbp-24]
     movl %edi, -20(%rbp)
                                                addl eax, edx
     movl %esi, -24(%rbp)
                                                movl [rbp-8], eax
     movl -20(%rbp), %edx
                                                movl eax,[rbp-8]
     movl -24(%rbp), %eax
                                                movl edi, eax
     addl %edx, %eax
                                                call func2
     movl %eax, -8(%rbp)
                                                movl [rbp-4], eax
     movl -8(%rbp), %eax
                                                movl edx, [rbp-8]
     movl %eax, %edi
                                                movl eax, [rbp-4]
     call func2
                                                addl eax, edx
     movl %eax, -4(%rbp)
                                                leave
     movI -8(%rbp), %edx
     movl -4(%rbp), %eax
                                                ret
     addl %edx, %eax
     leave
     ret
```

int func1(int a, int b)

int c,d;

c = a+b:

d = func2(c);

return c+d;

```
Espaço para cópia dos
parâmetros e variáveis
locais.
32 = (6+2)*4
 int func1(int a, int b)
    int c,d;
    c = a+b:
    d = func2(c);
    return c+d;
```

```
func1:
                                         func1:
     pushq
              %rbp
              %rsp, %rbp
     movq
     subq $32, %rsp
     movl %edi, -20(%rbp)
     movl %esi, -24(%rbp)
     movl -20(%rbp), %edx
     movl -24(%rbp), %eax
     addl %edx, %eax
     movl %eax, -8(%rbp)
     movl -8(%rbp), %eax
     movl %eax, %edi
     call func2
     movl %eax, -4(%rbp)
     movI -8(%rbp), %edx
     movl -4(%rbp), %eax
     addl %edx, %eax
     leave
     ret
```

rbp push mov rbp, rsp, sub rsp,32 movl [rbp-20],edi movl [rbp-24],esi movl edx, [rbp-20] movl eax, [rbp-24] addl eax, edx movl [rbp-8], eax movl eax, [rbp-8] movl edi, eax call func2 movl [rbp-4], eax movl edx, [rbp-8] movl eax, [rbp-4] addl eax, edx leave ret

```
func1:
Salva cópia dos
parâmetros passados por
valor na pilha.
                                   pushq
                                            %rbp
Não tem código
correspondente.
                                            %rsp, %rbp
                                   movq
 int func1(int a, int b)
                                  subq $32, %rsp
                                   movl %edi, -20(%rbp)
    int c,d;
                                  movl %esi, -24(%rbp)
   c = a+b:
                                   movl -20(%rbp), %edx
   d = func2(c);
                                   movl -24(%rbp), %eax
    return c+d;
                                   addl %edx, %eax
                                   movl %eax, -8(%rbp)
                                   movl -8(%rbp), %eax
                                   movl %eax, %edi
                                   call func2
                                   movl %eax, -4(%rbp)
                                   movI -8(%rbp), %edx
                                   movl -4(%rbp), %eax
                                   addl %edx, %eax
                                   leave
                                   ret
```

```
rbp
push
mov
          rbp, rsp,
sub rsp,32
movl [rbp-20],edi
movl [rbp-24],esi
movl edx, [rbp-20]
movl eax, [rbp-24]
addl eax, edx
movl [rbp-8], eax
movl eax,[rbp-8]
movl edi, eax
call func2
movl [rbp-4], eax
movl edx, [rbp-8]
movl eax, [rbp-4]
addl eax, edx
leave
ret
```

func1:

```
func1:
                                                                        func1:
                                                                             push
                                                                                       rbp
                                                                             mov
                                                                                       rbp, rsp,
                                  pushq
                                            %rbp
                                                                             sub rsp,32
                                                                             movl [rbp-20],edi
                                            %rsp, %rbp
                                                                             movl [rbp-24],esi
                                  movq
int func1(int a, int b)
                                                                             movl edx, [rbp-20]
                                  subq $32, %rsp
                                                                             movl eax, [rbp-24]
                                  movl %edi, -20(%rbp)
                                                                             addl eax, edx
  int c,d;
                                  movl %esi, -24(%rbp)
                                                                             movl [rbp-8], eax
  c = a+b:
                                  movl -20(%rbp), %edx
                                                                             movl eax,[rbp-8]
  d = func2(c);
                                  movl -24(%rbp), %eax
                                                                             movl edi, eax
  return c+d;
                                  addl %edx, %eax
                                                                             call func2
                                  movl %eax, -8(%rbp)
                                                                             movl [rbp-4], eax
                                  movl -8(%rbp), %eax
                                                                             movl edx, [rbp-8]
                                  movl %eax, %edi
                                                                             movl eax, [rbp-4]
                                  call func2
                                                                             addl eax, edx
                                  movl %eax, -4(%rbp)
                                                                             leave
                                  movI -8(%rbp), %edx
                                  movl -4(%rbp), %eax
                                                                             ret
                                  addl %edx, %eax
                                  leave
                                  ret
```

```
func1:
                                                                        func1:
                                                                             push
                                                                                       rbp
                                                                             mov
                                                                                       rbp, rsp,
                                  pushq
                                            %rbp
                                                                             sub rsp,32
                                                                             movl [rbp-20],edi
                                            %rsp, %rbp
                                                                             movl [rbp-24],esi
                                  movq
int func1(int a, int b)
                                                                             movl edx, [rbp-20]
                                  subq $32, %rsp
                                                                             movl eax, [rbp-24]
                                  movl %edi, -20(%rbp)
                                                                             addl eax, edx
  int c,d;
                                  movl %esi, -24(%rbp)
                                                                             movl [rbp-8], eax
  c = a+b:
                                  movl -20(%rbp), %edx
                                                                             movl eax,[rbp-8]
  d = func2(c);
                                  movl -24(%rbp), %eax
                                                                             movl edi, eax
  return c+d;
                                  addl %edx, %eax
                                                                             call func2
                                  movl %eax, -8(%rbp)
                                                                             movl [rbp-4], eax
                                  movl -8(%rbp), %eax
                                                                             movl edx, [rbp-8]
                                  movl %eax, %edi
                                                                             movl eax, [rbp-4]
                                  call func2
                                                                             addl eax, edx
                                  movl %eax, -4(%rbp)
                                                                             leave
                                  movl -8(%rbp), %edx
                                  movl -4(%rbp), %eax
                                                                             ret
                                  addl %edx, %eax
                                  leave
                                  ret
```

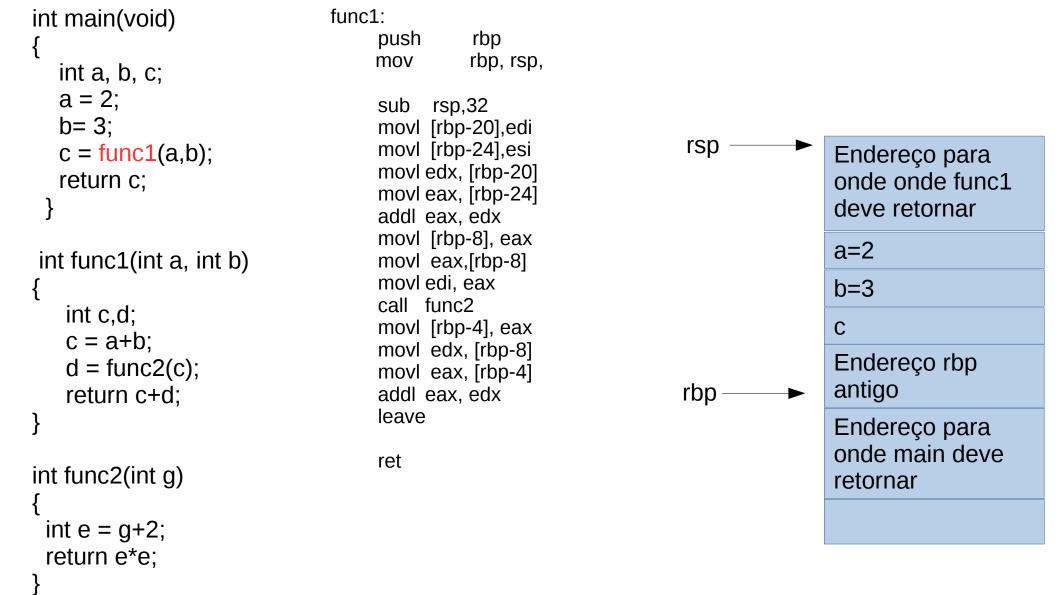
```
func1:
                                                                        func1:
                                                                             push
                                                                                       rbp
                                                                             mov
                                                                                       rbp, rsp,
                                  pushq
                                            %rbp
                                                                             sub rsp,32
                                                                             movl [rbp-20],edi
                                            %rsp, %rbp
                                                                             movl [rbp-24],esi
                                  movq
int func1(int a, int b)
                                                                             movl edx, [rbp-20]
                                  subq $32, %rsp
                                                                             movl eax, [rbp-24]
                                  movl %edi, -20(%rbp)
                                                                             addl eax, edx
  int c,d;
                                  movl %esi, -24(%rbp)
                                                                             movl [rbp-8], eax
  c = a+b:
                                  movl -20(%rbp), %edx
                                                                             movl eax,[rbp-8]
  d = func2(c);
                                  movl -24(%rbp), %eax
                                                                             movl edi, eax
  return c+d;
                                  addl %edx, %eax
                                                                             call func2
                                  movl %eax, -8(%rbp)
                                                                             movl [rbp-4], eax
                                  movl -8(%rbp), %eax
                                                                             movl edx, [rbp-8]
                                  movl %eax, %edi
                                                                             movl eax, [rbp-4]
                                  call func2
                                                                             addl eax, edx
                                  movl %eax, -4(%rbp)
                                                                             leave
                                  movI -8(%rbp), %edx
                                  movl -4(%rbp), %eax
                                                                             ret
                                  addl %edx, %eax
                                  leave
                                  ret
```

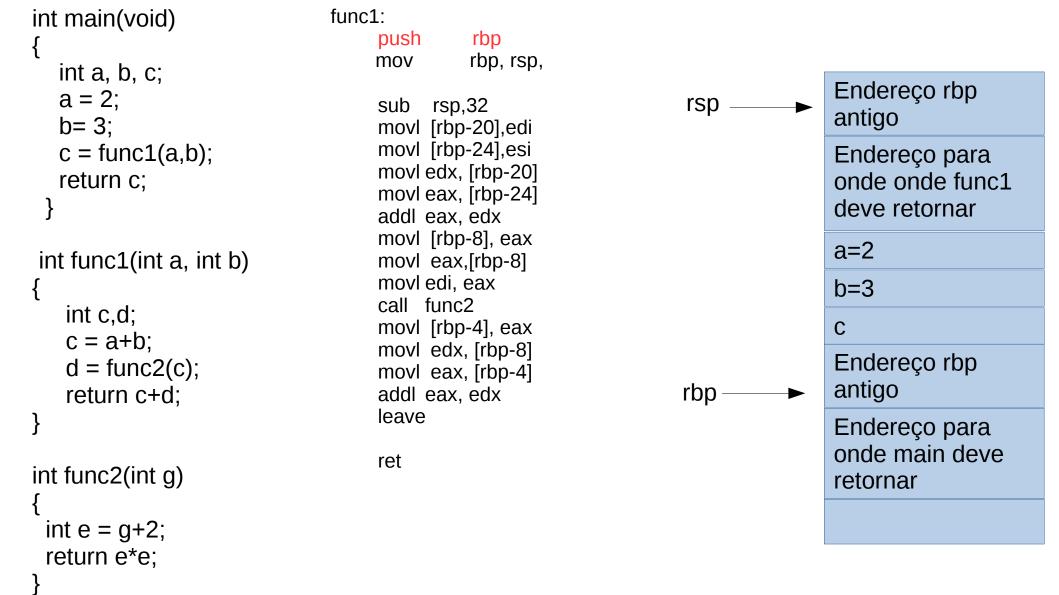
```
func1:
                                                                        func1:
                                                                             push
                                                                                       rbp
                                                                             mov
                                                                                       rbp, rsp,
                                  pushq
                                            %rbp
                                                                             sub rsp,32
                                                                             movl [rbp-20],edi
                                            %rsp, %rbp
                                                                             movl [rbp-24],esi
                                  movq
int func1(int a, int b)
                                                                             movl edx, [rbp-20]
                                  subq $32, %rsp
                                                                             movl eax, [rbp-24]
                                  movl %edi, -20(%rbp)
                                                                             addl eax, edx
  int c,d;
                                  movl %esi, -24(%rbp)
                                                                             movl [rbp-8], eax
  c = a+b:
                                  movl -20(%rbp), %edx
                                                                             movl eax,[rbp-8]
  d = func2(c);
                                  movl -24(%rbp), %eax
                                                                             movl edi, eax
  return c+d;
                                  addl %edx, %eax
                                                                             call func2
                                  movl %eax, -8(%rbp)
                                                                             movl [rbp-4], eax
                                  movl -8(%rbp), %eax
                                                                             movl edx, [rbp-8]
                                  movl %eax, %edi
                                                                             movl eax, [rbp-4]
                                  call func2
                                                                             addl eax, edx
                                  movl %eax, -4(%rbp)
                                                                             leave
                                  movl -8(%rbp), %edx
                                  movl -4(%rbp), %eax
                                                                             ret
                                  addl %edx, %eax
                                  leave
                                  ret
```

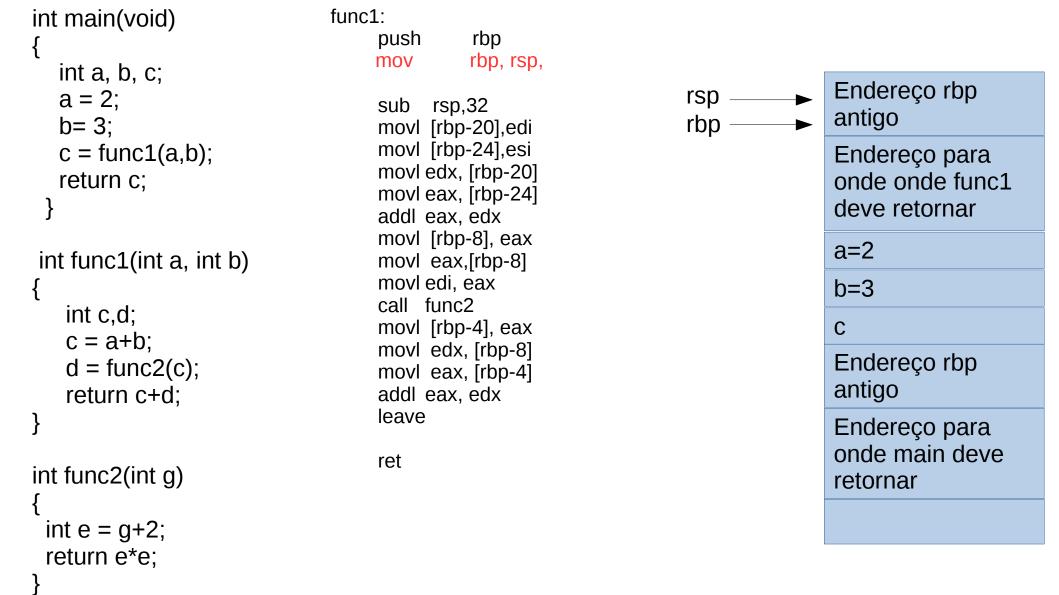
```
func1:
                                                                        func1:
                                                                             push
                                                                                       rbp
                                                                             mov
                                                                                       rbp, rsp,
                                  pushq
                                            %rbp
                                                                             sub rsp,32
                                                                             movl [rbp-20],edi
                                            %rsp, %rbp
                                                                             movl [rbp-24],esi
                                  movq
int func1(int a, int b)
                                                                             movl edx, [rbp-20]
                                  subq $32, %rsp
                                                                             movl eax, [rbp-24]
                                  movl %edi, -20(%rbp)
                                                                             addl eax, edx
  int c,d;
                                  movl %esi, -24(%rbp)
                                                                             movl [rbp-8], eax
  c = a+b:
                                  movl -20(%rbp), %edx
                                                                             movl eax,[rbp-8]
  d = func2(c);
                                  movl -24(%rbp), %eax
                                                                             movl edi, eax
  return c+d;
                                  addl %edx, %eax
                                                                             call func2
                                  movl %eax, -8(%rbp)
                                                                             movl [rbp-4], eax
                                  movl -8(%rbp), %eax
                                                                             movl edx, [rbp-8]
                                  movl %eax, %edi
                                                                             movl eax, [rbp-4]
                                  call func2
                                                                             addl eax, edx
                                  movl %eax, -4(%rbp)
                                                                             leave
                                  movl -8(%rbp), %edx
                                  movl -4(%rbp), %eax
                                                                             ret
                                  addl %edx, %eax
                                  leave
                                  ret
```

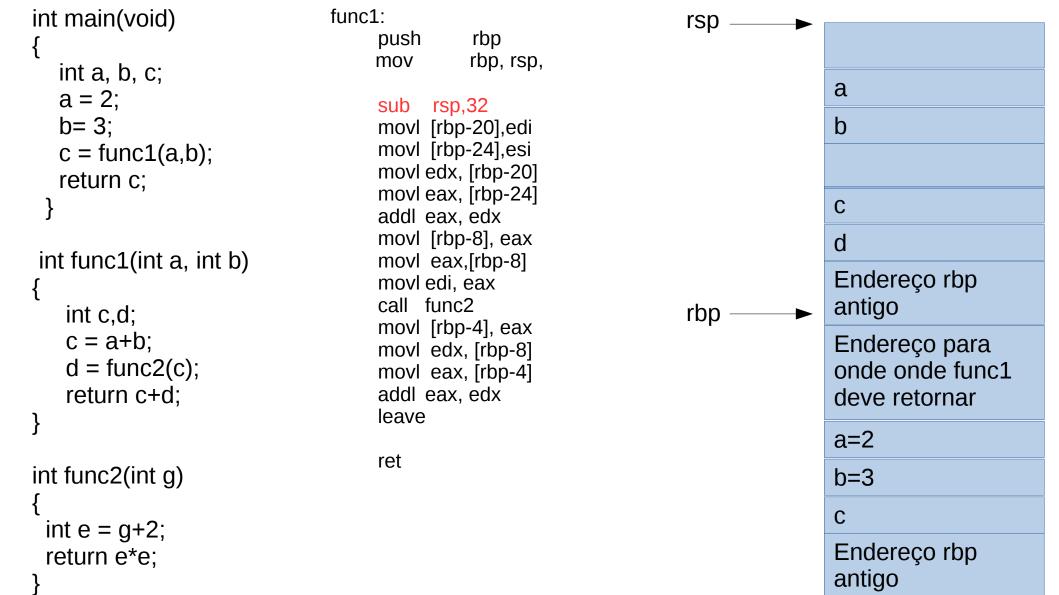
```
func1:
                                                                        func1:
                                                                             push
                                                                                       rbp
                                                                             mov
                                                                                       rbp, rsp,
                                  pushq
                                            %rbp
                                                                             sub rsp,32
                                                                             movl [rbp-20],edi
                                            %rsp, %rbp
                                                                             movl [rbp-24],esi
                                  movq
int func1(int a, int b)
                                                                             movl edx, [rbp-20]
                                  subq $32, %rsp
                                                                             movl eax, [rbp-24]
                                  movl %edi, -20(%rbp)
                                                                             addl eax, edx
  int c,d;
                                  movl %esi, -24(%rbp)
                                                                             movl [rbp-8], eax
  c = a+b:
                                  movl -20(%rbp), %edx
                                                                             movl eax,[rbp-8]
  d = func2(c);
                                  movl -24(%rbp), %eax
                                                                             movl edi, eax
  return c+d;
                                  addl %edx, %eax
                                                                             call func2
                                  movl %eax, -8(%rbp)
                                                                             movl [rbp-4], eax
                                  movl -8(%rbp), %eax
                                                                             movl edx, [rbp-8]
                                  movl %eax, %edi
                                                                             movl eax, [rbp-4]
                                  call func2
                                                                             addl eax, edx
                                  movl %eax, -4(%rbp)
                                                                             leave
                                  movl -8(%rbp), %edx
                                  movl -4(%rbp), %eax
                                                                             ret
                                  addl %edx, %eax
                                  leave
                                  ret
```

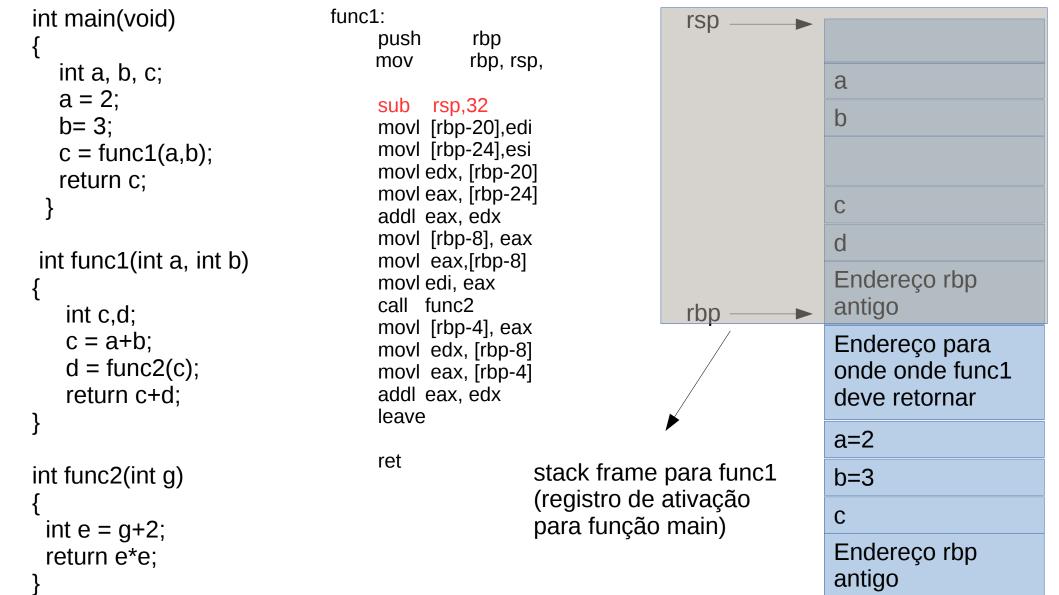
```
func1:
                                                                        func1:
                                                                             push
                                                                                       rbp
                                                                             mov
                                                                                       rbp, rsp,
                                  pushq
                                            %rbp
                                                                             sub rsp,32
                                                                             movl [rbp-20],edi
                                            %rsp, %rbp
                                                                             movl [rbp-24],esi
                                  movq
int func1(int a, int b)
                                                                             movl edx, [rbp-20]
                                  subq $32, %rsp
                                                                             movl eax, [rbp-24]
                                  movl %edi, -20(%rbp)
                                                                             addl eax, edx
  int c,d;
                                  movl %esi, -24(%rbp)
                                                                             movl [rbp-8], eax
  c = a+b:
                                  movl -20(%rbp), %edx
                                                                             movl eax,[rbp-8]
  d = func2(c);
                                  movl -24(%rbp), %eax
                                                                             movl edi, eax
  return c+d;
                                  addl %edx, %eax
                                                                             call func2
                                  movl %eax, -8(%rbp)
                                                                             movl [rbp-4], eax
                                  movl -8(%rbp), %eax
                                                                             movl edx, [rbp-8]
                                  movl %eax, %edi
                                                                             movl eax, [rbp-4]
                                  call func2
                                                                             addl eax, edx
                                  movl %eax, -4(%rbp)
                                                                             leave
                                  movI -8(%rbp), %edx
                                  movl -4(%rbp), %eax
                                                                             ret
                                  addl %edx, %eax
                                  leave
                                  ret
```

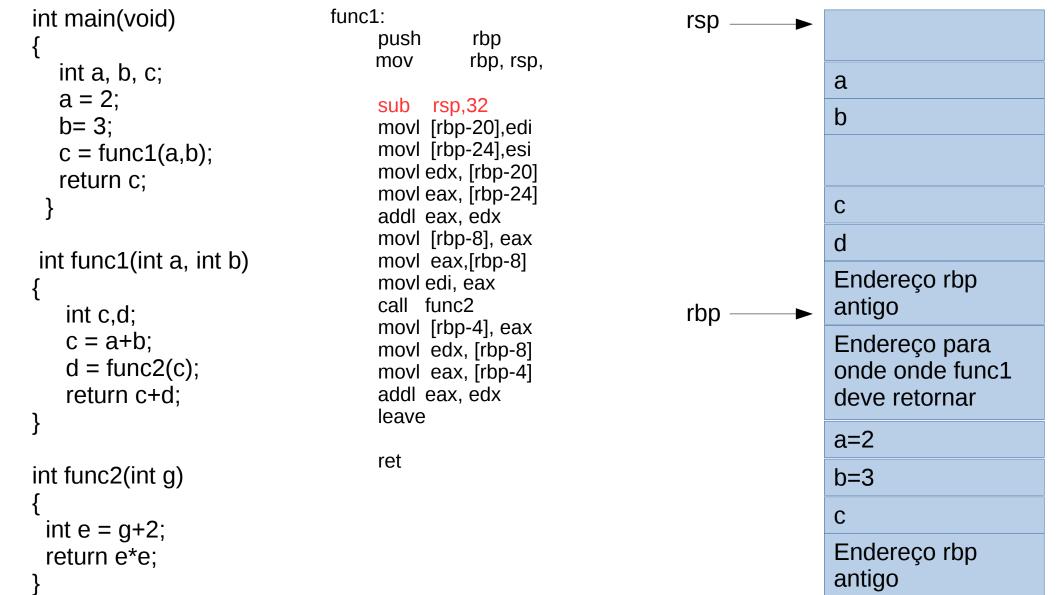


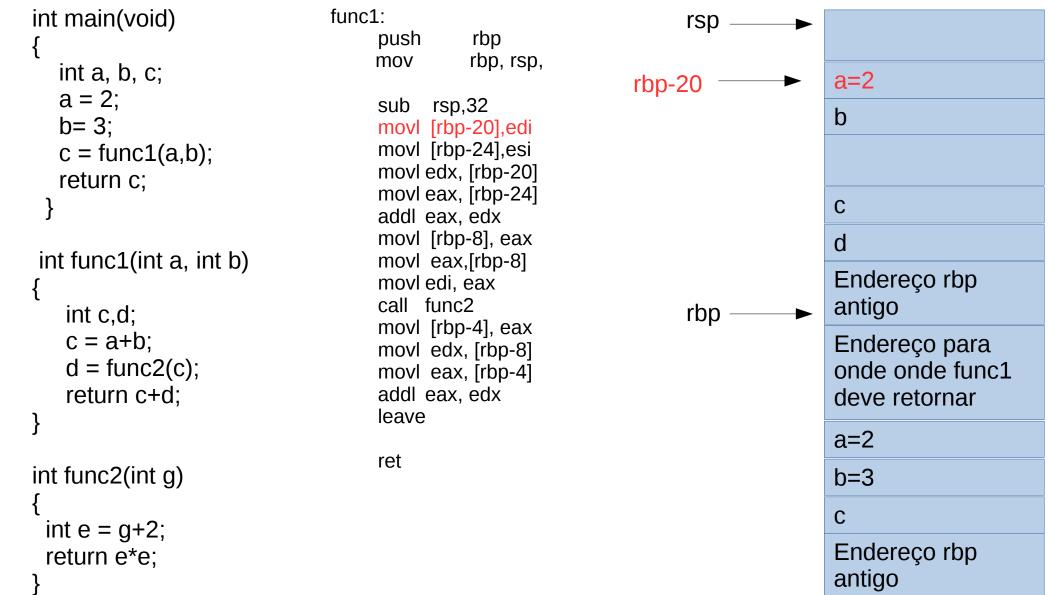


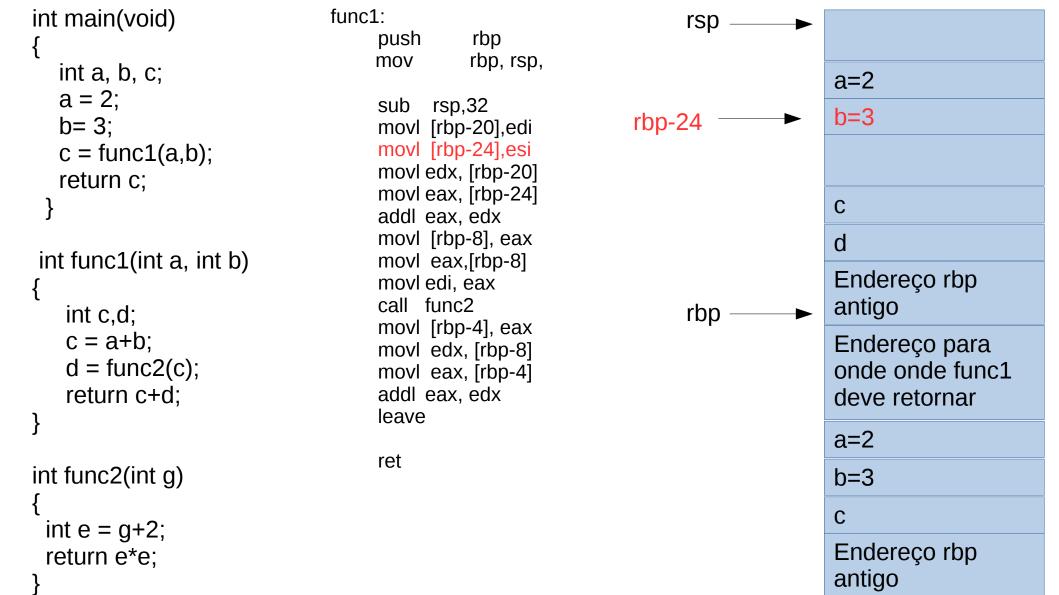


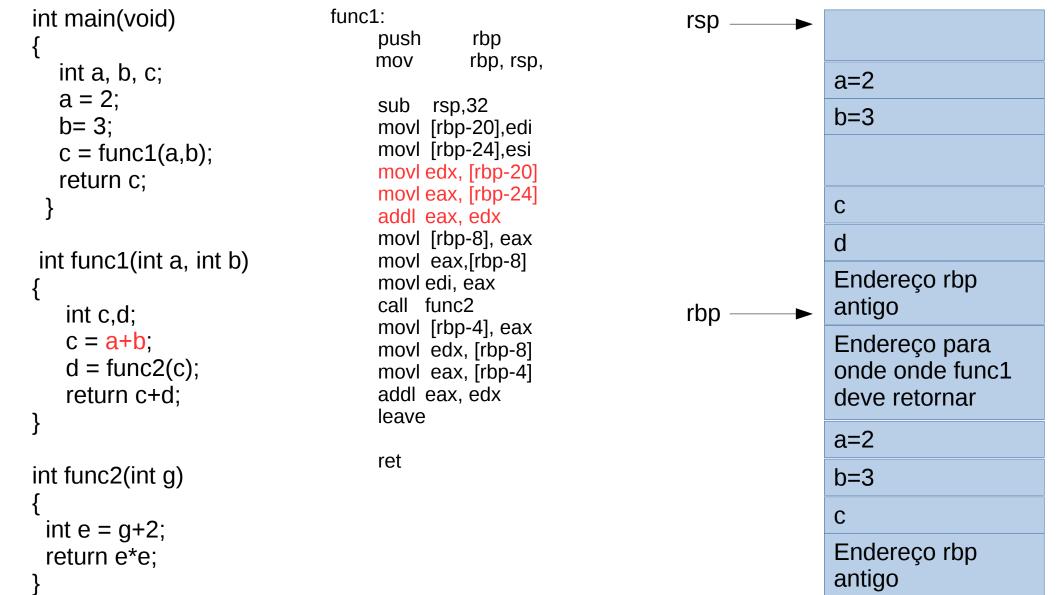


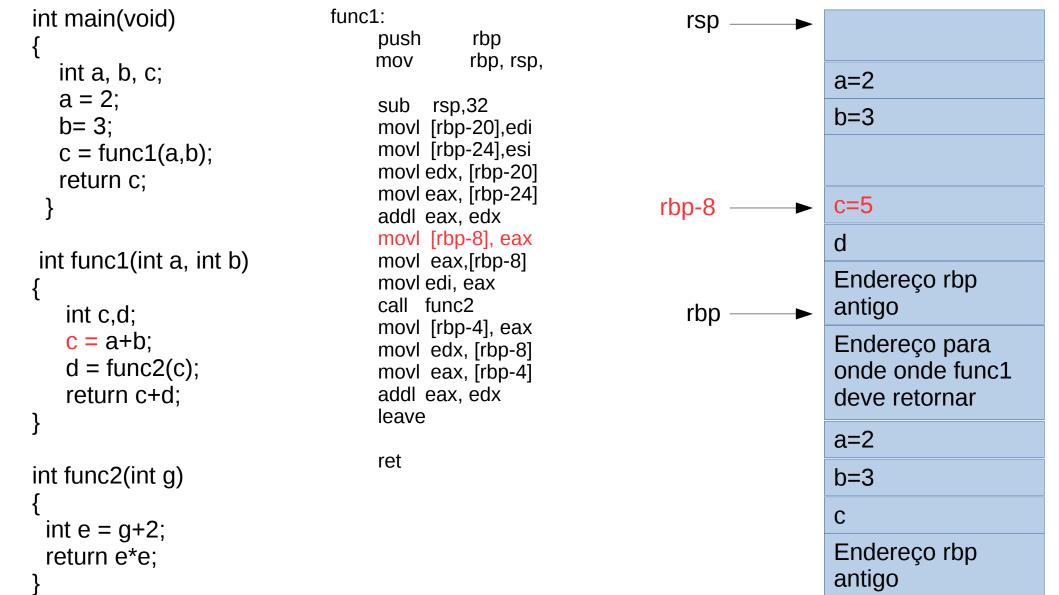


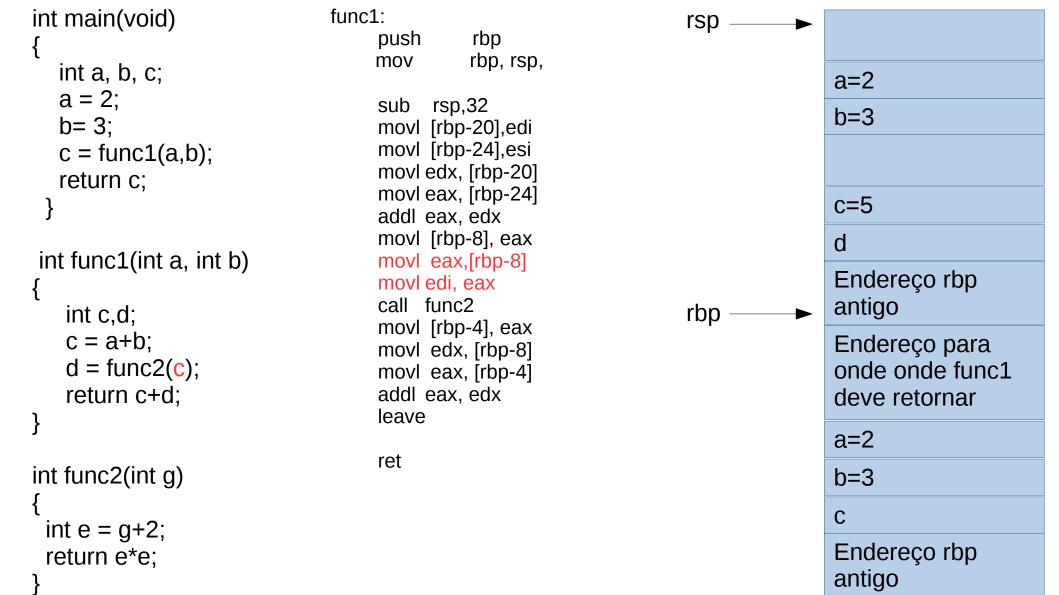


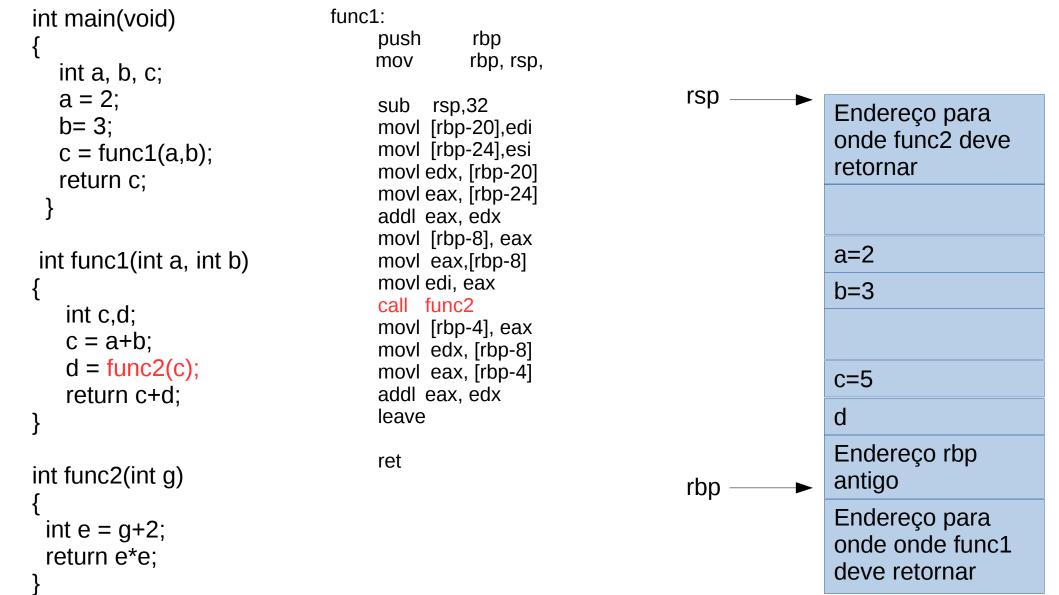


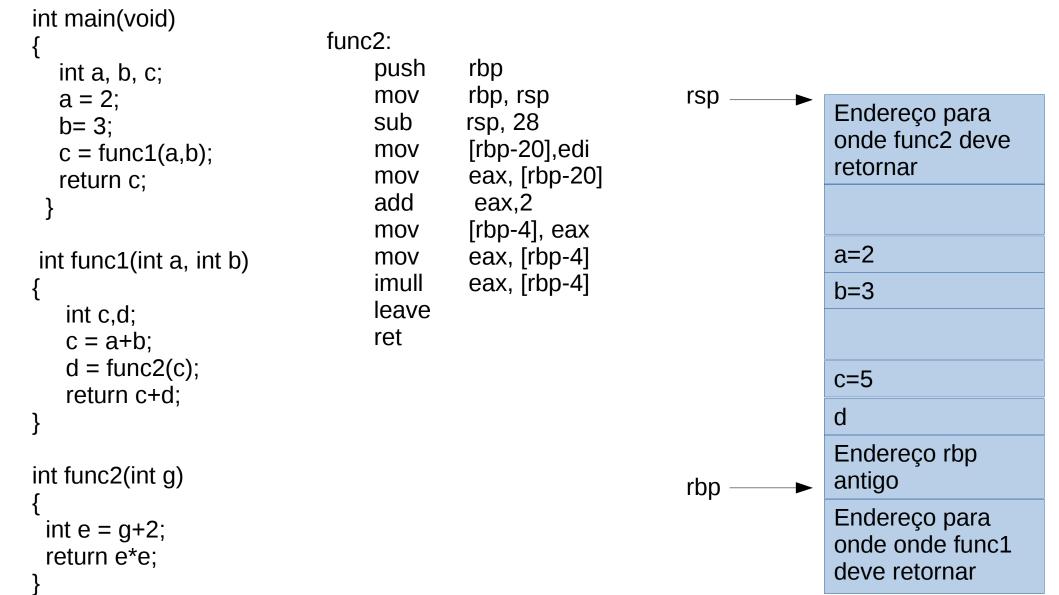


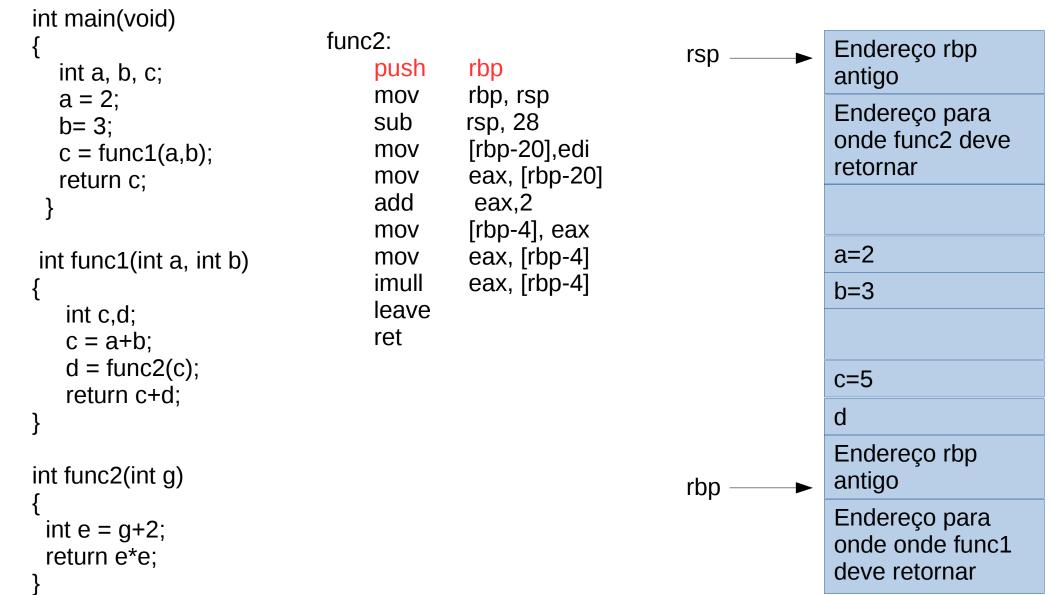


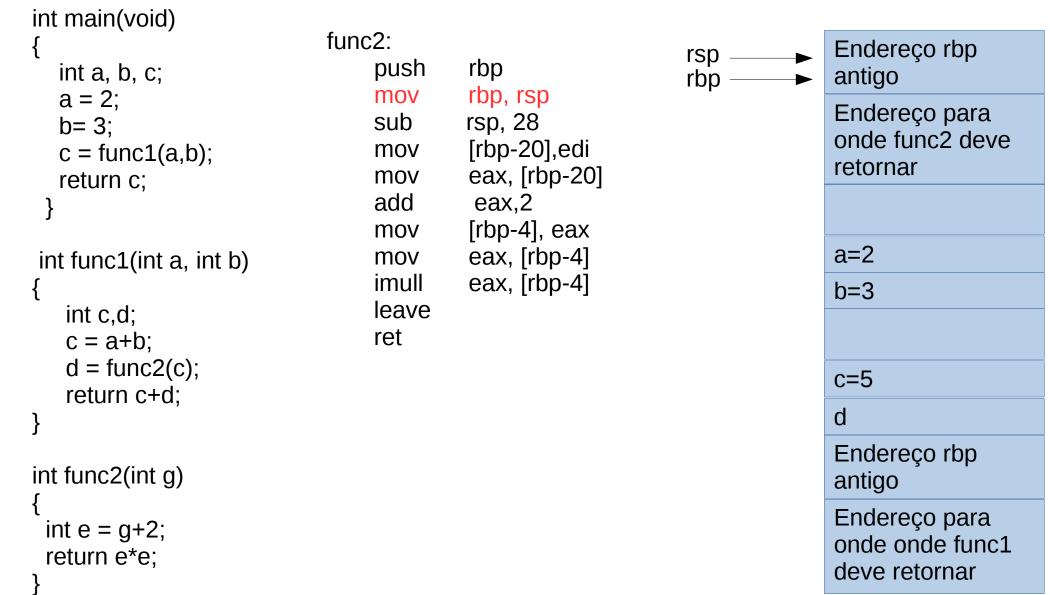


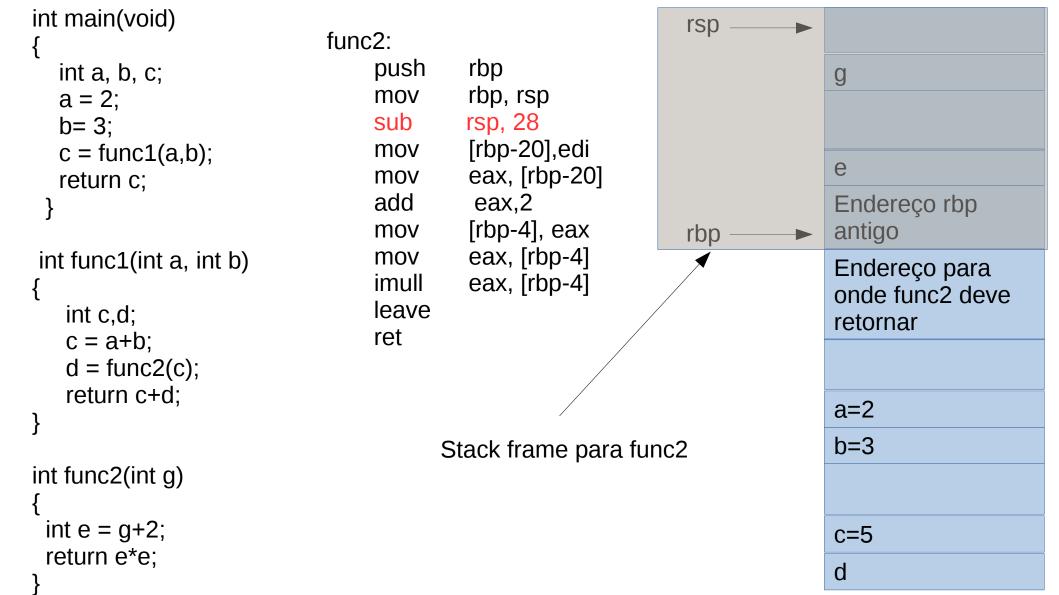


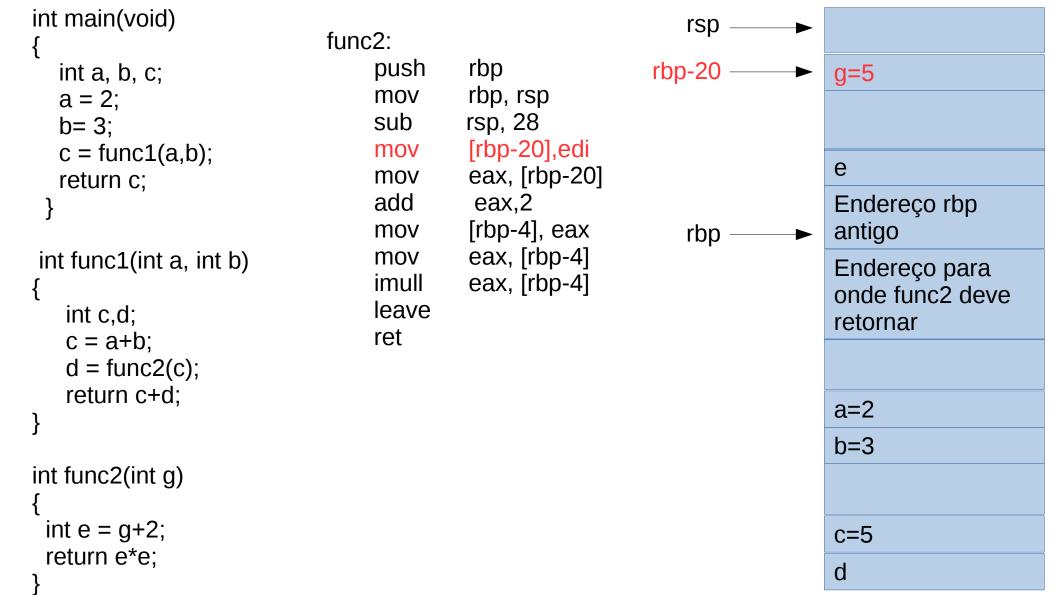


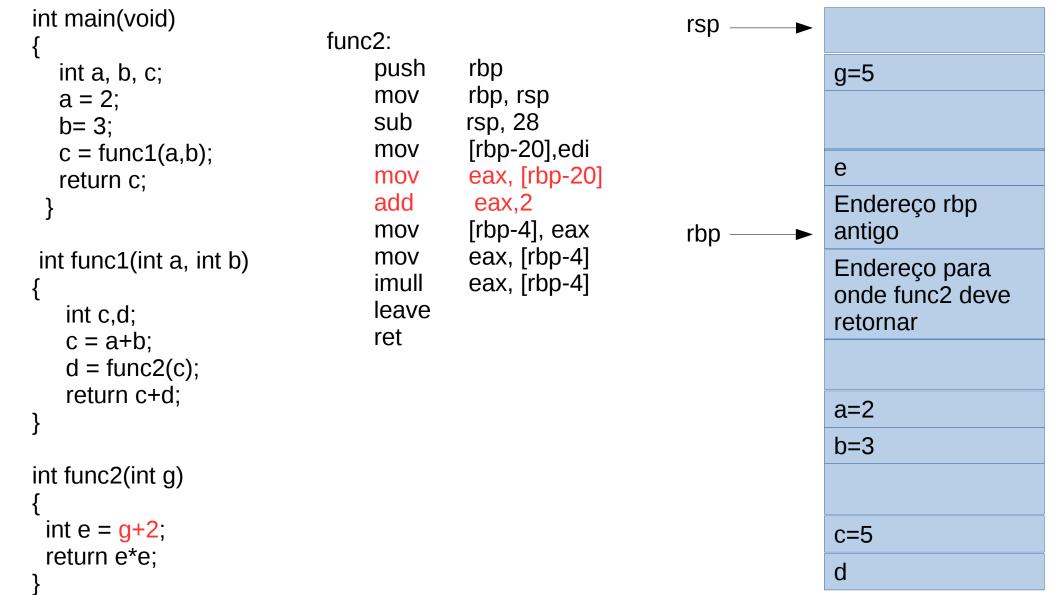


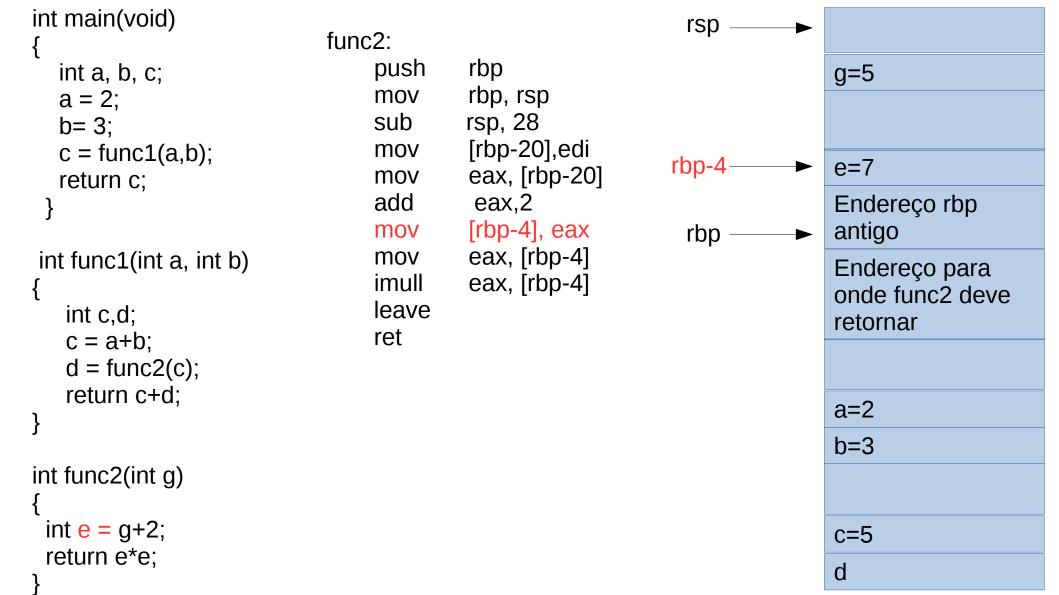


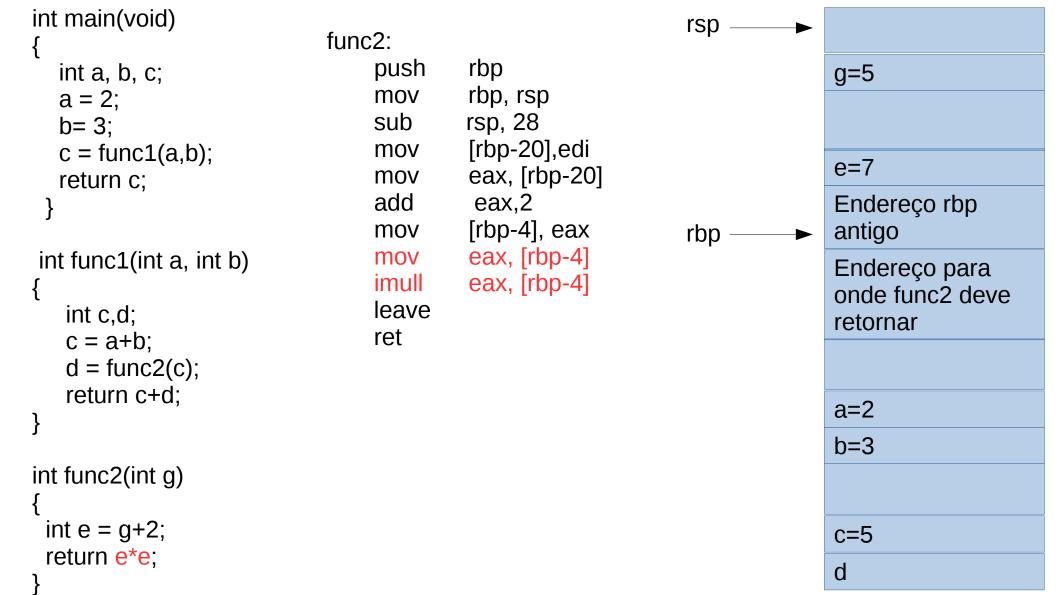


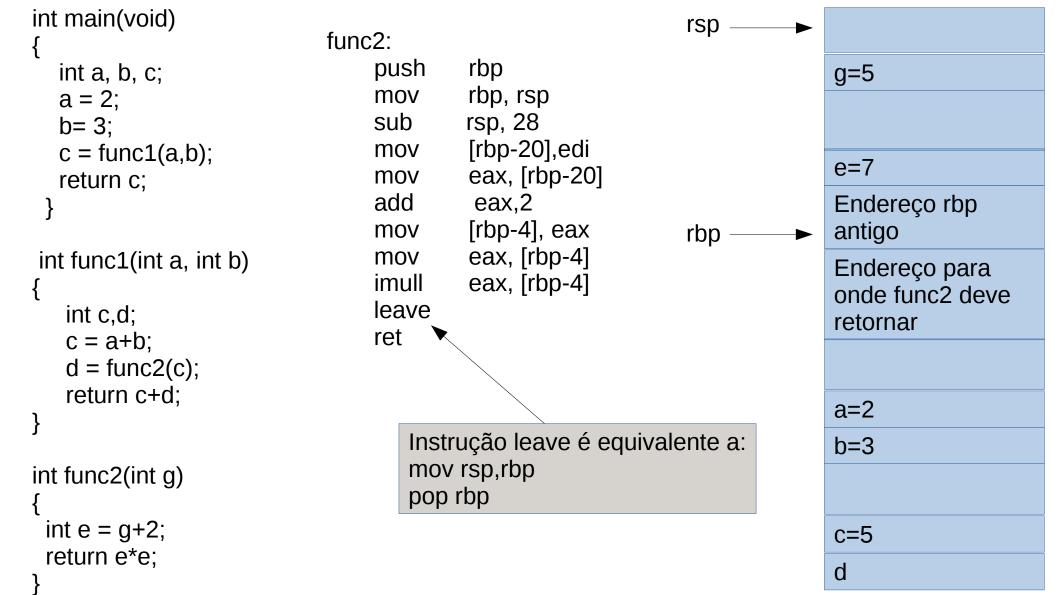


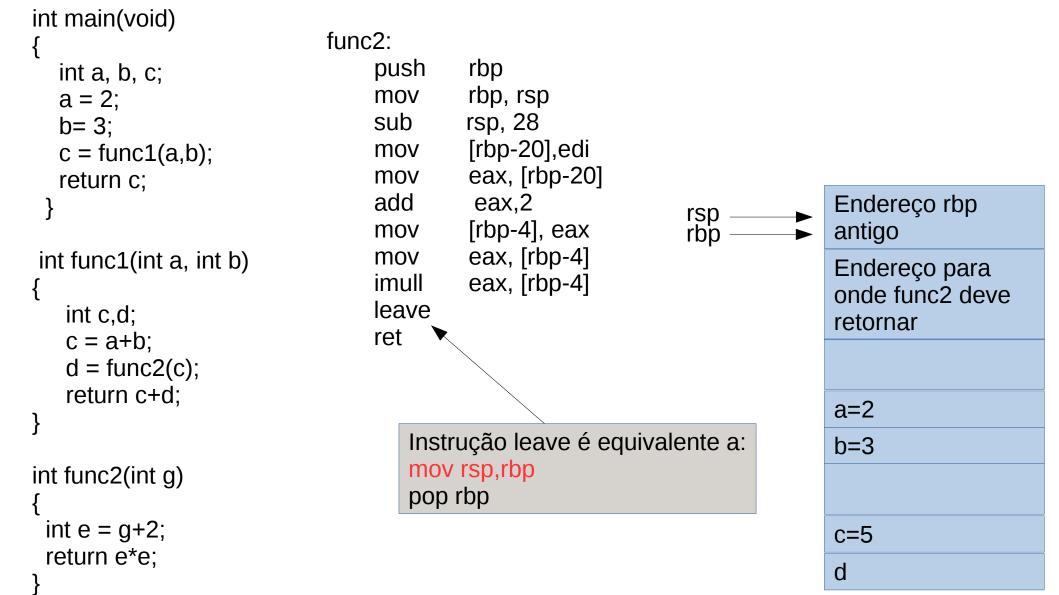


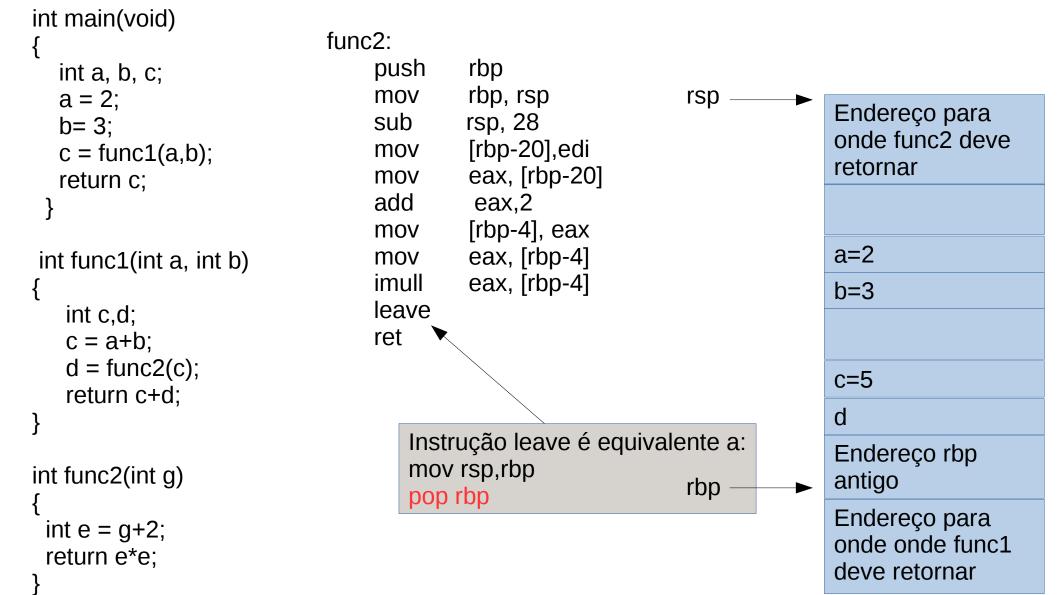


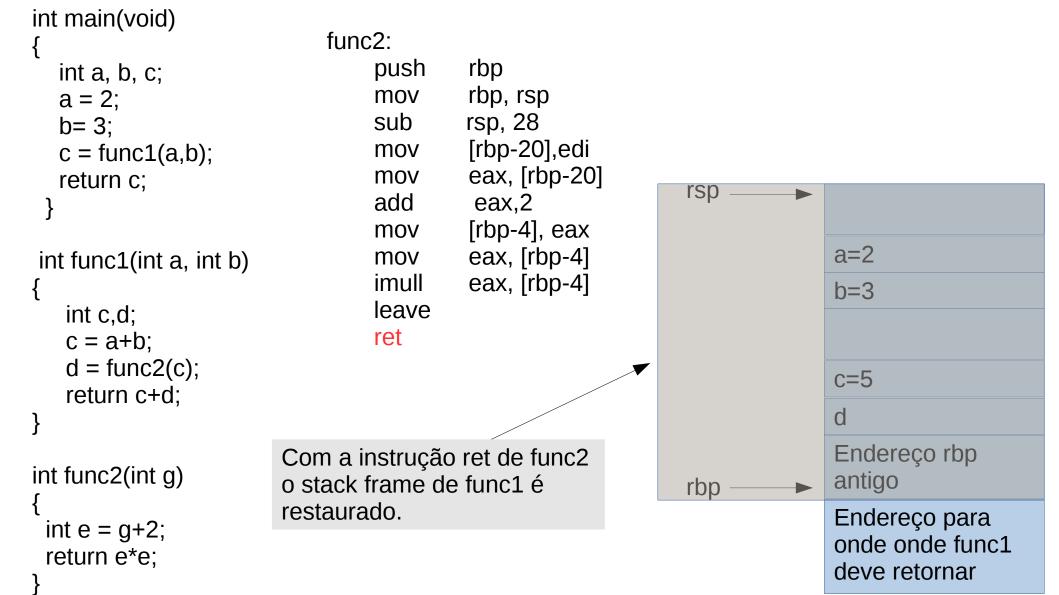


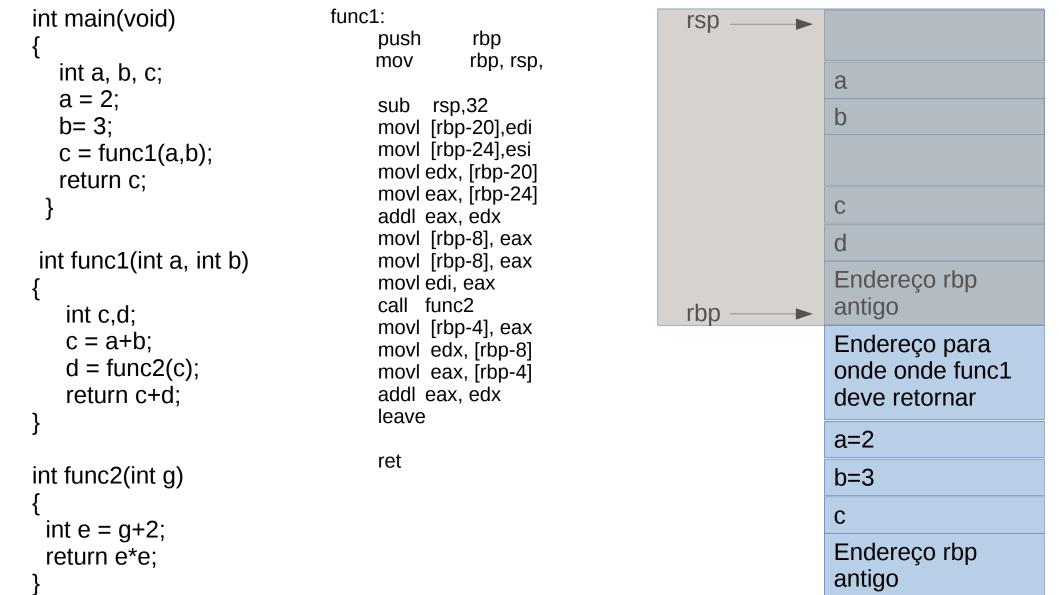


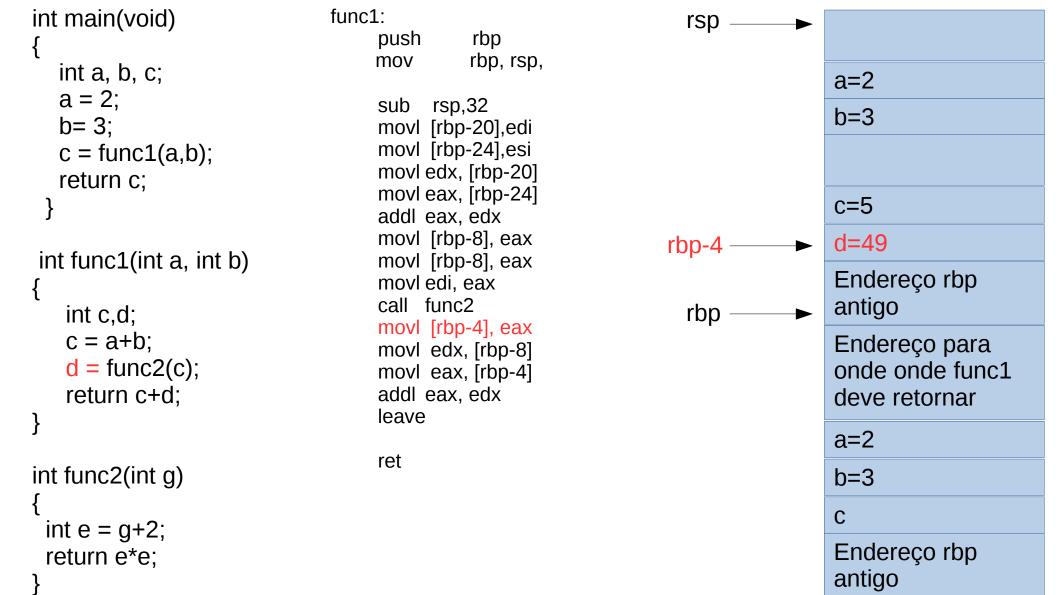


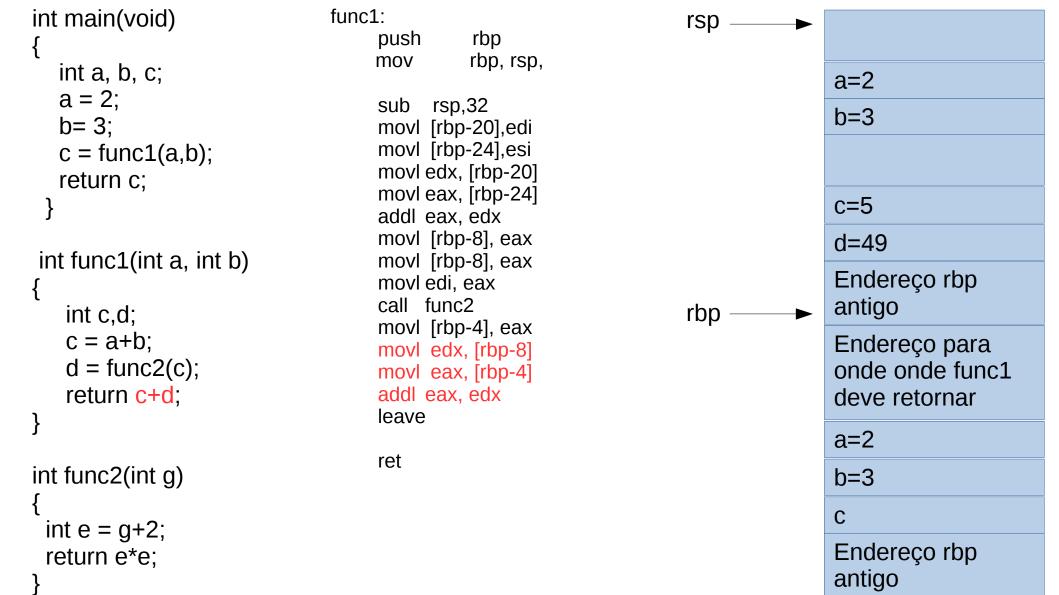


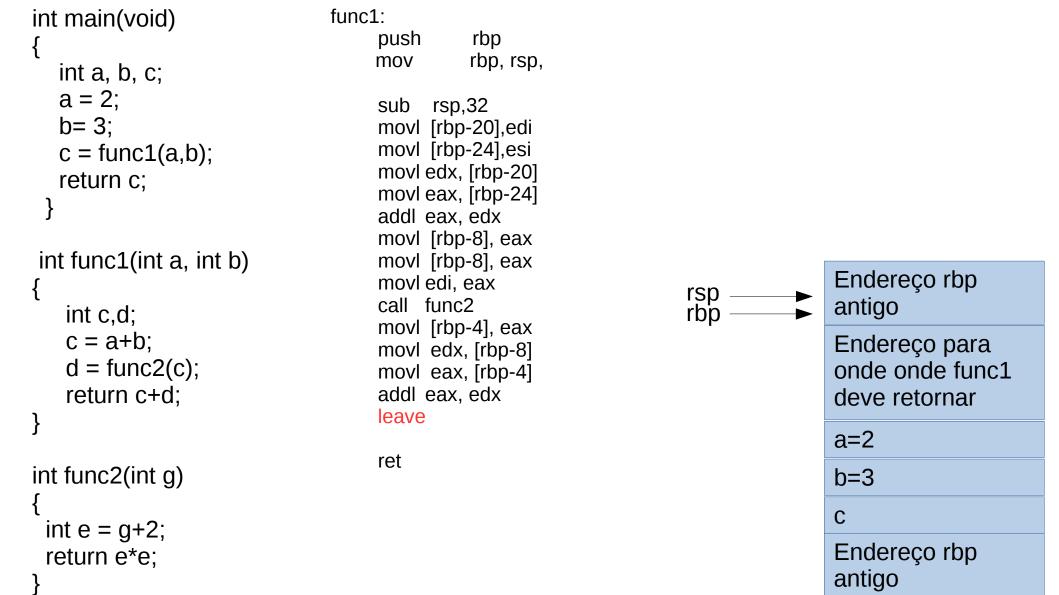


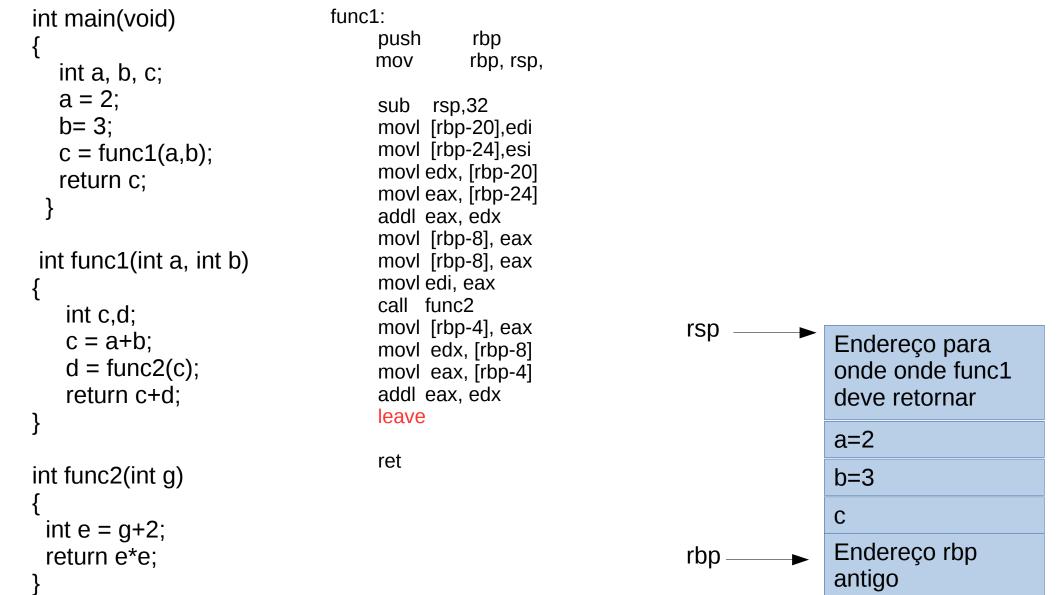


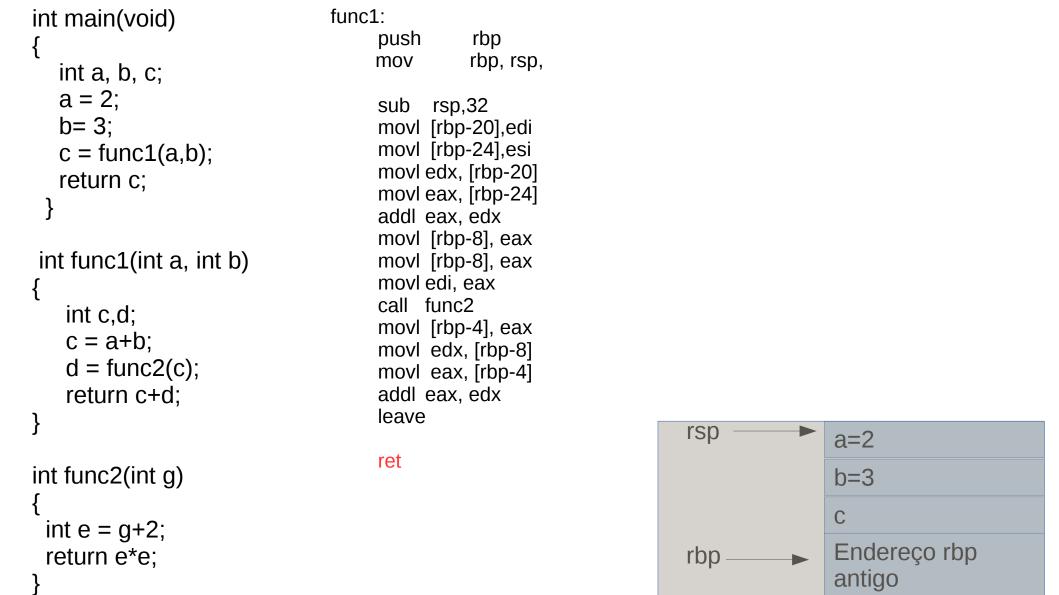


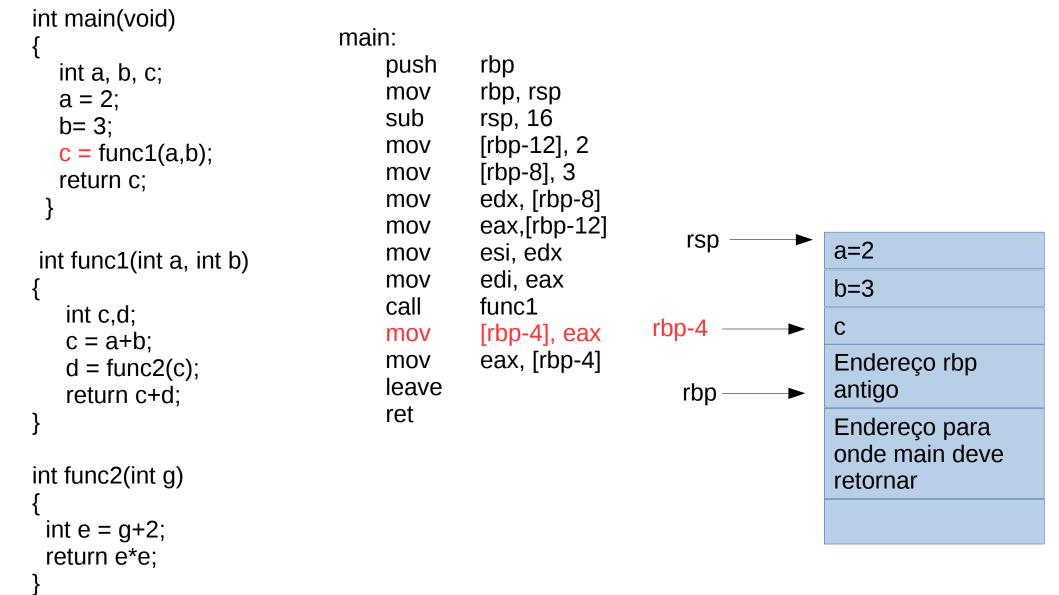


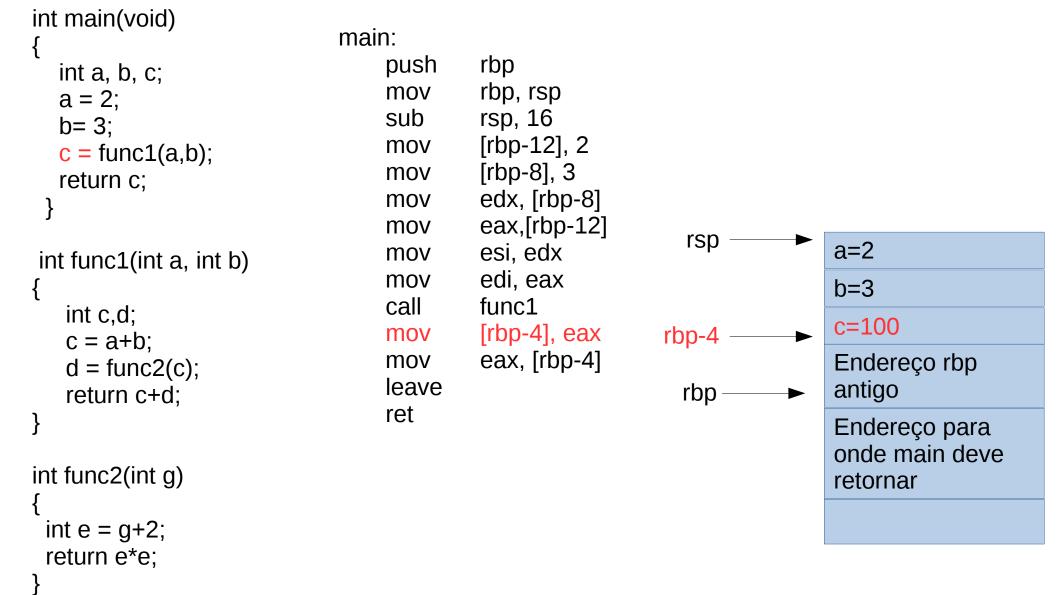












## Convenção de chamada em C - 64 bits

- Primeiro parâmetro em rdi Segundo parâmetro em rsi Terceiro parâmetro em rdx Quarto parâmetro em rcx Quinto parâmetro em r8 Sexto parâmetro em r9
- Valor de retorno em rax.

## E se tiver mais de 6 parâmetros

 Os demais parâmetros são empilhados, da mesma maneira que em 32 bits.