Arquitectura de Computadoras y Sistemas operativos



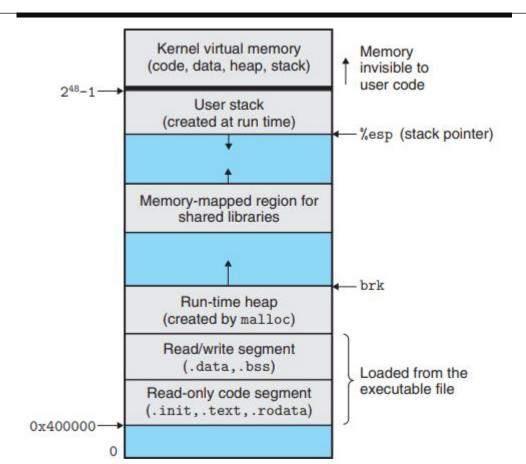
CLASE 4

Agenda



- Stack en funciones
- Registros x86-64
- Instrucciones x86-64 Intel
- Convenciones C x86
- Ejemplo CompletoStack Standards
- GDB Stack, PC, little-big endian
- Mapa de Memoria
- Tipo de Instrucciones de Assembly
- Syntaxis de registros, immediates, etc







```
int func_b(int a, int c) {
  int total = a + c;
  return total;
int func_a(int a) {
  int c = 7;
  if ((a + c) % 2) {
      func_b(a, c);
int main() {
  int d = 8;
  func_a(4);
```

0x7FFFFFFFda00

Main



```
int func_b(int a, int c) {
  int total = a + c;
  return total;
int func_a(int a) {
  int c = 7;
  if ((a + c) % 2) {
      func_b(a, c);
int main() {
  int d = 8;
  func_a(4); ←
```

0x7FFFFFFFda00

Main	
Int d 8	



```
int func_b(int a, int c) {
  int total = a + c;
  return total;
int func_a(int a) { -
  int c = 7;
  if ((a + c) % 2) {
      func_b(a, c);
int main() {
  int d = 8;
  func_a(4);
```

0x7FFFFFFFda00

Main int d 8		
Func_a		



```
int func_b(int a, int c) {
  int total = a + c;
  return total;
int func_a(int a) {
  int c = 7;
  if ((a + c) % 2) {
      func_b(a, c);
int main() {
  int d = 8;
  func_a(4);
```

0x7FFFFFFda00

Main int d 8	
Func_a int c 7	int a 4



```
int func_b(int a, int c) {
  int total = a + c;
  return total;
int func_a(int a) {
  int c = 7;
  if ((a + c) % 2) {
      func_b(a, c);
int main() {
  int d = 8;
  func_a(4);
```

0x7FFFFFFFda00

Main int d 8	
Func_a int c 7 int a 4	
Func_b int total int c 7	int a 4



```
int func_b(int a, int c) {
  int total = a + c;
  return total;
int func_a(int a) {
  int c = 7;
  if ((a + c) % 2) {
      func_b(a, c);
int main() {
  int d = 8;
  func_a(4);
```

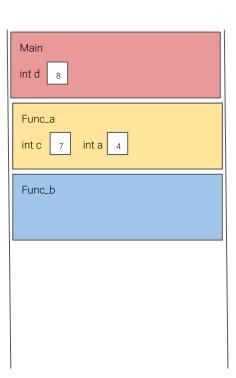
0x7FFFFFFFda00

Main int d 8	
Func_a int c 7 int a 4	
Func_b int total 11 int c 7	int a 4

Stack - funciones subsequentes



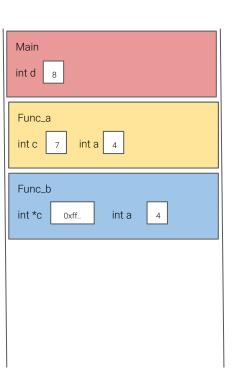
```
0x7FFFFFFFda00
void func b(int a, int *c) {
   *c = a + *c;
int func_a(int a) {
                            Se puede pasar la dir de una
   int c = 7;
   if ((a + c) % 2)
                            variable en el stack anterior
      func_b(a, &c);
int main() {
  int d = 8;
  func_a(4);
```



Stack - funciones subsequentes



```
0x7FFFFFFFda00
void func b(int a, int *c) {
   *c = a + *c;
int func_a(int a) {
                            Se puede pasar la dir de una
   int c = 7;
   if ((a + c) % 2)
                            variable en el stack anterior
      func_b(a, &c);
int main() {
  int d = 8;
  func_a(4);
```



Stack - funciones subsequentes



```
0x7FFFFFFFda00
void func b(int a, int *c) {
                                                                         Main
   *c = a + *c;
                                                                         int d 8
                                                                         Func_a
int func_a(int a) {
                                                                         int c 11 int a 4
                            Se puede pasar la dir de una
  int c = 7;
  if ((a + c) % 2) {
                            variable en el stack anterior
                                                                         Func_b
      func_b(a, &c);
                                                                         int *c
                                                                                    int a
int main() {
  int d = 8;
  func_a(4);
```

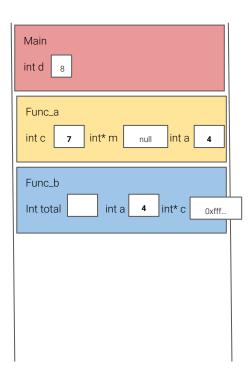
Stack - stack incorrecto



0x7FFFFFFFda00

```
int* func b(int a, int *c) {
   int total = *c;
   total = a + total;
   return &total;
int func a(int a) {
   int c = 7;
  int* m = NULL;
  if ((a + c) % 2) {
       m = func b(a, &c);
   printf("m: %d\n", *m);
int main() {
   int d = 8;
   func a(4);
```

No se puede pasar variables del stack a stacks anteriores





0x7FFFFFFda00

```
int* func b(int a, int *c) {
   int total = *c;
   total = a + total;
   return &total;
```

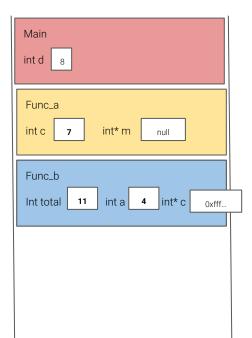
```
int func a(int a) {
  int c = 7;
```

```
int* m = NULL;
if ((a + c) % 2) {
    m = func b(a, &c);
```

```
printf("m: %d\n", *m);
```

```
int main() {
   int d = 8;
   func a(4);
```

No se puede pasar variables del stack a stacks anteriores

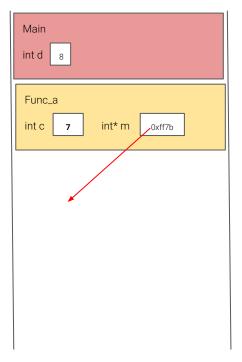


Stack - stack incorrecto



0x7FFFFFFda00

```
int* func b(int a, int *c) {
  int total = *c;
  total = a + total;
  return &total;
int func a(int a) {
                                No se puede pasar variables
  int c = 7;
                                del stack a stacks anteriores
  int* m = NULL;
  if ((a + c) % 2) {
      m = func b(a, &c);
  printf("m: %d\n", *m);
int main() {
  int d = 8;
  func a(4);
```



Stack - stack incorrecto



0x7FFFFFFFda00

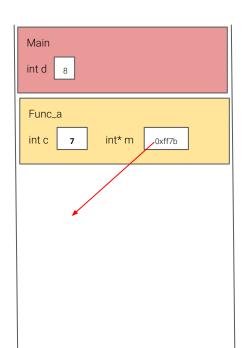
```
int total = *c;
   total = a + total;
   return &total;
int func a(int a) {
   int c = 7;
   int* m = NULL;
  if ((a + c) % 2) {
       m \neq func b(a, &c);
   printf "m: %d n", *m);
int main() {
   int d = 8;
```

func a(4);

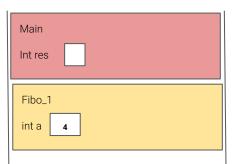
int* func b(int a, int *c) {

No se puede pasar variables del stack a stacks anteriores

Acceso a memoria no valida









```
int fibo(int a) {
   if (a <= 2) {
      return 1;
   }
      Cada llamando a función
   return fibo(a-1) + fibo(a-2); tiene su stack
}
int main() {
   int res = fibo(4);
}</pre>
```

Main Int res	
Fibo_1 int a 4	
Fibo_2 int a 3	
Fibo_3 int a 2	



0x7FFFFFFFda00

```
int fibo(int a) {
   if (a <= 2) {
      return 1;
   }
   return fibo(a-1) + fibo(a-2);
}
int main() {
   int res = fibo(4);
}</pre>
```

Main Int res
Fibo_1 int a 4
Fibo_2 int a 3
Fibo_3 int a 2
Fibo_4 int a 2



```
int fibo(int a) {
   if (a <= 2) {
      return 1;
   }
      Cada llamando a función
   return fibo(a-1) + fibo(a-2);
   int main() {
   int res = fibo(4);
}</pre>
```

Main Int res
Fibo_1 int a 4
Fibo_2 int a 3
Fibo_3 int a 2
Fibo_4 int a 1
Fibo_5



0x7FFFFFFda00

```
int fibo(int a) {
   if (a <= 2) {
      return 1;
   }
   return fibo(a-1) + fibo(a-2);
}
int main() {
   int res = fibo(4);
}</pre>
```

Main Int res
Fibo_1 int a 4
Fibo_2 int a 3
Fibo_3 int a 2
Fibo_4 int a 1



0x7FFFFFFFda00

```
int fibo(int a) {
   if (a <= 2) {
      return 1;
   }
   return fibo(a-1) + fibo(a-2);
}
int main() {
   int res = fibo(4);
}</pre>
```

Main Int res
Fibo_1 int a 4
Fibo_2 int a 3
Fibo_3 int a 2



0x7FFFFFFFda00

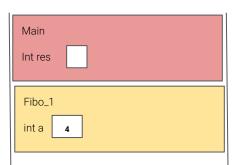
```
int fibo(int a) {
   if (a <= 2) {
      return 1;
   }
   return fibo(a-1) + fibo(a-2);
}
int main() {
   int res = fibo(4);
}</pre>
```

Main Int res	
Fibo_1 int a	4
Fibo_2 int a	3



0x7FFFFFFFda00

```
int fibo(int a) {
   if (a <= 2) {
      return 1;
   }
   return fibo(a-1) + fibo(a-2);
}
int main() {
   int res = fibo(4);
}</pre>
```





0x7FFFFFFFda00





63	31	15	7	
%rax	%eax	%ax	%al	return value
%rbx	%ebx	%bx	%bl	callee saved
%rcx	%ecx	%CX	%cl	4th argument
%rdx	%edx	%dx	%dl	3rd argument
%rsi	%esi	%si	%sil	2nd argument
%rdi	%edi	%di	%dil	1st argument



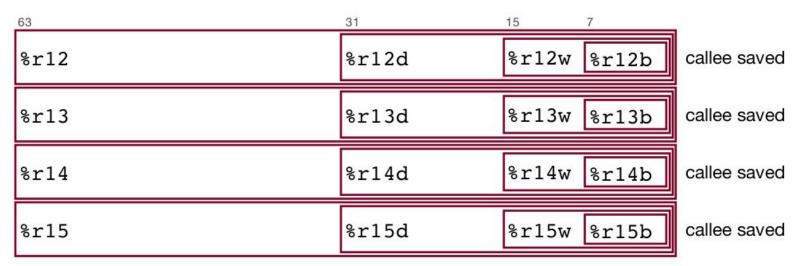
63	31	15	7	•
%rbp	%ebp	%bp	%bpl	callee saved
%rsp	%esp	%sp	%spl	stack pointer
%r8	%r8d	%r8w	%r8b	5th argument
%r9	%r9d	%r9w	%r9b	6th argument
%r10	%r10d	%r10w	%r10b	caller saved
%r11	%r11d	%r11w	%r11b	caller saved





Caller Saved: Son registros volatiles, si una funcion quiere preservar su valor, debe guardarlo en el Stack. El que llama debe guardar antes de llamar





Caller Saved: Son registros volatiles, si una función quiere preservar su valor, debe guardarlo en el Stack. El que llama debe guardar antes de llamar

Callee Saved: Son registros que deben preservar el valor después de un llamado (call), por lo tanto si la función llamada quiere usarlos, debe guardar y restaurar su valor después de usarlo



63	31	15	7	-
%r12	%r12d	%r12v	v %r12b	callee saved
%r13	%r13d	%r13v	v %r13b	callee saved
%r14	%r14d	%r14v	v %r14b	callee saved
%r15	%r15d	%r15v	v %r15b	callee saved

Mas adelante lo vamos a ver en detalle

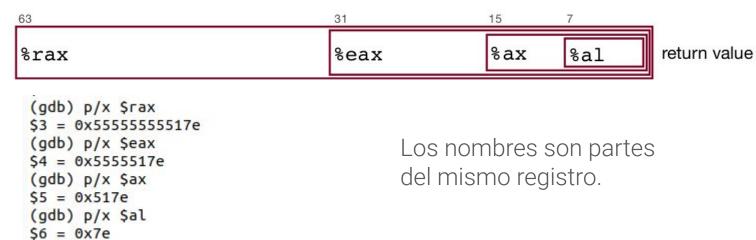


63	31	15	7	-
%r12	%r12d	%r12v	v %r12b	callee saved
%r13	%r13d	%r13v	v %r13b	callee saved
%r14	%r14d	%r14v	v %r14b	callee saved
%r15	%r15d	%r15v	v %r15b	callee saved

Mas adelante lo vamos a ver en detalle

(gdb)







```
000000000000117e <main>:
                f3 Of 1e fa
                                        endbr64
    117e:
                                        push
    1182:
                55
                                               гЬр
    1183:
                48 89 e5
                                        mov
                                               rbp,rsp
                bf 04 00 00 00
                                               edi,0x4
   1186:
                                        mov
               e8 b7 ff ff ff
                                               1147 <func_a>
    118b:
                                        call
    1190:
                b8 00 00 00 00
                                        MOV
                                               eax,0x0
                                               грр
    1195:
                5d
                                        pop
    1196:
                c3
                                        ret
```

Callee Saved: Guardar rbp en el Stack



```
000000000000117e <main>:
                f3 Of 1e fa
                                         endbr64
    117e:
    1182:
                                         push
                55
                                                 грр
    1183:
                48 89 e5
                                                 rbp,rsp
                                         MOV
                                                 edi,0x4
    1186:
                bf 04 00 00 00
                                         mov
    118b:
                e8 b7 ff ff ff
                                         call
                                                 1147 <func a>
    1190:
                b8 00 00 00 00
                                         MOV
                                                 eax,0x0
    1195:
                5d
                                                 грр
                                         pop
    1196:
                c3
                                         ret
```

Estamos usando syntax de intel, destino está a la izquierda. Syntax de AT&T es al reves y distinto



```
000000000000117e <main>:
                                        endbr64
    117e:
                f3 Of 1e fa
    1182:
                55
                                         push
                                                гЬр
    1183:
                48 89 e5
                                        mov
                                                rbp,rsp
                bf 04 00 00 00
                                                edi,0x4
    1186:
                                        MOV
    118b:
                e8 b7 ff ff ff
                                         call
                                                1147 <func_a>
    1190:
                b8 00 00 00 00
                                        MOV
                                                eax,0x0
                                                гЬр
    1195:
                5d
                                         pop
    1196:
                C3
                                        ret
```

Tamaño de instrucciones varia



```
000000000000117e <main>:
                f3 Of 1e fa
                                         endbr64
    117e:
    1182:
                55
                                         push
                                                 грр
    1183:
                48 89 e5
                                                 rbp,rsp
                                         MOV
                                                 edi,0x4
    1186:
                bf 04 00 00 00
                                         MOV
    118b:
                e8 b7 ff ff ff
                                         call
                                                 1147 <func_a>
    1190:
                b8 00 00 00 00
                                         MOV
                                                 eax,0x0
    1195:
                5d
                                                 грр
                                         pop
    1196:
                c3
                                         ret
```

Mover a lo que apunta el rsp al registro rbp.

Restarle el tamaño de lo movido al rsp



```
0000000000000117e <main>:
                    f3 Of 1e fa
                                              endbr64
      117e:
      1182:
                    55
                                              push
                                                      грр
      1183:
                   48 89 e5
                                                      rbp,rsp
                                              MOV
      1186:
                   bf 04 00 00 00
                                                      edi,0x4
                                              mov
      118b:
                   e8 b7 ff ff ff
                                              call
                                                      1147 <func a>
      1190:
                   b8 00 00 00 00
                                                      eax,0x0
                                              MOV
      1195:
                    5d
                                                      грр
                                              pop
      1196:
                    C3
                                              ret
00000000000001147 <func a>:
                                        endbr64
    1147:
                f3 Of 1e fa
   114b:
                55
                                        push
                                               грр
                48 89 e5
                                                rbp,rsp
    114c:
                                        MOV
   114f:
                48 83 ec 18
                                         sub
                                                rsp.0x18
   1153:
                89 7d ec
                                               DWORD PTR [rbp-0x14],edi
                                        mov
   1156:
                c7 45 fc 07 00 00 00
                                                DWORD PTR [rbp-0x4],0x7
                                        MOV
   115d:
                8b 55 ec
                                                edx, DWORD PTR [rbp-0x14]
                                        MOV
                8b 45 fc
   1160:
                                                eax, DWORD PTR [rbp-0x4]
                                        mov
   1163:
                01 d0
                                                eax,edx
                                         add
   1165:
                                                eax,0x1
                83 e0 01
                                         and
   1168:
                85 c0
                                         test
                                                eax.eax
   116a:
                74 Of
                                                117b <func a+0x34>
                                        je
   116c:
                8b 55 fc
                                                edx, DWORD PTR [rbp-0x4]
                                        MOV
   116f:
                8b 45 ec
                                                eax, DWORD PTR [rbp-0x14]
                                        MOV
                89 d6
   1172:
                                                esi,edx
                                        MOV
   1174:
                89 c7
                                                edi,eax
                                        MOV
                e8 ae ff ff ff
                                                1129 <func b>
                                        call
    1176:
   117b:
                90
                                        nop
   117c:
                c9
                                        leave
   117d:
                c3
                                        ret
```



```
000000000000117e <main>:
                   f3 Of 1e fa
                                             endbr64
      117e:
      1182:
                   55
                                             push
                                                     грр
      1183:
                   48 89 e5
                                                     rbp,rsp
                                             MOV
      1186:
                   bf 04 00 00 00
                                                     edi,0x4
                                             MOV
      118b:
                   e8 b7 ff ff ff
                                             call
                                                     1147 <func a>
      1190:
                   b8 00 00 00 00
                                                     eax,0x0
                                             MOV
      1195:
                   5d
                                                     грр
                                             pop
      1196:
                   C3
                                             ret
                                                                              Restarle 0x18 rsp, y guardar en rsp. por que?
00000000000001147 <func a>:
                                        endbr64
                f3 Of 1e fa
   1147:
   114b:
                55
                                        push
                                              грр
                48 89 e5
                                               rbp,rsp
   114c:
                                        MOV
   114f:
                48 83 ec 18
                                        sub
                                               rsp.0x18
   1153:
                89 7d ec
                                               DWORD PTR [rbp-0x14],edi
                                        mov
                c7 45 fc 07 00 00 00
                                               DWORD PTR [rbp-0x4],0x7
   1156:
                                        MOV
   115d:
                                               edx, DWORD PTR [rbp-0x14]
                8b 55 ec
                                        MOV
                8b 45 fc
   1160:
                                               eax, DWORD PTR [rbp-0x4]
                                        mov
   1163:
                01 d0
                                               eax,edx
                                        add
   1165:
                                               eax,0x1
                83 e0 01
                                        and
   1168:
                85 c0
                                        test
                                               eax.eax
                74 Of
                                               117b <func a+0x34>
   116a:
                                        je
   116c:
                8b 55 fc
                                               edx, DWORD PTR [rbp-0x4]
                                        MOV
   116f:
                8b 45 ec
                                               eax, DWORD PTR [rbp-0x14]
                                        MOV
                89 d6
   1172:
                                               esi,edx
                                        MOV
   1174:
                89 c7
                                               edi,eax
                e8 ae ff ff ff
                                               1129 <func b>
                                        call
   1176:
   117b:
                90
                                       nop
   117c:
                c9
                                        leave
   117d:
                c3
                                        ret
```



```
000000000000117e <main>:
                   f3 Of 1e fa
                                             endbr64
      117e:
      1182:
                   55
                                             push
                                                    грр
      1183:
                   48 89 e5
                                                    rbp,rsp
                                             MOV
      1186:
                   bf 04 00 00 00
                                                    edi,0x4
                                             MOV
      118b:
                   e8 b7 ff ff ff
                                             call
                                                    1147 <func a>
      1190:
                   b8 00 00 00 00
                                                    eax,0x0
                                             MOV
      1195:
                   5d
                                                    грр
                                             pop
      1196:
                   C3
                                             ret
00000000000001147 <func a>:
                                       endbr64
                f3 Of 1e fa
   1147:
   114b:
               55
                                       push
                                              rbp
               48 89 e5
                                              rbp,rsp
   114c:
                                       MOV
   114f:
                48 83 ec 18
                                              rsp.0x18
                                                                                           Mover 4 bytes del registro edi, al lugar de
   1153:
               89 7d ec
                                              DWORD PTR [rbp-0x14],edi
                                       MOV
   1156:
               c7 45 fc 07 00 00 00
                                              DWORD PTR [rbp-0x4],0x7
                                       MOV
                                                                                           memorias de [rbp-0x13]. Que tiene rbp?
   115d:
                                              edx, DWORD PTR [rbp-0x14]
                8b 55 ec
                                       MOV
   1160:
               8b 45 fc
                                              eax, DWORD PTR [rbp-0x4]
                                       mov
   1163:
               01 d0
                                              eax,edx
                                       add
   1165:
                                              eax,0x1
                83 e0 01
                                       and
   1168:
                85 c0
                                       test
                                              eax.eax
               74 Of
                                              117b <func a+0x34>
   116a:
                                       je
   116c:
               8b 55 fc
                                              edx, DWORD PTR [rbp-0x4]
                                       MOV
   116f:
               8b 45 ec
                                              eax, DWORD PTR [rbp-0x14]
                                       MOV
               89 d6
   1172:
                                              esi,edx
                                       MOV
   1174:
               89 c7
                                              edi,eax
               e8 ae ff ff ff
                                              1129 <func b>
                                       call
   1176:
   117b:
                90
                                       nop
   117c:
               c9
                                       leave
   117d:
                c3
                                       ret
```



```
000000000000117e <main>:
                   f3 Of 1e fa
                                             endbr64
      117e:
      1182:
                   55
                                             push
                                                     грр
      1183:
                   48 89 e5
                                                     rbp,rsp
                                             MOV
      1186:
                   bf 04 00 00 00
                                                     edi,0x4
                                             MOV
      118b:
                   e8 b7 ff ff ff
                                             call
                                                     1147 <func a>
      1190:
                   b8 00 00 00 00
                                                     eax,0x0
                                             MOV
      1195:
                   5d
                                                     грр
                                             pop
      1196:
                   C3
                                             ret
00000000000001147 <func a>:
                                        endbr64
                f3 Of 1e fa
   1147:
   114b:
                55
                                        push
                                              грр
                48 89 e5
                                               rbp,rsp
   114c:
                                        MOV
   114f:
                48 83 ec 18
                                        sub
                                               rsp.0x18
   1153:
                89 7d ec
                                               DWORD PTR [rbp-0x14],edi
                                        MOV
   1156:
                c7 45 fc 07 00 00 00
                                               DWORD PTR [rbp-0x4],0x7
                                        MOV
                                                                                                  Hacer bit operation and entre 4 bytes
   115d:
               8b 55 ec
                                               edx, DWORD PTR [rbp-0x14]
                                        MOV
                8b 45 fc
   1160:
                                               eax, DWORD PTR [rbp-0x4]
                                        mov
                                                                                                 de rax y 0x1
   1163:
                01 d0
                                               eax,edx
                                        add
   1165:
                                               eax,0x1
                83 e0 01
                                        and
   1168:
                85 c0
                                        test
                                               eax.eax
                74 Of
                                               117b <func a+0x34>
   116a:
                                        je
   116c:
                8b 55 fc
                                               edx, DWORD PTR [rbp-0x4]
                                        MOV
   116f:
                8b 45 ec
                                               eax, DWORD PTR [rbp-0x14]
                                        MOV
                89 d6
   1172:
                                               esi,edx
                                        MOV
   1174:
                89 c7
                                               edi,eax
                                        MOV
                e8 ae ff ff ff
                                               1129 <func b>
                                        call
   1176:
   117b:
                90
                                       nop
   117c:
                c9
                                        leave
   117d:
                c3
                                        ret
```



```
000000000000117e <main>:
                   f3 Of 1e fa
                                             endbr64
      117e:
      1182:
                   55
                                             push
                                                     грр
      1183:
                   48 89 e5
                                                     rbp,rsp
                                             MOV
      1186:
                   bf 04 00 00 00
                                                     edi,0x4
                                             MOV
      118b:
                   e8 b7 ff ff ff
                                             call
                                                     1147 <func a>
      1190:
                   b8 00 00 00 00
                                                     eax,0x0
                                             MOV
      1195:
                   5d
                                                     грр
                                             pop
      1196:
                   C3
                                             ret
00000000000001147 <func a>:
                                       endbr64
                f3 Of 1e fa
   1147:
   114b:
               55
                                        push
                                              rbp
               48 89 e5
                                              rbp,rsp
   114c:
                                        MOV
   114f:
                48 83 ec 18
                                        sub
                                              rsp.0x18
   1153:
               89 7d ec
                                              DWORD PTR [rbp-0x14],edi
                                        MOV
               c7 45 fc 07 00 00 00
                                              DWORD PTR [rbp-0x4],0x7
   1156:
                                        MOV
   115d:
                                              edx, DWORD PTR [rbp-0x14]
                8b 55 ec
                                        MOV
   1160:
               8b 45 fc
                                              eax, DWORD PTR [rbp-0x4]
                                        mov
   1163:
               01 d0
                                              eax,edx
                                        add
                                                                                                  Es un and, que setea el Signed Flag
   1165:
                                              eax,0x1
                83 e0 01
                                        and
   1168:
                85 c0
                                        test
                                              eax.eax
                                                                                                  (SF) si eax es negativo
               74 Of
                                              117b <func a+0x34>
   116a:
                                        ie
   116c:
                                              edx, DWORD PTR [rbp-0x4]
                8b 55 fc
                                        MOV
   116f:
               8b 45 ec
                                              eax, DWORD PTR [rbp-0x14]
                                        MOV
               89 d6
   1172:
                                              esi,edx
                                        MOV
   1174:
               89 c7
                                              edi,eax
                                        MOV
               e8 ae ff ff ff
                                              1129 <func b>
                                        call
   1176:
   117b:
                90
                                       nop
   117c:
               c9
                                        leave
   117d:
                c3
                                        ret
```



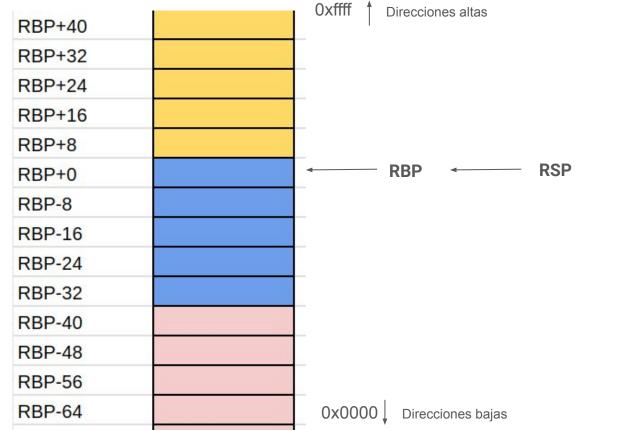
```
000000000000117e <main>:
                                              endbr64
      117e:
                    f3 Of 1e fa
      1182:
                    55
                                               push
                                                      грр
      1183:
                    48 89 e5
                                                      rbp,rsp
                                               MOV
      1186:
                    bf 04 00 00 00
                                                      edi,0x4
                                               MOV
      118b:
                    e8 b7 ff ff ff
                                              call
                                                      1147 <func a>
      1190:
                    b8 00 00 00 00
                                                      eax,0x0
                                               MOV
      1195:
                    5d
                                                      грр
                                               DOD
      1196:
                    C3
                                               ret
00000000000001147 <func a>:
                                         endbr64
                f3 Of 1e fa
    1147:
   114b:
                55
                                         push
                                                rbp
    114c:
                48 89 e5
                                                rbp,rsp
                                         MOV
    114f:
                48 83 ec 18
                                                rsp.0x18
   1153:
                                                DWORD PTR [rbp-0x14],edi
                89 7d ec
                                         MOV
                                                DWORD PTR [rbp-0x4],0x7
    1156:
                c7 45 fc 07 00 00 00
                                         MOV
   115d:
                                                edx, DWORD PTR [rbp-0x14]
                8b 55 ec
                                         MOV
   1160:
                8b 45 fc
                                                eax, DWORD PTR [rbp-0x4]
                                         mov
                01 d0
                                                eax,edx
    1163:
                                         add
   1165:
                                                eax,0x1
                83 e0 01
                                         and
    1168:
                85 c0
                                         test
                                                eax.eax
                                                117b <func a+0x34>
    116a:
                74 Of
                                         ie
   116c:
                                                edx, DWORD PTR [rbp-0x4]
                8b 55 fc
                                         MOV
   116f:
                8b 45 ec
                                                eax, DWORD PTR [rbp-0x14]
                                         MOV
                89 d6
   1172:
                                                esi,edx
                                         MOV
   1174:
                89 c7
                                                edi,eax
                e8 ae ff ff ff
                                                1129 <func b>
                                         call
    1176:
   117b:
                90
                                         nop
   117c:
                c9
                                         leave
    117d:
                c3
                                         ret
```

Es lo opuesto a ENTER, que es que saca copia rbp a rsp, y empuja lo que esta en el stack a rbp (pop rbp)

PUSH



RAX = 0x77cc



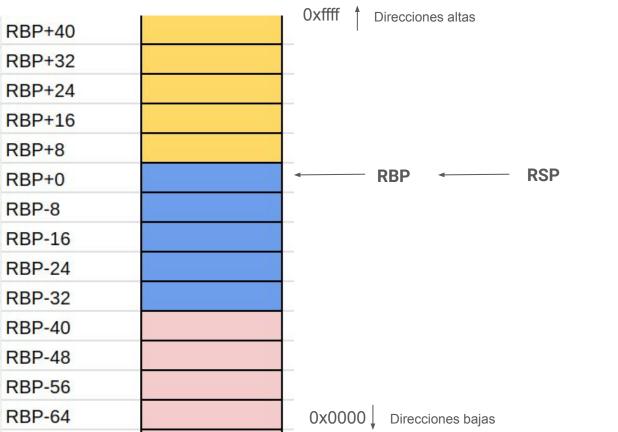
PUSH



RAX = 0x77cc

push rax

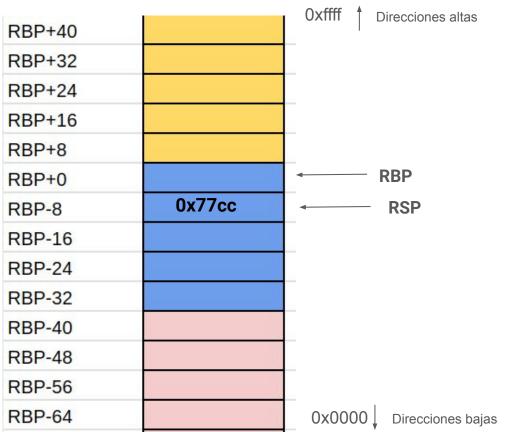
Hacer lugar para el contenido de rax en el stack, y mover el contenido de rax al stack





RAX	=	0x7	7cc
, ., .		0, .,	, 00

push rax

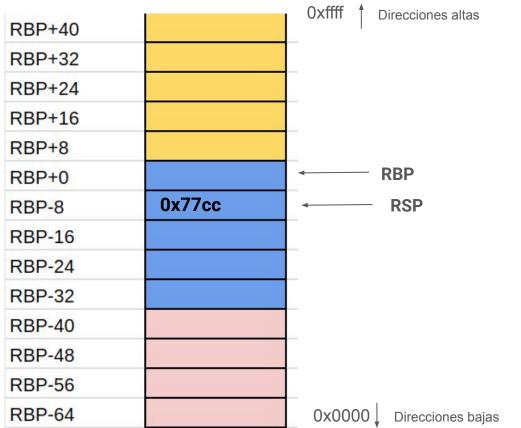




RAX = 0x77cc

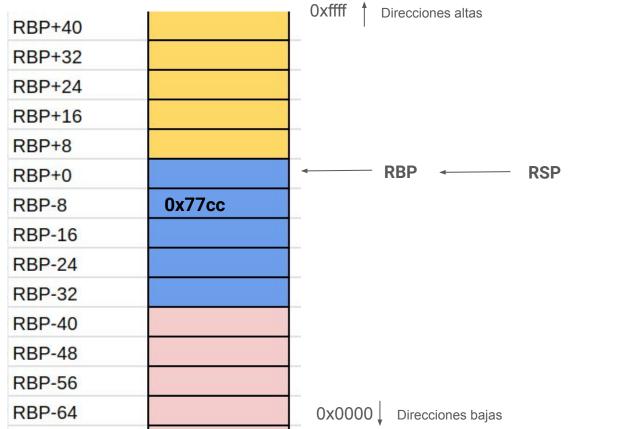
pop rdi

Mover a lo que está apuntando el stack, al lugar denominado por pop. Reducir el stack.





RAX = 0x77ccRDI = 0x77cc





Como se pasan los parámetros en el llamado de una función?

```
0000000000001147 <func a>:
                                         endbr64
   1147:
                f3 Of 1e fa
   114b:
                55
                                         push
                                               грр
   114c:
                48 89 e5
                                                rbp,rsp
                                         MOV
   114f:
                48 83 ec 18
                                         sub
                                                rsp,0x18
                89 7d ec
                                                DWORD PTR [rbp-0x14],edi
   1153:
                                         MOV
                c7 45 fc 07 00 00 00
                                                DWORD PTR [rbp-0x4],0x7
   1156:
                                         MOV
   115d:
                8b 55 ec
                                                edx, DWORD PTR [rbp-0x14]
                                         MOV
                8b 45 fc
   1160:
                                                eax, DWORD PTR [rbp-0x4]
                                         MOV
    1163:
                01 d0
                                         add
                                                eax,edx
   1165:
                83 e0 01
                                         and
                                                eax,0x1
   1168:
                85 c0
                                         test
                                                eax,eax
                                                117b <func a+0x34>
    116a:
                74 Of
                                         je
   116c:
                8b 55 fc
                                                edx, DWORD PTR [rbp-0x4]
                                         mov
   116f:
                8b 45 ec
                                                eax, DWORD PTR [rbp-0x14]
                                         MOV
   1172:
                89 d6
                                                esi,edx
                                         MOV
   1174:
                89 c7
                                                edi,eax
                                         MOV
                e8 ae ff ff ff
   1176:
                                         call
                                               1129 <func_b>
   117b:
                90
                                         nop
   117c:
                c9
                                         leave
   117d:
                c3
                                         ret
```



Como se pasan los parámetros en el llamado de una función?

```
0000000000001147 <func a>:
                                         endbr64
   1147:
                f3 Of 1e fa
   114b:
                55
                                         push
                                                грр
   114c:
                48 89 e5
                                                rbp, rsp
                                         MOV
   114f:
                48 83 ec 18
                                         sub
                                                rsp,0x18
                89 7d ec
                                                DWORD PTR [rbp-0x14],edi
   1153:
                                         MOV
                c7 45 fc 07 00 00 00
                                                DWORD PTR [rbp-0x4],0x7
   1156:
                                         MOV
   115d:
                8b 55 ec
                                                edx, DWORD PTR [rbp-0x14]
                                         MOV
                8b 45 fc
                                                eax, DWORD PTR [rbp-0x4]
    1160:
                                         MOV
    1163:
                01 d0
                                         add
                                                eax,edx
   1165:
                83 e0 01
                                         and
                                                eax,0x1
    1168:
                85 c0
                                         test
                                                eax,eax
                                                117b <func a+0x34>
    116a:
                74 Of
                                         je
                8b 55 fc
                                                edx, DWORD PTR [rbp-0x4]
    116c:
                                         mov
   116f:
                8b 45 ec
                                                eax, DWORD PTR [rbp-0x14]
                                         MOV
   1172:
                89 d6
                                                esi,edx
                                         MOV
   1174:
                89 c7
                                                edi,eax
                                         MOV
                e8 ae ff ff ff
                                         call
                                                1129 <func_b>
   1176:
   117b:
                90
                                         nop
                c9
                                         leave
    117c:
   117d:
                c3
                                         ret
```

Cómo se retorna un valor?



```
0000000000001147 <func a>:
                                         endbr64
    1147:
                f3 Of 1e fa
   114b:
                55
                                         push
                                                грр
                                                rbp,rsp
   114c:
                48 89 e5
                                         MOV
   114f:
                48 83 ec 18
                                         sub
                                                rsp,0x18
                89 7d ec
                                                DWORD PTR [rbp-0x14],edi
   1153:
                                         MOV
   1156:
                                                DWORD PTR [rbp-0x4],0x7
                c7 45 fc 07 00 00 00
                                         MOV
   115d:
                8b 55 ec
                                                edx, DWORD PTR [rbp-0x14]
                                         MOV
                8b 45 fc
                                                eax, DWORD PTR [rbp-0x4]
    1160:
                                         MOV
    1163:
                01 d0
                                         add
                                                eax,edx
   1165:
                83 e0 01
                                         and
                                                eax,0x1
   1168:
                85 c0
                                         test
                                                eax,eax
   116a:
                74 Of
                                                117b <func a+0x34>
                                         je
                8b 55 fc
                                                edx, DWORD PTR [rbp-0x4]
    116c:
                                         mov
   116f:
                8b 45 ec
                                                eax, DWORD PTR [rbp-0x14]
                                         MOV
   1172:
                89 d6
                                                esi,edx
                                         MOV
   1174:
                89 c7
                                                edi,eax
                                         MOV
                e8 ae ff ff ff
                                         call
                                                1129 <func b>
   1176:
   117b:
                90
                                         nop
                c9
                                         leave
    117c:
   117d:
                c3
                                         ret
```

Parametros de integers:

- **%rdi** primer parametro
- **%rsi** segundo parametro
- %rdx tercer parametro
- %rcx cuarto parametro
- %r8 quinto parametro
- **%r9** sexto parametro
- %xmm0-7 8 parametros de float



```
0000000000001147 <func a>:
                                         endbr64
    1147:
                f3 Of 1e fa
   114b:
                55
                                         push
                                                грр
   114c:
                48 89 e5
                                                rbp, rsp
                                         MOV
   114f:
                48 83 ec 18
                                         sub
                                                rsp,0x18
                89 7d ec
   1153:
                                                DWORD PTR [rbp-0x14].edi
                                         MOV
   1156:
                c7 45 fc 07 00 00 00
                                                DWORD PTR [rbp-0x4],0x7
                                         MOV
   115d:
                8b 55 ec
                                                edx, DWORD PTR [rbp-0x14]
                                         MOV
                8b 45 fc
                                                eax, DWORD PTR [rbp-0x4]
    1160:
                                         MOV
    1163:
                01 d0
                                         add
                                                eax,edx
   1165:
                83 e0 01
                                         and
                                                eax,0x1
   1168:
                85 c0
                                         test
                                                eax, eax
   116a:
                74 Of
                                                117b <func a+0x34>
                                         je
                8b 55 fc
                                                edx, DWORD PTR [rbp-0x4]
    116c:
                                         mov
   116f:
                8b 45 ec
                                                eax, DWORD PTR [rbp-0x14]
                                         MOV
   1172:
                89 d6
                                                esi,edx
                                         MOV
   1174:
                89 c7
                                                edi,eax
                                         MOV
                e8 ae ff ff ff
                                         call
                                                1129 <func b>
    1176:
   117b:
                90
                                         nop
                c9
                                         leave
    117c:
   117d:
                c3
                                         ret
```

Parametros de integers:

- **%rdi** primer parametro
- %rsi segundo parametro
- %rdx tercer parametro
- %rcx cuarto parametro
- %r8 quinto parametro
- **%r9** sexto parametro
- %xmm0-7 8 parametros de float

Y si hay mas?



```
0000000000001147 <func a>:
                                         endbr64
    1147:
                f3 Of 1e fa
   114b:
                55
                                         push
                                                грр
                                                rbp,rsp
   114c:
                48 89 e5
                                         MOV
   114f:
                48 83 ec 18
                                         sub
                                                rsp,0x18
                89 7d ec
   1153:
                                                DWORD PTR [rbp-0x14].edi
                                         MOV
   1156:
                c7 45 fc 07 00 00 00
                                                DWORD PTR [rbp-0x4],0x7
                                         MOV
   115d:
                8b 55 ec
                                                edx, DWORD PTR [rbp-0x14]
                                         MOV
                8b 45 fc
                                                eax, DWORD PTR [rbp-0x4]
    1160:
                                         MOV
    1163:
                01 d0
                                         add
                                                eax,edx
   1165:
                83 e0 01
                                         and
                                                eax,0x1
   1168:
                85 c0
                                         test
                                                eax, eax
                74 Of
                                                117b <func a+0x34>
    116a:
                                         je
                8b 55 fc
                                                edx, DWORD PTR [rbp-0x4]
    116c:
                                         mov
   116f:
                8b 45 ec
                                                eax, DWORD PTR [rbp-0x14]
                                         MOV
   1172:
                89 d6
                                                esi.edx
                                         MOV
   1174:
                89 c7
                                                edi,eax
                                         MOV
                e8 ae ff ff ff
                                         call
                                                1129 <func_b>
    1176:
   117b:
                90
                                         nop
                c9
                                         leave
    117c:
   117d:
                c3
                                         ret
```

Parametros de integers:

- **%rdi** primer parametro
- **%rsi** segundo parametro
- **%rdx** tercer parametro
- **%rcx** cuarto parametro
- %r8 quinto parametro
- **%r9** sexto parametro
- **%xmm0-7** 8 parametros de float

Por que %esi, %edi?



```
Parametros de integers:
0000000000001147 <func a>:
                                                                                      %rdi primer parametro
                                      endbr64
   1147:
               f3 Of 1e fa
   114b:
               55
                                      push
                                            грр
                                                                                      %rsi segundo parametro
                                            rbp,rsp
   114c:
               48 89 e5
                                      MOV
                                                                                      %rdx tercer parametro
   114f:
               48 83 ec 18
                                      sub
                                            rsp,0x18
               89 7d ec
   1153:
                                            DWORD PTR [rbp-0x14].edi
                                      MOV
                                                                                      %rcx cuarto parametro
                                            DWORD PTR [rbp-0x4],0x7
   1156:
               c7 45 fc 07 00 00 00
                                      MOV
   115d:
               8b 55 ec
                                            edx, DWORD PTR [rbp-0x14]
                                                                                             quinto parametro
                                      MOV
               8b 45 fc
                                             eax, DWORD PTR [rbp-0x4]
   1160:
                                      MOV
                                                                                             sexto parametro
   1163:
               01 d0
                                      add
                                             eax,edx
                                                                                      %xmm0-7 8 parametros de float
   1165:
               83 e0 01
                                      and
                                            eax,0x1
   1168:
               85 c0
                                      test
                                            eax,eax
               74 Of
                                            117b <func a+0x34>
   116a:
                                      je
               8b 55 fc
                                            edx, DWORD PTR [rbp-0x4]
   116c:
                                      mov
   116f:
               8b 45 ec
                                             eax, DWORD PTR [rbp-0x14]
                                      MOV
   1172:
               89 d6
                                             esi.edx
                                      MOV
   1174:
               89 c7
                                             edi,eax
                                      MOV
               e8 ae ff ff ff
                                      call
                                            1129 <func_b>
   1176:
   117b:
               90
                                      nop
                                      leave
   117c:
               c9
   117d:
                                                                                           Son ints de 4 bytes
               c3
                                      ret
```



6	90000000000000	147 <func_a>:</func_a>		
	1147:	f3 Of 1e fa	endbr64	
	114b:	55	push rbp	
	114c:	48 89 e5	mov rbp,rsp	
	114f:	48 83 ec 18	sub rsp,0x18	
	1153:	89 7d ec	mov DWORD PTR [rbp-0x14],edi	
	1156:	c7 45 fc 07 00 00 00	mov DWORD PTR [rbp-0x4],0x7	
	115d:	8b 55 ec	mov edx,DWORD PTR [rbp-0x14]	
	1160:	8b 45 fc	mov eax,DWORD PTR [rbp-0x4]	
	1163:	01 d0	add eax,edx	
	1165:	83 e0 01	and eax,0x1	
	1168:	85 c0	test eax,eax	
	116a:	74 Of	je 117b <func_a+0x34></func_a+0x34>	
	116c:	8b 55 fc	mov edx, DWORD PTR [rbp-0x4]	
	116f:	8b 45 ec	mov eax,DWORD PTR [rbp-0x14]	
	1172:	89 d6	mov esi,edx	
	1174:	89 c7	mov edi,eax	
	1176:	e8 ae ff ff ff	call 1129 <func_b></func_b>	
	117b:	90	nop	
	117c:	c9	leave	
	117d:	c3	ret	

Como sabe donde volver después del call?



```
0000000000001147 <func a>:
                                         endbr64
    1147:
                f3 Of 1e fa
   114b:
                 55
                                         push
                                                грр
                                                 rbp,rsp
    114c:
                48 89 e5
                                          mov
    114f:
                48 83 ec 18
                                          sub
                                                 rsp,0x18
                89 7d ec
   1153:
                                                 DWORD PTR [rbp-0x14].edi
                                          MOV
                                                 DWORD PTR [rbp-0x4],0x7
    1156:
                c7 45 fc 07 00 00 00
                                          MOV
   115d:
                                                 edx, DWORD PTR [rbp-0x14]
                8b 55 ec
                                          MOV
                8b 45 fc
                                                 eax, DWORD PTR [rbp-0x4]
    1160:
                                          MOV
    1163:
                01 d0
                                          add
                                                 eax,edx
   1165:
                83 e0 01
                                         and
                                                 eax,0x1
    1168:
                85 c0
                                          test
                                                 eax, eax
                                                 117b <func a+0x34>
    116a:
                74 Of
                                         je
                                                 edx, DWORD PTR [rbp-0x4]
    116c:
                8b 55 fc
                                          mov
                8b 45 ec
                                                 eax, DWORD PTR [rbp-0x14]
    116f:
                                          MOV
   1172:
                89 d6
                                                 esi.edx
                                          mov
    1174:
                89 c7
                                                 edi,eax
                                          MOV
                e8 ae ff ff ff
                                         call
                                                1129 <func b>
    1176:
   117b:
                90
                                          nop
                                         leave
    117c:
                c9
    117d:
                c3
                                          ret
```

CALL <label>: hace un push de **%rip** al stack y hace un jump incondicional a la direccion del label. **%rip** contiene la dirección de la instrucción justo después del call.



```
0000000000001147 <func a>:
                f3 Of 1e fa
                                          endbr64
    1147:
   114b:
                 55
                                          push
                                                 грр
                                                 rbp,rsp
    114c:
                48 89 e5
                                          mov
    114f:
                48 83 ec 18
                                          sub
                                                 rsp,0x18
                89 7d ec
   1153:
                                                 DWORD PTR [rbp-0x14].edi
                                          MOV
                                                 DWORD PTR [rbp-0x4],0x7
    1156:
                c7 45 fc 07 00 00 00
                                          MOV
   115d:
                                                 edx, DWORD PTR [rbp-0x14]
                8b 55 ec
                                          MOV
                8b 45 fc
                                                 eax, DWORD PTR [rbp-0x4]
    1160:
                                          MOV
    1163:
                01 d0
                                          add
                                                 eax,edx
   1165:
                83 e0 01
                                          and
                                                 eax,0x1
    1168:
                85 c0
                                          test
                                                 eax, eax
                                                 117b <func a+0x34>
    116a:
                74 Of
                                          je
                                                 edx, DWORD PTR [rbp-0x4]
    116c:
                8b 55 fc
                                          mov
                8b 45 ec
                                                 eax, DWORD PTR [rbp-0x14]
    116f:
                                          MOV
   1172:
                89 d6
                                                 esi.edx
                                          mov
    1174:
                89 c7
                                                 edi,eax
                                          MOV
                e8 ae ff ff ff
                                          call
                                                 1129 <func b>
    1176:
   117b:
                90
                                          nop
                                          leave
    117c:
                c9
    117d:
                c3
                                          ret
```

CALL <label>: hace un push de **%rip** al stack y hace un jump incondicional a la direccion del label. **%rip** contiene la dirección de la instrucción justo después del call.

ret: hace un pop de %rsp a %rip.



La convencion que estamos mostrando es:

System V AMD 64 ABI (linux)

Hay otras

(https://en.wikipedia.org/wiki/X86_calling_conventions):

- Microsoft x86 calling conventions
- X32 ABI o cdecl (Linux)



La convencion que estamos mostrando es:

System V AMD 64 ABI (linux)

Hay otras

(https://en.wikipedia.org/wiki/X86_calling_conventions):

- Microsoft x86 calling conventions
- X32 ABI o cdecl (Linux)

Por qué es importante seguir la convención?



Ejemplo completo:

Objdump -M intel -d simple



RBP+0		1	•	-RBP	•	-RSP
RBP-8						
RBP-16						
RBP-24						
RBP-32						
RBP-40						
RBP-48						
RBP-56						
RBP-64						
RBP-72						
RBP-80						
RBP-88						
RBP-96						
RBP-104						
RBP-112						
RBP-120						
RBP-128						

	0x555555551a7	<main></main>	endbr6	4
	0x555555551ab		push	rbp +
	0x555555551ac		mov	rbp,rsp
D	0x555555551af	<main+8></main+8>	sub	rsp,0x10
Ρ	0x555555551b3		mov	rax, OWORD PTR fs:0x28
	0x555555551bc		MOV	OWORD PTR [rbp-0x8],rax
	0x5555555551c0		хог	eax,eax
	0x555555551c2	<main+27></main+27>	mov	DWORD PTR [rbp-0x10],0x0
	0x555555551c9	<main+34></main+34>	lea	rax,[rbp-0x10]
	0x555555551cd	<main+38></main+38>	mov	rdx rax
	0x555555551d0	<main+41></main+41>	MOV	esi.0x8
	0x555555551d5	<main+46></main+46>	mov	edi.0x4
	0x555555551da	<main+51></main+51>	call	0x555555555169 <div></div>
	0x555555551df	<main+56></main+56>	mov	DWORD PTR [rbp-0xc],eax
	0x555555551e2	<main+59></main+59>	CMD	DWORD PTR [rbp-0xc],0x0
	0x555555551e6	<main+63></main+63>	ie	0x555555555201 <main+90></main+90>
	0x555555551e8	<main+65></main+65>	mov	eax, DWORD PTR [rbp-0x10]
	0x555555551eb	<main+68></main+68>	mov	esi.eax
	0x555555551ed	<main+70></main+70>	lea	rax,[rip+0xe10] # 0x55555556004
	0x555555551f4	<main+77></main+77>	mov	rdi rax
	0x5555555551f7	<main+80></main+80>	mov	eax,0x0
	0x555555551fc	<main+85></main+85>	call	0x555555555070 <printf@plt></printf@plt>
	0x555555555201	<main+90></main+90>	mov	eax,0x0
	0x555555555206	<main+95></main+95>	mov	rdx,QWORD PTR [rbp-0x8]
	0x5555555520a	<main+99></main+99>	sub	rdx, OWORD PTR fs:0x28
	0x555555555213	<main+108></main+108>	je	0x55555555521a <main+115></main+115>
	0x555555555215	<main+110></main+110>	call	0x555555555060 < _stack_chk_fail@plt>
	0x5555555521a	<main+115></main+115>	leave	The state of the s
			Landay (1.3)	

rip	55551ab	inst pointer				
rsp	fff1000	stack pointer	rax	return	rsi	2ndo arg
rbp	fff1000	base pointer	rdi	1er arg	rdx	3er arg



RBP-8	SIACK		
RBP-16 RBP-24 RBP-32 RBP-40 RBP-48 RBP-56 RBP-64 RBP-72 RBP-80 RBP-80 RBP-88 RBP-96 RBP-104 RBP-112 RBP-120	RBP+0		← RBP
RBP-24 RBP-32 RBP-40 RBP-48 RBP-56 RBP-64 RBP-72 RBP-80 RBP-88 RBP-96 RBP-104 RBP-112 RBP-120	RBP-8	fff1000	← RSP
RBP-32 RBP-40 RBP-48 RBP-56 RBP-64 RBP-72 RBP-80 RBP-88 RBP-96 RBP-104 RBP-112 RBP-120	RBP-16		
RBP-40 RBP-48 RBP-56 RBP-64 RBP-72 RBP-80 RBP-88 RBP-96 RBP-104 RBP-112 RBP-120	RBP-24		
RBP-48 RBP-56 RBP-64 RBP-72 RBP-80 RBP-88 RBP-96 RBP-104 RBP-112 RBP-120	RBP-32		
RBP-56 RBP-64 RBP-72 RBP-80 RBP-88 RBP-96 RBP-104 RBP-112 RBP-120	RBP-40		
RBP-64 RBP-72 RBP-80 RBP-88 RBP-96 RBP-104 RBP-112 RBP-120	RBP-48		
RBP-72 RBP-80 RBP-88 RBP-96 RBP-104 RBP-112 RBP-120	RBP-56		
RBP-80 RBP-88 RBP-96 RBP-104 RBP-112 RBP-120	RBP-64		
RBP-88 RBP-96 RBP-104 RBP-112 RBP-120	RBP-72		
RBP-96 RBP-104 RBP-112 RBP-120	RBP-80		
RBP-104 RBP-112 RBP-120	RBP-88		
RBP-112 RBP-120	RBP-96		
RBP-120	RBP-104		
	RBP-112		
RBP-128	RBP-120		
	RBP-128		

,0000x0	Direcciones	bajas
---------	-------------	-------

_				Jan
	0x555555551a7	<main></main>	endbr6	4
	0x555555551ab	<main+4></main+4>	push	rbp
	0x555555551ac	<main+5></main+5>	MOV	rbp,rsp ←
	0x555555551af	<main+8></main+8>	sub	rsp,0x10
	0x555555551b3	<main+12></main+12>	mov	rax,QWORD PTR fs:0x28
	0x555555551bc	<main+21></main+21>	MOV	QWORD PTR [rbp-0x8],rax
	0x555555551c0	<main+25></main+25>	хог	eax,eax
	0x555555551c2	<main+27></main+27>	MOV	DWORD PTR [rbp-0x10],0x0
	0x555555551c9	<main+34></main+34>	lea	rax,[rbp-0x10]
	0x555555551cd	<main+38></main+38>	MOV	rdx,rax
	0x555555551d0	<main+41></main+41>	MOV	esi,0x8
	0x555555551d5	<main+46></main+46>	MOV	edi,0x4
	0x555555551da	<main+51></main+51>	call	0x555555555169 <div></div>
	0x555555551df	<main+56></main+56>	MOV	DWORD PTR [rbp-0xc],eax
	0x555555551e2	<main+59></main+59>	стр	DWORD PTR [rbp-0xc],0x0
	0x555555551e6	<main+63></main+63>	je	0x555555555201 <main+90></main+90>
	0x555555551e8	<main+65></main+65>	mov	eax, DWORD PTR [rbp-0x10]
	0x555555551eb	<main+68></main+68>	MOV	esi,eax
	0x555555551ed	<main+70></main+70>	lea	rax,[rip+0xe10] # 0x55555556004
	0x555555551f4	<main+77></main+77>	mov	rdi,rax
	0x5555555551f7	<main+80></main+80>	mov	eax,0x0
	0x555555551fc	<main+85></main+85>	call	0x555555555070 <printf@plt></printf@plt>
	0x555555555201	<main+90></main+90>	MOV	eax,0x0
	0x55555555206	<main+95></main+95>	mov	rdx,QWORD PTR [rbp-0x8]
	0x5555555520a	<main+99></main+99>	sub	rdx.OWORD PTR fs:0x28
	0x555555555213	<main+108></main+108>	je	0x55555555521a <main+115></main+115>
	0x555555555215	<main+110></main+110>	call	0x555555555060 < stack chk fail@plt>
	0x55555555521a	<main+115></main+115>	leave	

rip	55551ac	inst pointer				
rsp	fffOff8	stack pointer	rax	return	rsi	2ndo arg
rbp	fff0000	base pointer	rdi	1er arg	rdx	3er arg



STACK

fff1000

RSP

	0x555555551a7	<main></main>	endbr6	4
	0x555555551ab	<main+4></main+4>	push	грр
	0x555555551ac	<main+5></main+5>	MOV	rbp,rsp
	0x555555551af	<main+8></main+8>	sub	rsp,0x10
_	0x555555551b3	<main+12></main+12>	mov	rax,QWORD PTR fs:0x28
Ρ	0x555555551bc	<main+21></main+21>	mov	QWORD PTR [rbp-0x8],rax
	0x555555551c0	<main+25></main+25>	XOL	eax,eax
	0x555555551c2	<main+27></main+27>	mov	DWORD PTR [rbp-0x10],0x0
	0x555555551c9	<main+34></main+34>	lea	rax,[rbp-0x10]
	0x555555551cd	<main+38></main+38>	mov	rdx,rax
	0x555555551d0	<main+41></main+41>	MOV	esi,0x8
	0x555555551d5	<main+46></main+46>	mov	edi,0x4
	0x555555551da	<main+51></main+51>	call	0x55555555169 <div></div>
	0x555555551df	<main+56></main+56>	mov	DWORD PTR [rbp-0xc],eax
	0x555555551e2	<main+59></main+59>	cmp	DWORD PTR [rbp-0xc],0x0
	0x555555551e6	<main+63></main+63>	je	0x555555555201 <main+90></main+90>
	0x555555551e8	<main+65></main+65>	mov	eax,DWORD PTR [rbp-0x10]
	0x555555551eb	<main+68></main+68>	MOV	esi,eax
	0x555555551ed	<main+70></main+70>	lea	rax,[rip+0xe10] # 0x55555556004
	0x555555551f4	<main+77></main+77>	mov	rdi,rax
	0x5555555551f7	<main+80></main+80>	mov	eax,0x0
	0x555555551fc	<main+85></main+85>	call	0x555555555070 <printf@plt></printf@plt>
	0x555555555201	<main+90></main+90>	mov	eax,0x0
	0x555555555206	<main+95></main+95>	mov	rdx,QWORD PTR [rbp-0x8]
	0x5555555520a	<main+99></main+99>	sub	rdx,QWORD PTR fs:0x28
	0x555555555213	<main+108></main+108>	je	0x55555555521a <main+115></main+115>
	0x555555555215	<main+110></main+110>	call	0x555555555060 <stack_chk_fail@plt></stack_chk_fail@plt>
	0x55555555521a	<main+115></main+115>	leave	The state of the s
			Landon, Maria	

rip	55551af	inst pointer				
rsp	fffOff8	stack pointer	rax	return	rsi	2ndo arg
rbp	fff0ff8	base pointer	rdi	1er arg	rdx	3er arg

0x0000 Direcciones bajas



SIACK		
RBP+0		
RBP+0	fff1000	← RBP
RBP-8		
RBP-16		← RSI
RBP-24		
RBP-32		
RBP-40		
RBP-48		
RBP-56		
RBP-64		
RBP-72		
RBP-80		
RBP-88		
RBP-96		
RBP-104		
RBP-112		
RBP-120		

0x0000 L	Direcciones	bajas
----------	-------------	-------

_				Jan
	0x555555551a7	<main></main>	endbr6	4
	0x555555551ab	<main+4></main+4>	push	гЬр
	0x555555551ac	<main+5></main+5>	MOV	rbp,rsp
	0x555555551af	<main+8></main+8>	sub	rsp,0x10
	0x555555551b3	<main+12></main+12>	MOV	rax,QWORD PTR fs:0x28 ←
	0x555555551bc	<main+21></main+21>	MOV	QWORD PTR [rbp-0x8],rax
	0x555555551c0	<main+25></main+25>	XOL	eax,eax
	0x555555551c2	<main+27></main+27>	MOV	DWORD PTR [rbp-0x10],0x0
	0x555555551c9	<main+34></main+34>	lea	rax,[rbp-0x10]
	0x555555551cd	<main+38></main+38>	MOV	rdx,rax
	0x555555551d0	<main+41></main+41>	MOV	esi,0x8
	0x555555551d5	<main+46></main+46>	mov	edi,0x4
	0x555555551da	<main+51></main+51>	call	0x555555555169 <div></div>
	0x555555551df	<main+56></main+56>	mov	DWORD PTR [rbp-0xc],eax
	0x555555551e2	<main+59></main+59>	cmp	DWORD PTR [rbp-0xc],0x0
	0x555555551e6	<main+63></main+63>	je	0x555555555201 <main+90></main+90>
	0x555555551e8	<main+65></main+65>	mov	eax,DWORD PTR [rbp-0x10]
	0x555555551eb	<main+68></main+68>	MOV	esi,eax
	0x555555551ed	<main+70></main+70>	lea	rax,[rip+0xe10] # 0x55555556004
	0x555555551f4	<main+77></main+77>	mov	rdi,rax
	0x5555555551f7	<main+80></main+80>	MOV	eax,0x0
	0x555555551fc	<main+85></main+85>	call	0x555555555070 <printf@plt></printf@plt>
	0x55555555201	<main+90></main+90>	MOV	eax,0x0
	0x55555555206	<main+95></main+95>	MOV	rdx,QWORD PTR [rbp-0x8]
	0x5555555520a	<main+99></main+99>	sub	rdx, QWORD PTR fs:0x28
	0x555555555213	<main+108></main+108>	je	0x55555555521a <main+115></main+115>
	0x555555555215	<main+110></main+110>	call	0x555555555060 < stack chk fail@plt>
	0x55555555521a	<main+115></main+115>	leave	

rip	55551b3	inst pointer				
rsp	fff0fe8	stack pointer	rax	return	rsi	2ndo arg
rbp	fff0ff8	base pointer	rdi	1er arg	rdx	3er arg



STACK		
RBP+0		
RBP+0	fff1000	← RBP
RBP-8	Ptr fs:0x28	
RBP-16		←RSP
RBP-24		
RBP-32		
RBP-40		
RBP-48		
RBP-56		
RBP-64		
RBP-72		
RBP-80		
RBP-88		
RBP-96		
RBP-104		
RBP-112		
RBP-120		

1	
0x0000	Direcciones bajas

			Jan
0x555555551a7	<main></main>	endbr6	4
0x555555551ab	<main+4></main+4>	push	грр
0x555555551ac	<main+5></main+5>	MOV	rbp,rsp
0x555555551af	<main+8></main+8>	sub	rsp,0x10
0x555555551b3	<main+12></main+12>	mov	rax,QWORD PTR fs:0x28
0x555555551bc	<main+21></main+21>	mov	QWORD PTR [rbp-0x8],rax
0x555555551c0	<main+25></main+25>	XOL	eax,eax -
0x555555551c2	<main+27></main+27>	mov	DWORD PTR [rbp-0x10],0x0
0x555555551c9	<main+34></main+34>	lea	rax,[rbp-0x10]
0x555555551cd	<main+38></main+38>	MOV	rdx,rax
0x555555551d0	<main+41></main+41>	MOV	esi,0x8
0x555555551d5	<main+46></main+46>	mov	edi,0x4
0x555555551da	<main+51></main+51>	call	0x555555555169 <div></div>
0x555555551df	<main+56></main+56>	MOV	DWORD PTR [rbp-0xc],eax
0x555555551e2	<main+59></main+59>	cmp	DWORD PTR [rbp-0xc],0x0
0x555555551e6	<main+63></main+63>	je	0x555555555201 <main+90></main+90>
0x555555551e8	<main+65></main+65>	mov	eax,DWORD PTR [rbp-0x10]
0x555555551eb	<main+68></main+68>	MOV	esi,eax
0x555555551ed	<main+70></main+70>	lea	rax,[rip+0xe10] # 0x55555556004
0x555555551f4	<main+77></main+77>	mov	rdi,rax
0x555555551f7	<main+80></main+80>	MOV	eax,0x0
0x555555551fc	<main+85></main+85>	call	0x555555555070 <printf@plt></printf@plt>
0x55555555201	<main+90></main+90>	MOV	eax,0x0
0x55555555206	<main+95></main+95>	mov	rdx,QWORD PTR [rbp-0x8]
0x5555555520a	<main+99></main+99>	sub	rdx,QWORD PTR fs:0x28
0x55555555213	<main+108></main+108>	je	0x55555555521a <main+115></main+115>
0x55555555215	<main+110></main+110>	call	0x555555555060 <stack_chk_fail@plt></stack_chk_fail@plt>
0x5555555521a	<main+115></main+115>	leave	
		t ring it. I	

rip	55551c0	inst pointer					
rsp	fff0fe8	stack pointer	rax	Ptr fs:0x28	return	rsi	2ndo arg
rbp	fff0ff8	base pointer	rdi		1er arg	rdx	3er arg



STACK		
RBP+0		
RBP+0	fff1000	← RBP
RBP-8	Ptr fs:0x28	
RBP-16		←RSP
RBP-24		
RBP-32		
RBP-40		
RBP-48		
RBP-56		
RBP-64		
RBP-72		
RBP-80		
RBP-88		
RBP-96		
RBP-104		
RBP-112		
RBP-120		

1	
0x0000	Direcciones bajas

			Jan
0x555555551a7	<main></main>	endbr6	4
0x555555551ab	<main+4></main+4>	push	грр
0x555555551ac	<main+5></main+5>	MOV	rbp,rsp
0x555555551af	<main+8></main+8>	sub	rsp,0x10
0x555555551b3	<main+12></main+12>	mov	rax,QWORD PTR fs:0x28
0x555555551bc	<main+21></main+21>	mov	QWORD PTR [rbp-0x8],rax
0x555555551c0	<main+25></main+25>	XOL	eax,eax -
0x555555551c2	<main+27></main+27>	mov	DWORD PTR [rbp-0x10],0x0
0x555555551c9	<main+34></main+34>	lea	rax,[rbp-0x10]
0x555555551cd	<main+38></main+38>	MOV	rdx,rax
0x555555551d0	<main+41></main+41>	MOV	esi,0x8
0x555555551d5	<main+46></main+46>	mov	edi,0x4
0x555555551da	<main+51></main+51>	call	0x555555555169 <div></div>
0x555555551df	<main+56></main+56>	MOV	DWORD PTR [rbp-0xc],eax
0x555555551e2	<main+59></main+59>	cmp	DWORD PTR [rbp-0xc],0x0
0x555555551e6	<main+63></main+63>	je	0x555555555201 <main+90></main+90>
0x555555551e8	<main+65></main+65>	mov	eax,DWORD PTR [rbp-0x10]
0x555555551eb	<main+68></main+68>	MOV	esi,eax
0x555555551ed	<main+70></main+70>	lea	rax,[rip+0xe10] # 0x55555556004
0x555555551f4	<main+77></main+77>	mov	rdi,rax
0x555555551f7	<main+80></main+80>	MOV	eax,0x0
0x555555551fc	<main+85></main+85>	call	0x555555555070 <printf@plt></printf@plt>
0x55555555201	<main+90></main+90>	MOV	eax,0x0
0x55555555206	<main+95></main+95>	mov	rdx,QWORD PTR [rbp-0x8]
0x5555555520a	<main+99></main+99>	sub	rdx,QWORD PTR fs:0x28
0x55555555213	<main+108></main+108>	je	0x55555555521a <main+115></main+115>
0x55555555215	<main+110></main+110>	call	0x555555555060 <stack_chk_fail@plt></stack_chk_fail@plt>
0x5555555521a	<main+115></main+115>	leave	
		t ring it. I	

rip	55551c0	inst pointer					
rsp	fff0fe8	stack pointer	rax	Ptr fs:0x28	return	rsi	2ndo arg
rbp	fff0ff8	base pointer	rdi		1er arg	rdx	3er arg



STACK		
RBP+0		
RBP+0	fff1000	← RBP
RBP-8	Ptr fs:0x28	
RBP-16		←RSP
RBP-24		
RBP-32		
RBP-40		
RBP-48		
RBP-56		
RBP-64		
RBP-72		
RBP-80		
RBP-88		
RBP-96		
RBP-104		
RBP-112		
RBP-120		

0x555555551a7	<main></main>	endbr6	4
0x555555551ab	<main+4></main+4>	push	rbp
0x555555551ac	<main+5></main+5>	MOV	rbp,rsp
0x555555551af	<main+8></main+8>	sub	rsp,0x10
0x555555551b3	<main+12></main+12>	mov	rax,QWORD PTR fs:0x28
0x555555551bc	<main+21></main+21>	MOV	QWORD PTR [rbp-0x8],rax
0x555555551c0	<main+25></main+25>	XOL	eax,eax
0x555555551c2	<main+27></main+27>	mov	DWORD PTR [rbp-0x10],0x0 ←
0x555555551c9	<main+34></main+34>	lea	rax,[rbp-0x10]
0x555555551cd	<main+38></main+38>	MOV	rdx,rax
0x555555551d0	<main+41></main+41>	MOV	esi,0x8
0x555555551d5	<main+46></main+46>	MOV	edi,0x4
0x555555551da	<main+51></main+51>	call	0x555555555169 <div></div>
0x555555551df	<main+56></main+56>	MOV	DWORD PTR [rbp-0xc],eax
0x555555551e2	<main+59></main+59>	CMP	DWORD PTR [rbp-0xc],0x0
0x555555551e6	<main+63></main+63>	je	0x555555555201 <main+90></main+90>
0x555555551e8	<main+65></main+65>	mov	eax,DWORD PTR [rbp-0x10]
0x555555551eb	<main+68></main+68>	MOV	esi,eax
0x555555551ed	<main+70></main+70>	lea	rax,[rip+0xe10] # 0x55555556004
0x555555551f4	<main+77></main+77>	mov	rdi,rax
0x555555551f7	<main+80></main+80>	MOV	eax,0x0
0x555555551fc	<main+85></main+85>	call	0x555555555070 <printf@plt></printf@plt>
0x55555555201	<main+90></main+90>	MOV	eax,0x0
0x55555555206	<main+95></main+95>	mov	rdx,QWORD PTR [rbp-0x8]
0x5555555520a	<main+99></main+99>	sub	rdx,QWORD PTR fs:0x28
0x555555555213	<main+108></main+108>	je	0x55555555521a <main+115></main+115>
0x555555555215	<main+110></main+110>	call	<pre>0x555555555060 <stack_chk_fail@plt></stack_chk_fail@plt></pre>
0x55555555521a	<main+115></main+115>	leave	

rip	55551c2	inst pointer					
rsp	fff0fe8	stack pointer	rax	0	return	rsi	2ndo arg
rbp	fff0ff8	base pointer	rdi		1er arg	rdx	3er arg



2ndo arg

3er arg

STACK

• RBP
x28
₹

Direcciones bajas

0x555555551c2	<main+27></main+27>	MOV	DWORD PTR [rbp-0x10],0x0
0x555555551c9	<main+34></main+34>	lea	rax,[rbp-0x10]
0x555555551cd	<main+38></main+38>	MOV	rdx,rax
0x555555551d0	<main+41></main+41>	MOV	esi,0x8
0x555555551d5	<main+46></main+46>	mov	edi,0x4
0x555555551da	<main+51></main+51>	call	0x555555555169 <div></div>
0x555555551df	<main+56></main+56>	mov	DWORD PTR [rbp-0xc],eax
0x555555551e2	<main+59></main+59>	CMP	DWORD PTR [rbp-0xc],0x0
0x555555551e6	<main+63></main+63>	je	0x555555555201 <main+90></main+90>
0x555555551e8	<main+65></main+65>	mov	eax, DWORD PTR [rbp-0x10]
0x555555551eb	<main+68></main+68>	mov	esi,eax
0x555555551ed	<main+70></main+70>	lea	rax.[rip+0xe10] # 0x55555555600
0x555555551f4	<main+77></main+77>	mov	rdi.rax
0x555555551f7	<main+80></main+80>	mov	eax,0x0
0x5555555551fc	<main+85></main+85>	call	0x555555555070 <printf@plt></printf@plt>
0x555555555201	<main+90></main+90>	MOV	eax.0x0
0x555555555206	<main+95></main+95>	mov	rdx, QWORD PTR [rbp-0x8]
0x55555555520a	<main+99></main+99>	sub	rdx.OWORD PTR fs:0x28
0x555555555213		ie	0x55555555521a <main+115></main+115>
0x555555555215		call	0x555555555060 < stack chk fail@plt>
0x55555555521a		leave	

inst pointer

stack pointer rax

base pointer rdi

endbr64

грр

rbp,rsp

rsp,0x10

eax,eax

rax, QWORD PTR fs:0x28

QWORD PTR [rbp-0x8], rax

return

1er arg

rsi

rdx

push

MOV

sub

MOV

MOV

XOL

0x55555555551a7 <main>

0x5555555551ab <main+4>

0x5555555551ac <main+5>

0x5555555551af <main+8>

0x55555555551b3 <main+12>

0x5555555555bc <main+21>

0x5555555551c0 <main+25>

rip

rsp

rbp

555...51c9

fff..0fe8

fff..0ff8



STACK

SIACK		
RBP+0		
RBP+0	fff1000	← RBP
RBP-8	Ptr fs:0x28	
RBP-16	0x0000	←RSP
RBP-24		
RBP-32		
RBP-40		
RBP-48		
RBP-56		
RBP-64		
RBP-72		
RBP-80		
RBP-88		
RBP-96		
RBP-104		
RBP-112		
RBP-120		

Direcciones bajas

RSP	

_				Sai
	0x555555551a7	<main></main>	endbr6	4
	0x555555551ab	<main+4></main+4>	push	rbp
	0x555555551ac	<main+5></main+5>	MOV	rbp,rsp
	0x555555551af	<main+8></main+8>	sub	rsp,0x10
	0x555555551b3	<main+12></main+12>	MOV	rax,QWORD PTR fs:0x28
	0x555555551bc	<main+21></main+21>	MOV	QWORD PTR [rbp-0x8],rax
	0x555555551c0	<main+25></main+25>	XOL	eax,eax
	0x555555551c2	<main+27></main+27>	MOV	DWORD PTR [rbp-0x10],0x0
	0x555555551c9	<main+34></main+34>	lea	rax,[rbp-0x10]
	0x555555551cd	<main+38></main+38>	MOV	rdx,rax -
	0x555555551d0	<main+41></main+41>	MOV	esi,0x8
	0x555555551d5	<main+46></main+46>	mov	edi,0x4
	0x555555551da	<main+51></main+51>	call	0x555555555169 <div></div>
	0x555555551df	<main+56></main+56>	MOV	DWORD PTR [rbp-0xc],eax
	0x555555551e2	<main+59></main+59>	cmp	DWORD PTR [rbp-0xc],0x0
	0x555555551e6	<main+63></main+63>	je	0x555555555201 <main+90></main+90>
	0x555555551e8	<main+65></main+65>	mov	eax,DWORD PTR [rbp-0x10]
	0x555555551eb	<main+68></main+68>	MOV	esi,eax
	0x555555551ed	<main+70></main+70>	lea	rax,[rip+0xe10] # 0x555555556004
	0x555555551f4	<main+77></main+77>	MOV	rdi,rax
	0x555555551f7	<main+80></main+80>	MOV	eax,0x0
	0x555555551fc	<main+85></main+85>	call	0x555555555070 <printf@plt></printf@plt>
	0x55555555201	<main+90></main+90>	MOV	eax,0x0
	0x55555555206	<main+95></main+95>	MOV	rdx,QWORD PTR [rbp-0x8]
	0x5555555520a	<main+99></main+99>	sub	rdx,QWORD PTR fs:0x28
	0x555555555213	<main+108></main+108>	je	0x55555555521a <main+115></main+115>
	0x555555555215	<main+110></main+110>	call	0x555555555060 <stack_chk_fail@plt></stack_chk_fail@plt>
	0x55555555521a	<main+115></main+115>	leave	
			-	

rip	55551cd	inst pointer					
rsp	fff0fe8	stack pointer	rax	0xff0fe8	return	rsi	2ndo arg
rbp	fff0ff8	base pointer	rdi		1er arg	rdx	3er arg



SIACK		
RBP+0		
RBP+0	fff1000	← RBP
RBP-8	Ptr fs:0x28	
RBP-16	0x0000	←RSP
RBP-24		
RBP-32		
RBP-40		
RBP-48		
RBP-56		
RBP-64		
RBP-72		
RBP-80		
RBP-88		
RBP-96		
RBP-104		
RBP-112		
RBP-120		

_				Jai
	0x555555551a7	<main></main>	endbr6	4
	0x555555551ab	<main+4></main+4>	push	rbp
	0x555555551ac	<main+5></main+5>	MOV	rbp,rsp
	0x555555551af	<main+8></main+8>	sub	rsp,0x10
	0x555555551b3	<main+12></main+12>	MOV	rax,QWORD PTR fs:0x28
	0x555555551bc	<main+21></main+21>	MOV	QWORD PTR [rbp-0x8],rax
	0x555555551c0	<main+25></main+25>	XOL	eax,eax
	0x555555551c2	<main+27></main+27>	MOV	DWORD PTR [rbp-0x10],0x0
	0x555555551c9	<main+34></main+34>	lea	rax,[rbp-0x10]
	0x555555551cd	<main+38></main+38>	MOV	rdx,rax
	0x555555551d0	<main+41></main+41>	MOV	esi,0x8 <
	0x555555551d5	<main+46></main+46>	mov	edi,0x4
	0x555555551da	<main+51></main+51>	call	0x555555555169 <div></div>
	0x555555551df	<main+56></main+56>	mov	DWORD PTR [rbp-0xc],eax
	0x555555551e2	<main+59></main+59>	CMP	DWORD PTR [rbp-0xc],0x0
	0x555555551e6	<main+63></main+63>	je	0x555555555201 <main+90></main+90>
	0x555555551e8	<main+65></main+65>	mov	eax, DWORD PTR [rbp-0x10]
	0x555555551eb	<main+68></main+68>	MOV	esi,eax
	0x555555551ed	<main+70></main+70>	lea	rax,[rip+0xe10] # 0x555555556004
	0x555555551f4	<main+77></main+77>	mov	rdi,rax
	0x5555555551f7	<main+80></main+80>	mov	eax,0x0
	0x555555551fc	<main+85></main+85>	call	0x555555555070 <printf@plt></printf@plt>
	0x555555555201	<main+90></main+90>	MOV	eax,0x0
	0x555555555206	<main+95></main+95>	MOV	rdx,QWORD PTR [rbp-0x8]
	0x55555555520a	<main+99></main+99>	sub	rdx, QWORD PTR fs:0x28
	0x555555555213	<main+108></main+108>	je	0x55555555521a <main+115></main+115>
	0x555555555215	<main+110></main+110>	call	0x555555555060 < stack chk fail@plt>
	0x55555555521a	<main+115></main+115>	leave	THE PROPERTY OF THE PARTY OF TH

rip	55551d0	inst pointer						
rsp	fff0fe8	stack pointer	rax	0xff0fe8	return	rsi		2ndo arg
rbp	fffOff8	base pointer	rdi		1er arg	rdx	0xff0fe8	3er arg



STACK

fff1000	← RBP
Ptr fs:0x28	
0x0000	← RS
	Ptr fs:0x28

Direcciones bajas

endbr64 0x5555555551a7 <main> 0x5555555551ab <main+4> push грр 0x5555555551ac <main+5> rbp, rsp MOV 0x5555555551af <main+8> sub rsp,0x10 0x55555555551b3 <main+12> rax.OWORD PTR fs:0x28 MOV 0x55555555551bc <main+21> QWORD PTR [rbp-0x8], rax MOV 0x5555555551c0 <main+25> eax, eax XOL DWORD PTR [rbp-0x10],0x0 0x55555555551c2 <main+27> MOV 0x55555555551c9 <main+34> lea rax,[rbp-0x10] 0x5555555551cd <main+38> rdx, rax MOV 0x5555555551d0 <main+41> esi.0x8 MOV 0x55555555551d5 <main+46> edi,0x4 MOV 0x5555555551da <main+51> call 0x555555555169 <div> DWORD PTR [rbp-0xc].eax 0x55555555551df <main+56> MOV 0x5555555551e2 <main+59> DWORD PTR [rbp-0xc],0x0 CMD 0x5555555551e6 <main+63> 0x5555555555201 <main+90> je 0x5555555551e8 <main+65> eax.DWORD PTR [rbp-0x10] MOV 0x5555555551eb <main+68> esi.eax MOV 0x5555555551ed <main+70> lea rax,[rip+0xe10] 0x5555555551f4 <main+77> rdi.rax MOV 0x5555555551f7 <main+80> eax,0x0 MOV 0x5555555551fc <main+85> call 0x5555555555070 <printf@plt> 0x5555555555201 <main+90> eax.0x0 MOV rdx, QWORD PTR [rbp-0x8] 0x5555555555206 <main+95> MOV rdx.OWORD PTR fs:0x28 0x555555555520a <main+99> sub 0x5555555555213 <main+108> je 0x555555555521a <main+115> 0x5555555555215 <main+110> call 0x5555555555060 < __stack_chk_fail@plt> 0x555555555521a <main+115> leave

rip	55551d5	inst pointer						
rsp	fff0fe8	stack pointer	rax	0xff0fe8	return	rsi	0x8	2ndo arg
rbp	fff0ff8	base pointer	rdi		1er arg	rdx	0xff0fe8	3er arg



STACK

SIACK		
RBP+0		
RBP+0	fff1000	← RBP
RBP-8	Ptr fs:0x28	
RBP-16	0x0000	← RS
RBP-24		
RBP-32		
RBP-40		
RBP-48		
RBP-56		
RBP-64		
RBP-72		
RBP-80		
RBP-88		
RBP-96		
RBP-104		
RBP-112		
RBP-120		

Direcciones bajas

----RSP

endbr64 0x5555555551a7 <main> 0x5555555551ab <main+4> push грр 0x5555555551ac <main+5> rbp, rsp MOV 0x5555555551af <main+8> sub rsp,0x10 rax.OWORD PTR fs:0x28 0x55555555551b3 <main+12> MOV 0x55555555551bc <main+21> QWORD PTR [rbp-0x8], rax MOV 0x5555555551c0 <main+25> eax, eax XOL DWORD PTR [rbp-0x10],0x0 0x55555555551c2 <main+27> MOV 0x55555555551c9 <main+34> lea rax,[rbp-0x10] 0x5555555551cd <main+38> rdx, rax MOV 0x5555555551d0 <main+41> esi.0x8 MOV 0x55555555551d5 <main+46> edi,0x4 MOV 0x5555555551da <main+51> call 0x555555555169 <div> DWORD PTR [rbp-0xcl.eax 0x5555555551df <main+56> MOV 0x5555555551e2 <main+59> DWORD PTR [rbp-0xc],0x0 CMD 0x5555555551e6 <main+63> 0x5555555555201 <main+90> je 0x5555555551e8 <main+65> eax.DWORD PTR [rbp-0x10] MOV 0x5555555551eb <main+68> esi.eax MOV 0x5555555551ed <main+70> lea rax,[rip+0xe10] 0x5555555551f4 <main+77> rdi.rax MOV 0x5555555551f7 <main+80> eax,0x0 MOV 0x5555555551fc <main+85> call 0x5555555555070 <printf@plt> 0x5555555555201 <main+90> eax.0x0 MOV rdx, QWORD PTR [rbp-0x8] 0x5555555555206 <main+95> MOV rdx.OWORD PTR fs:0x28 0x555555555520a <main+99> sub 0x5555555555213 <main+108> je 0x555555555521a <main+115> 0x5555555555215 <main+110> call 0x5555555555060 < __stack_chk_fail@plt> 0x555555555521a <main+115> leave

rip	55551da	inst pointer						
rsp	fff0fe8	stack pointer	rax	0xff0fe8	return	rsi	0x8	2ndo arg
rbp	fff0ff8	base pointer	rdi	0x4	1er arg	rdx	0xff0fe8	3er arg



STACK

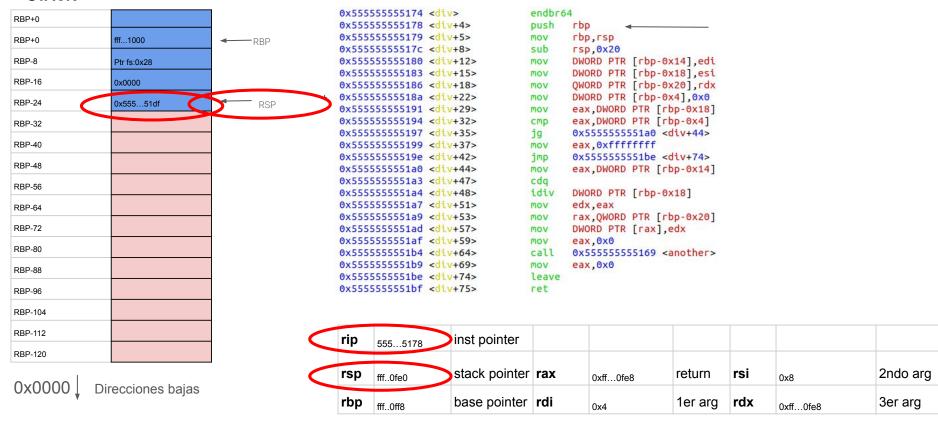
SIACK		
RBP+0		
RBP+0	fff1000	← RBP
RBP-8	Ptr fs:0x28	
RBP-16	0x0000	← RS
RBP-24		
RBP-32		
RBP-40		
RBP-48		
RBP-56		
RBP-64		
RBP-72		
RBP-80		
RBP-88		
RBP-96		
RBP-104		
RBP-112		
RBP-120		

Direcciones bajas

				Jaim
0x555555551a7	<main></main>	endbr6	4	
0x555555551ab	<main+4></main+4>	push	rbp	
0x555555551ac	<main+5></main+5>	MOV	rbp,rsp	
0x555555551af	<main+8></main+8>	sub	rsp,0x10	
0x555555551b3	<main+12></main+12>	MOV	rax, QWORD PTR fs:0x28	
0x555555551bc	<main+21></main+21>	MOV	QWORD PTR [rbp-0x8],rax	Push %RIP
0x555555551c0	<main+25></main+25>	XOL	eax,eax	
0x555555551c2	<main+27></main+27>	MOV	DWORD PTR [rbp-0x10],0x0	Jump a Div
0x555555551c9	<main+34></main+34>	lea	rax,[rbp-0x10]	
0x555555551cd	<main+38></main+38>	MOV	rdx,rax	
0x555555551d0	<main+41></main+41>	MOV	esi,0x8	
0x555555551d5	<main+46></main+46>	MOV	edi,0x4	
0x555555551da	<main+51></main+51>	call	0x555555555169 <div></div>	
0x555555551df	<main+56></main+56>	MOV	DWORD PTR [rbp-0xc],eax	
0x555555551e2	<main+59></main+59>	cmp	DWORD PTR [rbp-0xc],0x0	
0x555555551e6	<main+63></main+63>	je	0x5555555555201 <main+90></main+90>	
0x555555551e8	<main+65></main+65>	mov	eax, DWORD PTR [rbp-0x10]	
0x555555551eb	<main+68></main+68>	MOV	esi,eax	
0x555555551ed	<main+70></main+70>	lea	rax,[rip+0xe10] # 0x5	
0x555555551f4	<main+77></main+77>	MOV	rdi,rax	
0x5555555551f7	<main+80></main+80>	MOV	eax,0x0	
0x555555551fc	<main+85></main+85>	call	0x5555555555070 <printf@plt></printf@plt>	
0x555555555201	<main+90></main+90>	MOV	eax,0x0	
0x555555555206	<main+95></main+95>	MOV	rdx,QWORD PTR [rbp-0x8]	
0x5555555520a	<main+99></main+99>	sub	rdx,QWORD PTR fs:0x28	
0x555555555213	<main+108></main+108>	je	0x555555555521a <main+115></main+115>	
0x555555555215	<main+110></main+110>	call	0x555555555060 <stack_chk_< td=""><td>fail@plt></td></stack_chk_<>	fail@plt>
0x55555555521a	<main+115></main+115>	leave		

rip	55551da	inst pointer						
rsp	fff0fe8	stack pointer	rax	0xff0fe8	return	rsi	0x8	2ndo arg
rbp	fff0ff8	base pointer	rdi	0x4	1er arg	rdx	0xff0fe8	3er arg







STACK

SIACK		
RBP+0		
RBP+0	fff1000	← RBP
RBP-8	Ptr fs:0x28	
RBP-16	0x0000	
RBP-24	0x55551df	
RBP-32	fffOff8	← RSP
RBP-40		
RBP-48		
RBP-56		
RBP-64		
RBP-72		
RBP-80		
RBP-88		
RBP-96		
RBP-104		
RBP-112		
RBP-120		

```
endbr64
  0x555555555174 <div>
                                    push
                                           rbp
  0x5555555555178 <div+4>
  0x5555555555179 <div+5>
                                    MOV
                                           rbp,rsp
                                           rsp,0x20
  0x555555555517c <div+8>
                                    sub
                                           DWORD PTR [rbp-0x14],edi
  0x5555555555180 <div+12>
                                    MOV
                                           DWORD PTR [rbp-0x18],esi
  0x5555555555183 <div+15>
                                    MOV
                                           OWORD PTR [rbp-0x20],rdx
  0x5555555555186 <div+18>
                                    MOV
+ 0x55555555518a <div+22>
                                           DWORD PTR [rbp-0x4],0x0
                                    MOV
                                           eax, DWORD PTR [rbp-0x18]
  0x5555555555191 <div+29>
                                    MOV
                                           eax DWORD PTR [rbp-0x4]
  0x5555555555194 <div+32>
                                    CMP
                                           0x55555555551a0 <div+44>
  0x5555555555197 <div+35>
                                    jq
  0x5555555555199 <div+37>
                                           eax, 0xffffffff
                                    MOV
  0x555555555519e <div+42>
                                    jmp
                                           0x55555555551be <div+74>
  0x5555555551a0 <div+44>
                                           eax, DWORD PTR [rbp-0x14]
                                    MOV
  0x5555555551a3 <div+47>
                                    cda
                                    idiv
                                           DWORD PTR [rbp-0x18]
  0x55555555551a4 <div+48>
  0x55555555551a7 <div+51>
                                           edx, eax
                                    MOV
  0x5555555551a9 <div+53>
                                    MOV
                                           rax, OWORD PTR [rbp-0x20]
  0x5555555551ad <div+57>
                                           DWORD PTR [rax],edx
                                    MOV
  0x5555555551af <div+59>
                                           eax.0x0
                                    MOV
  0x5555555551b4 <div+64>
                                    call
                                           0x5555555555169 <another>
  0x5555555551b9 <div+69>
                                    MOV
                                           eax.0x0
  0x5555555551be <div+74>
                                    leave
  0x55555555551bf <div+75>
                                    ret
```

rip	5555179	inst pointer						
rsp	fff0fd8	stack pointer	rax	0xff0fe8	return	rsi	0x8	2ndo arg
rbp	fff0ff8	base pointer	rdi	0x4	1er arg	rdx	0xff0fe8	3er arg



STACK

0171011			
RBP+44			
RBP+40	fff1000		
RBP+32	Ptr fs:0x28		
RBP+24	0x0000		
RBP+16	0x55551df		
RBP+8	fff0ff8	←	RSP
RBP+0			
RBP-8			
RBP-16			
RBP-24			
RBP-32			
RBP-40			
RBP-48			
RBP-56			
RBP-64			
RBP-72			
RBP-80			

	0x555555555174	<div></div>	endbr6	4
	0x555555555178	<div+4></div+4>	push	rbp
	0x555555555179	<div+5></div+5>	MOV	rbp,rsp
	0x55555555517c	<div+8></div+8>	sub	rsp,0x20 ◀
	0x55555555180	<div+12></div+12>	MOV	DWORD PTR [rbp-0x14],edi
	0x55555555183	<div+15></div+15>	mov	DWORD PTR [rbp-0x18],esi
	0x55555555186	<div+18></div+18>	MOV	QWORD PTR [rbp-0x20],rdx
۲	0x5555555518a	<div+22></div+22>	MOV	DWORD PTR [rbp-0x4],0x0
	0x555555555191	<div+29></div+29>	MOV	eax, DWORD PTR [rbp-0x18]
BP	0x555555555194	<div+32></div+32>	стр	eax, DWORD PTR [rbp-0x4]
	0x555555555197	<div+35></div+35>	jg	0x5555555551a0 <div+44></div+44>
	0x555555555199	<div+37></div+37>	mov	eax,0xffffffff
	0x55555555519e	<div+42></div+42>	jmp	0x5555555551be <div+74></div+74>
	0x555555551a0	<div+44></div+44>	mov	eax, DWORD PTR [rbp-0x14]
	0x555555551a3	<div+47></div+47>	cdq	
	0x555555551a4	<div+48></div+48>	idiv	DWORD PTR [rbp-0x18]
	0x555555551a7	<div+51></div+51>	mov	edx eax
	0x555555551a9	<div+53></div+53>	mov	rax, OWORD PTR [rbp-0x20]
	0x555555551ad	<div+57></div+57>	mov	DWORD PTR [rax],edx
	0x555555551af	<div+59></div+59>	mov	eax,0x0
	0x555555551b4	<div+64></div+64>	call	0x5555555555169 <another></another>
	0x555555551b9	<div+69></div+69>	mov	eax,0x0
	0x555555551be	<div+74></div+74>	leave	NO DESCRIPTION OF PERSONS ASSESSMENT ASSESSMENT OF PERSONS ASSESSMENT OF PERSONS ASSESSMENT ASSESSMENT ASSESSM
	0x555555551bf	<div+75></div+75>	ret	

rip	555517c	inst pointer						
rsp	fff0fd8	stack pointer	rax	0xff0fe8	return	rsi	0x8	2ndo arg
rbp	fff0fd8	base pointer	rdi	0x4	1er arg	rdx	0xff0fe8	3er arg



SIACK		
RBP+44		
RBP+40	fff1000	
RBP+32	Ptr fs:0x28	
RBP+24	0x0000	
RBP+16	0x55551df	
RBP+8	fffOff8	RBP
RBP+0		
RBP-8		
RBP-16		
RBP-24		← RSP
RBP-32		Rop
RBP-40		
RBP-48		
RBP-56		
RBP-64		
RBP-72		
RBP-80		

	0x55555555174	<div></div>	endbr6	4
	0x55555555178	<div+4></div+4>	push	гьр
	0x555555555179	<div+5></div+5>	MOV	rbp,rsp
	0x5555555517c	<div+8></div+8>	sub	rsp,0x20
	0x55555555180	<div+12></div+12>	MOV	DWORD PTR [rbp-0x14],edi ←
	0x55555555183	<div+15></div+15>	MOV	DWORD PTR [rbp-0x18],esi
	0x55555555186	<div+18></div+18>	MOV	QWORD PTR [rbp-0x20],rdx
F	0x5555555518a	<div+22></div+22>	MOV	DWORD PTR [rbp-0x4],0x0
	0x555555555191	<div+29></div+29>	MOV	eax, DWORD PTR [rbp-0x18]
	0x555555555194	<div+32></div+32>	cmp	eax, DWORD PTR [rbp-0x4]
	0x555555555197	<div+35></div+35>	jg	0x555555551a0 <div+44></div+44>
	0x55555555199	<div+37></div+37>	MOV	eax,0xffffffff
	0x5555555519e	<div+42></div+42>	jmp	0x5555555551be <div+74></div+74>
	0x555555551a0	<div+44></div+44>	MOV	eax, DWORD PTR [rbp-0x14]
	0x555555551a3	<div+47></div+47>	cdq	
	0x555555551a4	<div+48></div+48>	idiv	DWORD PTR [rbp-0x18]
	0x555555551a7	<div+51></div+51>	MOV	edx,eax
	0x555555551a9	<div+53></div+53>	MOV	rax,QWORD PTR [rbp-0x20]
	0x555555551ad	<div+57></div+57>	MOV	DWORD PTR [rax],edx
	0x555555551af	<div+59></div+59>	MOV	eax,0x0
	0x555555551b4	<div+64></div+64>	call	0x555555555169 <another></another>
	0x555555551b9	<div+69></div+69>	mov	eax,0x0
	0x555555551be	<div+74></div+74>	leave	
	0x555555551bf	<div+75></div+75>	ret	

rip	5555180	inst pointer						
rsp	fff0fb8	stack pointer	rax	0xff0fe8	return	rsi	0x8	2ndo arg
rbp	fff0fd8	base pointer	rdi	0x4	1er arg	rdx	0xff0fe8	3er arg



SIACK			
RBP+44			
RBP+40	fff1000		
RBP+32	Ptr fs:0x28		
RBP+24	0x0000		
RBP+16	0x55551df		
RBP+8	fff0ff8	_	RBP
RBP+0			
RBP-8			
RBP-16	0x4		
RBP-24		_	RSP
RBP-32			KOP
RBP-40			
RBP-48			
RBP-56			
RBP-64			
RBP-72			
RBP-80			



0x555555555174 <div></div>	endbre	54	
0x5555555555178 <div+4></div+4>	push	гЬр	
0x5555555555179 <div+5></div+5>	MOV	rbp,rsp	
0x55555555517c <div+8></div+8>	sub	rsp,0x20	
0x555555555180 <div+12></div+12>	MOV	DWORD PTR [rbp-0x14],edi	
0x5555555555183 <div+15></div+15>	mov	DWORD PTR [rbp-0x18],esi ←	
0x5555555555186 <div+18></div+18>	mov	QWORD PTR [rbp-0x20],rdx	
0x55555555518a <div+22></div+22>	mov	DWORD PTR [rbp-0x4],0x0	
0x5555555555191 <div+29></div+29>	mov	eax, DWORD PTR [rbp-0x18]	
0x5555555555194 <div+32></div+32>	стр	eax, DWORD PTR [rbp-0x4]	
0x5555555555197 <div+35></div+35>	jg	0x5555555551a0 <div+44></div+44>	
0x555555555199 <div+37></div+37>	mov	eax,0xffffffff	
0x55555555519e <div+42></div+42>	jmp	0x5555555551be <div+74></div+74>	
0x5555555551a0 <div+44></div+44>	mov	eax, DWORD PTR [rbp-0x14]	
0x5555555551a3 <div+47></div+47>	cdq		
0x5555555551a4 <div+48></div+48>	idiv	DWORD PTR [rbp-0x18]	
0x5555555551a7 <div+51></div+51>	mov	edx,eax	
0x5555555551a9 <div+53></div+53>	mov	rax,QWORD PTR [rbp-0x20]	
0x5555555551ad <div+57></div+57>	MOV	DWORD PTR [rax],edx	
0x5555555551af <div+59></div+59>	mov	eax,0x0	
0x5555555551b4 <div+64></div+64>	call	0x5555555555169 <another></another>	
0x5555555551b9 <div+69></div+69>	mov	eax,0x0	
0x5555555551be <div+74></div+74>	leave	a the attoring	
0x55555555551bf <div+75></div+75>	ret		

rip	5555183	inst pointer						
rsp	fff0fb8	stack pointer	rax	0xff0fe8	return	rsi	0x8	2ndo arg
rbp	fff0fd8	base pointer	rdi	0x4	1er arg	rdx	0xff0fe8	3er arg



SIACK			
RBP+44			
RBP+40	fff1000		
RBP+32	Ptr fs:0x28		
RBP+24	0x0000		
RBP+16	0x55551df		
RBP+8	fff0ff8	_	RBP
RBP+0			
RBP-8			
RBP-16	0x4 0x8		
RBP-24		_	RSP
RBP-32			KOP
RBP-40			
RBP-48			
RBP-56			
RBP-64			
RBP-72			
RBP-80			

	0x55555555174	<div></div>	endbr6	64
	0x555555555178	<div+4></div+4>	push	rbp
	0x555555555179	<div+5></div+5>	MOV	rbp,rsp
	0x55555555517c	<div+8></div+8>	sub	rsp,0x20
	0x55555555180	<div+12></div+12>	mov	DWORD PTR [rbp-0x14],edi
	0x55555555183	<div+15></div+15>	mov	DWORD PTR [rbp-0x18],esi
	0x55555555186	<div+18></div+18>	MOV	QWORD PTR [rbp-0x20],rdx ◀
+	0x5555555518a	<div+22></div+22>	mov	DWORD PTR [rbp-0x4],0x0
	0x555555555191	<div+29></div+29>	MOV	eax, DWORD PTR [rbp-0x18]
	0x555555555194	<div+32></div+32>	cmp	eax, DWORD PTR [rbp-0x4]
	0x555555555197	<div+35></div+35>	jg	0x555555551a0 <div+44></div+44>
	0x55555555199	<div+37></div+37>	mov	eax,0xffffffff
	0x55555555519e	<div+42></div+42>	jmp	0x5555555551be <div+74></div+74>
	0x555555551a0	<div+44></div+44>	mov	eax, DWORD PTR [rbp-0x14]
	0x555555551a3	<div+47></div+47>	cdq	A STATE OF THE STA
	0x555555551a4	<div+48></div+48>	idiv	DWORD PTR [rbp-0x18]
	0x555555551a7	<div+51></div+51>	MOV	edx,eax
	0x555555551a9	<div+53></div+53>	mov	rax, OWORD PTR [rbp-0x20]
	0x555555551ad	<div+57></div+57>	MOV	DWORD PTR [rax],edx
	0x555555551af	<div+59></div+59>	mov	eax,0x0
	0x555555551b4	<div+64></div+64>	call	0x555555555169 <another></another>
	0x555555551b9	<div+69></div+69>	mov	eax,0x0
	0x555555551be	<div+74></div+74>	leave	Printed Table 19 1
	0x555555551bf	<div+75></div+75>	ret	
	0x555555551bf	<div+75></div+75>	ret	

rip	5555186	inst pointer						
rsp	fff0fb8	stack pointer	rax	0xff0fe8	return	rsi	0x8	2ndo arg
rbp	fff0fd8	base pointer	rdi	0x4	1er arg	rdx	0xff0fe8	3er arg



SIACK			
RBP+44			
RBP+40	fff1000		
RBP+32	Ptr fs:0x28		
RBP+24	0x0000		
RBP+16	0x55551df		
RBP+8	fff0ff8	_	RBP
RBP+0			
RBP-8			
RBP-16	0x4 0x8		
RBP-24	0xff0fe8		RSP
RBP-32			NOF
RBP-40			
RBP-48			
RBP-56			
RBP-64			
RBP-72			
RBP-80			

0x0000	Direcciones bajas
OVOCOC 1	Direcciones bajas

	0x55555555174	<div></div>	endbr6	4	
	0x55555555178	<div+4></div+4>	push	грр	
	0x555555555179	<div+5></div+5>	MOV	rbp,rsp	
	0x5555555517c	<div+8></div+8>	sub	rsp,0x20	
	0x55555555180	<div+12></div+12>	MOV	DWORD PTR [rbp-0x14],edi	
	0x55555555183	<div+15></div+15>	MOV	DWORD PTR [rbp-0x18],esi	
	0x55555555186	<div+18></div+18>	MOV	QWORD PTR [rbp-0x20],rdx	
F	0x5555555518a	<div+22></div+22>	MOV	DWORD PTR [rbp-0x4],0x0	←
	0x555555555191	<div+29></div+29>	MOV	eax, DWORD PTR [rbp-0x18]	
	0x555555555194	<div+32></div+32>	стр	eax, DWORD PTR [rbp-0x4]	
	0x555555555197	<div+35></div+35>	jg	0x5555555551a0 <div+44></div+44>	
	0x55555555199	<div+37></div+37>	MOV	eax,0xffffffff	
	0x55555555519e	<div+42></div+42>	jmp	0x5555555551be <div+74></div+74>	
	0x555555551a0	<div+44></div+44>	MOV	eax, DWORD PTR [rbp-0x14]	
	0x555555551a3	<div+47></div+47>	cdq		
	0x555555551a4	<div+48></div+48>	idiv	DWORD PTR [rbp-0x18]	
	0x555555551a7	<div+51></div+51>	MOV	edx,eax	
	0x555555551a9	<div+53></div+53>	mov	rax, OWORD PTR [rbp-0x20]	
	0x555555551ad	<div+57></div+57>	MOV	DWORD PTR [rax],edx	
	0x555555551af	<div+59></div+59>	MOV	eax,0x0	
	0x555555551b4	<div+64></div+64>	call	0x5555555555169 <another></another>	
	0x555555551b9	<div+69></div+69>	mov	eax,0x0	
	0x555555551be	<div+74></div+74>	leave	NO SECURITION OF	
	0x555555551bf	<div+75></div+75>	ret		

rip	555518a	inst pointer						
rsp	fff0fb8	stack pointer	rax	0xff0fe8	return	rsi	0x8	2ndo arg
rbp	fff0fd8	base pointer	rdi	0x4	1er arg	rdx	0xff0fe8	3er arg



SIACK			
RBP+44			
RBP+40	fff1000		
RBP+32	Ptr fs:0x28		
RBP+24	0x0000		
RBP+16	0x55551df		
RBP+8	fff0ff8	_	RBP
RBP+0	0x0		
RBP-8			
RBP-16	0x4 0x8		
RBP-24	0xff0fe8	_	RSP
RBP-32			KOP
RBP-40			
RBP-48			
RBP-56			
RBP-64			
RBP-72			
RBP-80			

1	
0x0000	Direcciones bajas

0x555555555174 <div></div>	endbro	54
0x555555555178 <div+4></div+4>	push	гЬр
0x5555555555179 <div+5></div+5>	MOV	rbp,rsp
0x55555555517c <div+8></div+8>	sub	rsp,0x20
0x555555555180 <div+12></div+12>	MOV	DWORD PTR [rbp-0x14],edi
0x5555555555183 <div+15></div+15>	MOV	DWORD PTR [rbp-0x18],esi
0x555555555186 <div+18></div+18>	MOV	QWORD PTR [rbp-0x20],rdx
+ 0x55555555518a <div+22></div+22>	MOV	DWORD PTR [rbp-0x4],0x0
0x555555555191 <div+29></div+29>	MOV	eax,DWORD PTR [rbp-0x18] ←
0x5555555555194 <div+32></div+32>	cmp	eax, DWORD PTR [rbp-0x4]
0x5555555555197 <div+35></div+35>	jg	0x555555551a0 <div+44></div+44>
0x5555555555199 <div+37></div+37>	MOV	eax,0xffffffff
0x55555555519e <div+42></div+42>	jmp	0x555555551be <div+74></div+74>
0x5555555551a0 <div+44></div+44>	MOV	eax, DWORD PTR [rbp-0x14]
0x5555555551a3 <div+47></div+47>	cdq	
0x5555555551a4 <div+48></div+48>	idiv	DWORD PTR [rbp-0x18]
0x5555555551a7 <div+51></div+51>	MOV	edx,eax
0x5555555551a9 <div+53></div+53>	MOV	rax,QWORD PTR [rbp-0x20]
0x5555555551ad <div+57></div+57>	MOV	DWORD PTR [rax],edx
0x5555555551af <div+59></div+59>	MOV	eax,0x0
0x5555555551b4 <div+64></div+64>	call	0x555555555169 <another></another>
0x5555555551b9 <div+69></div+69>	MOV	eax,0x0
0x5555555551be <div+74></div+74>	leave	
0x5555555551bf <div+75></div+75>	ret	

rip	5555191	inst pointer						
rsp	fff0fb8	stack pointer	rax	0xff0fe8	return	rsi	0x8	2ndo arg
rbp	fff0fd8	base pointer	rdi	0x4	1er arg	rdx	0xff0fe8	3er arg



STACK

OIAGIC			
RBP+44			
RBP+40	fff1000		
RBP+32	Ptr fs:0x28		
RBP+24	0x0000		
RBP+16	0x55551df		
RBP+8	fff0ff8	_	RBP
RBP+0	0x0		
RBP-8			
RBP-16	0x4 0x8		
RBP-24	0xff0fe8	_	RSP
RBP-32			ROP
RBP-40			
RBP-48			
RBP-56			
RBP-64			
RBP-72			
RBP-80			

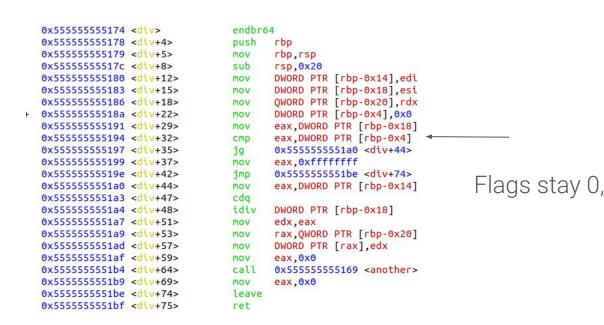
	0x55555555174	<div></div>	endbr6	4
	0x555555555178	<div+4></div+4>	push	гьр
	0x555555555179	<div+5></div+5>	MOV	rbp,rsp
	0x5555555517c	<div+8></div+8>	sub	rsp,0x20
	0x55555555180	<div+12></div+12>	MOV	DWORD PTR [rbp-0x14],edi
	0x55555555183	<div+15></div+15>	mov	DWORD PTR [rbp-0x18],esi
	0x55555555186	<div+18></div+18>	MOV	QWORD PTR [rbp-0x20],rdx
F	0x5555555518a	<div+22></div+22>	MOV	DWORD PTR [rbp-0x4],0x0
	0x555555555191	<div+29></div+29>	MOV	eax, DWORD PTR [rbp-0x18]
	0x555555555194	<div+32></div+32>	cmp	eax, DWORD PTR [rbp-0x4]
	0x555555555197	<div+35></div+35>	jg	0x555555551a0 <div+44></div+44>
	0x55555555199	<div+37></div+37>	mov	eax,0xffffffff
	0x55555555519e	<div+42></div+42>	jmp	0x555555551be <div+74></div+74>
	0x555555551a0	<div+44></div+44>	mov	eax, DWORD PTR [rbp-0x14]
	0x555555551a3	<div+47></div+47>	cdq	10 10 10 10 10 10 10 10 10 10 10 10 10 1
	0x555555551a4	<div+48></div+48>	idiv	DWORD PTR [rbp-0x18]
	0x555555551a7	<div+51></div+51>	MOV	edx,eax
	0x555555551a9	<div+53></div+53>	mov	rax,QWORD PTR [rbp-0x20]
	0x555555551ad	<div+57></div+57>	MOV	DWORD PTR [rax],edx
	0x555555551af	<div+59></div+59>	mov	eax,0x0
	0x555555551b4	<div+64></div+64>	call	0x555555555169 <another></another>
	0x555555551b9	<div+69></div+69>	mov	eax,0x0
	0x555555551be	<div+74></div+74>	leave	CONTRACTOR OF THE CONTRACTOR O
	0x555555551bf	<div+75></div+75>	ret	

rip	5555194	inst pointer						
rsp	fff0fb8	stack pointer	rax	0x8	return	rsi	0x8	2ndo arg
rbp	fff0fd8	base pointer	rdi	0x4	1er arg	rdx	0xff0fe8	3er arg



RBP+44		
RBP+40	fff1000	
RBP+32	Ptr fs:0x28	
RBP+24	0x0000	
RBP+16	0x55551df	
RBP+8	fff0ff8	 RBP
RBP+0	0x0	
RBP-8		
RBP-16	0x4 0x8	
RBP-24	0xff0fe8	 DOD
RBP-32		RSP
RBP-40		
RBP-48		
RBP-56		
RBP-64		
RBP-72		
RBP-80		





rip	5555194	inst pointer						
rsp	fff0fb8	stack pointer	rax	0x8	return	rsi	0x8	2ndo arg
rbp	fff0fd8	base pointer	rdi	0x4	1er arg	rdx	0xff0fe8	3er arg



STACK

OIAGIC			
RBP+44			
RBP+40	fff1000		
RBP+32	Ptr fs:0x28		
RBP+24	0x0000		
RBP+16	0x55551df		
RBP+8	fff0ff8	_	RBP
RBP+0	0x0		
RBP-8			
RBP-16	0x4 0x8		
RBP-24	0xff0fe8	_	RSP
RBP-32			ROP
RBP-40			
RBP-48			
RBP-56			
RBP-64			
RBP-72			
RBP-80			

	0x555555555174	<div></div>	endbr6	4	
	0x555555555178	<div+4></div+4>	push	гЬр	
	0x555555555179	<div+5></div+5>	MOV	rbp,rsp	
	0x5555555517c	<div+8></div+8>	sub	rsp,0x20	
	0x55555555180	<div+12></div+12>	MOV	DWORD PTR [rbp-0x14],edi	
	0x55555555183	<div+15></div+15>	mov	DWORD PTR [rbp-0x18],esi	
	0x55555555186	<div+18></div+18>	MOV	QWORD PTR [rbp-0x20],rdx	
F	0x5555555518a	<div+22></div+22>	MOV	DWORD PTR [rbp-0x4],0x0	
	0x555555555191	<div+29></div+29>	MOV	eax, DWORD PTR [rbp-0x18]	
	0x555555555194	<div+32></div+32>	cmp	eax, DWORD PTR [rbp-0x4]	
	0x555555555197	<div+35></div+35>	jg	0x5555555551a0 <div+44></div+44>	←
	0x55555555199	<div+37></div+37>	mov	eax,0xffffffff	
	0x55555555519e	<div+42></div+42>	jmp	0x5555555551be <div+74></div+74>	
	0x555555551a0	<div+44></div+44>	mov	eax, DWORD PTR [rbp-0x14]	
	0x555555551a3	<div+47></div+47>	cdq		
	0x555555551a4	<div+48></div+48>	idiv	DWORD PTR [rbp-0x18]	
	0x555555551a7	<div+51></div+51>	MOV	edx,eax	
	0x555555551a9	<div+53></div+53>	mov	rax, QWORD PTR [rbp-0x20]	
	0x555555551ad	<div+57></div+57>	MOV	DWORD PTR [rax],edx	
	0x555555551af	<div+59></div+59>	mov	eax,0x0	
	0x555555551b4	<div+64></div+64>	call	0x5555555555169 <another></another>	
	0x555555551b9	<div+69></div+69>	mov	eax,0x0	
	0x555555551be	<div+74></div+74>	leave		
	0x555555551bf	<div+75></div+75>	ret		

rip	5555197	inst pointer						
rsp	fff0fb8	stack pointer	rax	0x8	return	rsi	0x8	2ndo arg
rbp	fff0fd8	base pointer	rdi	0x4	1er arg	rdx	0xff0fe8	3er arg



SIACK			
RBP+44			
RBP+40	fff1000		
RBP+32	Ptr fs:0x28		
RBP+24	0x0000		
RBP+16	0x55551df		
RBP+8	fff0ff8	_	RBP
RBP+0	0x0		
RBP-8			
RBP-16	0x4 0x8		
RBP-24	0xff0fe8	_	RSP
RBP-32			KOP
RBP-40			
RBP-48			
RBP-56			
RBP-64			
RBP-72			
RBP-80			

0x0000	Direcciones	bajas
OVOCCO 1	Directiones	Dajas

	0x555555555174	<div></div>	endbr6	4	
	0x55555555178	<div+4></div+4>	push	rbp	
	0x55555555179	<div+5></div+5>	MOV	rbp,rsp	
	0x5555555517c	<div+8></div+8>	sub	rsp,0x20	
	0x55555555180	<div+12></div+12>	MOV	DWORD PTR [rbp-0x14],edi	
	0x55555555183	<div+15></div+15>	MOV	DWORD PTR [rbp-0x18],esi	
	0x55555555186	<div+18></div+18>	MOV	QWORD PTR [rbp-0x20],rdx	
۲	0x5555555518a	<div+22></div+22>	MOV	DWORD PTR [rbp-0x4],0x0	
	0x555555555191	<div+29></div+29>	MOV	eax, DWORD PTR [rbp-0x18]	
	0x55555555194	<div+32></div+32>	стр	eax, DWORD PTR [rbp-0x4]	
	0x555555555197	<div+35></div+35>	jg	0x5555555551a0 <div+44></div+44>	
	0x55555555199	<div+37></div+37>	MOV	eax,0xffffffff	
	0x5555555519e	<div+42></div+42>	jmp	0x5555555551be <div+74></div+74>	
	0x555555551a0	<div+44></div+44>	MOV	eax, DWORD PTR [rbp-0x14]	←
	0x555555551a3	<div+47></div+47>	cdq		
	0x555555551a4	<div+48></div+48>	idiv	DWORD PTR [rbp-0x18]	
	0x555555551a7	<div+51></div+51>	MOV	edx,eax	
	0x555555551a9	<div+53></div+53>	MOV	rax,QWORD PTR [rbp-0x20]	
	0x555555551ad	<div+57></div+57>	MOV	DWORD PTR [rax],edx	
	0x555555551af	<div+59></div+59>	mov	eax,0x0	
	0x555555551b4	<div+64></div+64>	call	0x5555555555169 <another></another>	
	0x555555551b9	<div+69></div+69>	mov	eax,0x0	
	0x555555551be	<div+74></div+74>	leave		
	0x555555551bf	<div+75></div+75>	ret		

rip	55551a0	inst pointer						
rsp	fff0fb8	stack pointer	rax	0x8	return	rsi	0x8	2ndo arg
rbp	fff0fd8	base pointer	rdi	0x4	1er arg	rdx	0xff0fe8	3er arg



STACK

OIAGIC			
RBP+44			
RBP+40	fff1000		
RBP+32	Ptr fs:0x28		
RBP+24	0x0000		
RBP+16	0x55551df		
RBP+8	fffOff8	_	RBP
RBP+0	0x0		
RBP-8			
RBP-16	0x4 0x8		
RBP-24	0xff0fe8	_	RSP
RBP-32			ROP
RBP-40			
RBP-48			
RBP-56			
RBP-64			
RBP-72			
RBP-80			

```
endbr64
  0x555555555174 <div>
  0x5555555555178 <div+4>
                                    push
                                           грр
  0x5555555555179 <div+5>
                                           rbp,rsp
                                    MOV
                                           rsp,0x20
  0x555555555517c <div+8>
                                    sub
                                           DWORD PTR [rbp-0x14],edi
  0x5555555555180 <div+12>
                                    MOV
                                           DWORD PTR [rbp-0x18],esi
  0x5555555555183 <div+15>
                                    MOV
                                           OWORD PTR [rbp-0x20],rdx
  0x5555555555186 <div+18>
                                    MOV
+ 0x55555555518a <div+22>
                                           DWORD PTR [rbp-0x4],0x0
                                    MOV
                                           eax, DWORD PTR [rbp-0x18]
  0x5555555555191 <div+29>
                                    MOV
                                           eax DWORD PTR [rbp-0x4]
  0x5555555555194 <div+32>
                                    CMP
  0x5555555555197 <div+35>
                                           0x5555555551a0 <div+44>
                                    jq
  0x5555555555199 <div+37>
                                           eax, 0xffffffff
                                    MOV
                                           0x55555555551be <div+74>
  0x555555555519e <div+42>
                                    jmp
  0x5555555551a0 <div+44>
                                           eax, DWORD PTR [rbp-0x14]
                                    MOV
  0x5555555551a3 <div+47>
                                    cda
                                    idiv
                                           DWORD PTR [rbp-0x18]
  0x55555555551a4 <div+48>
  0x55555555551a7 <div+51>
                                           edx, eax
                                    MOV
  0x5555555551a9 <div+53>
                                    MOV
                                           rax, OWORD PTR [rbp-0x20]
  0x5555555551ad <div+57>
                                           DWORD PTR [rax],edx
                                    MOV
  0x5555555551af <div+59>
                                           eax,0x0
                                    MOV
  0x5555555551b4 <div+64>
                                    call
                                           0x5555555555169 <another>
  0x5555555551b9 <div+69>
                                           eax.0x0
                                    MOV
  0x5555555551be <div+74>
                                    leave
  0x55555555551bf <div+75>
                                    ret
```

rip	55551a3	inst pointer						
rsp	fff0fb8	stack pointer	rax	0x4	return	rsi	0x8	2ndo arg
rbp	fff0fd8	base pointer	rdi	0x4	1er arg	rdx	0xff0fe8	3er arg



STACK

SIACK			
RBP+44			
RBP+40	fff1000		
RBP+32	Ptr fs:0x28		
RBP+24	0x0000		
RBP+16	0x55551df		
RBP+8	fff0ff8	_	RBP
RBP+0	0x0	,	
RBP-8			
RBP-16	0x4 0x8		
RBP-24	0xff0fe8	_	RSP
RBP-32			NOF
RBP-40			
RBP-48			
RBP-56			
RBP-64			
RBP-72			
RBP-80			



```
0x555555555174 <div>
                                    endbr64
                                    push
  0x5555555555178 <div+4>
                                           грр
  0x5555555555179 <div+5>
                                           rbp,rsp
                                    MOV
  0x555555555517c <div+8>
                                    sub
                                           rsp.0x20
                                           DWORD PTR [rbp-0x14],edi
  0x5555555555180 <div+12>
                                           DWORD PTR [rbp-0x18],esi
  0x5555555555183 <div+15>
                                    MOV
                                           OWORD PTR [rbp-0x20],rdx
  0x5555555555186 <div+18>
+ 0x55555555518a <div+22>
                                           DWORD PTR [rbp-0x4],0x0
                                    MOV
                                           eax, DWORD PTR [rbp-0x18]
  0x5555555555191 <div+29>
                                    MOV
                                           eax DWORD PTR [rbp-0x4]
  0x5555555555194 <div+32>
                                           0x55555555551a0 <div+44>
  0x5555555555197 <div+35>
                                    jq
  0x5555555555199 <div+37>
                                           eax, 0xffffffff
                                    MOV
  0x555555555519e <div+42>
                                    jmp
                                           0x55555555551be <div+74>
  0x5555555551a0 <div+44>
                                           eax, DWORD PTR [rbp-0x14]
                                    MOV
  0x5555555551a3 <div+47>
                                    cda
                                    idiv
                                           DWORD PTR [rbp-0x18]
  0x5555555551a4 <div+48>
  0x55555555551a7 <div+51>
                                           edx, eax
                                    MOV
  0x5555555551a9 <div+53>
                                           rax, OWORD PTR [rbp-0x20]
  0x5555555551ad <div+57>
                                           DWORD PTR [rax],edx
                                    MOV
  0x5555555551af <div+59>
                                           eax.0x0
                                    MOV
  0x5555555551b4 <div+64>
                                    call
                                           0x5555555555169 <another>
  0x5555555551b9 <div+69>
                                    MOV
                                           eax.0x0
  0x5555555551be <div+74>
                                    leave
  0x55555555551bf <div+75>
                                    ret
```

https://www.felixcloutier.com/x86/idiv

rip	55551a4	inst pointer						
rsp	fff0fb8	stack pointer	rax	0x4	return	rsi	0x8	2ndo arg
rbp	fff0fd8	base pointer	rdi	0x4	1er arg	rdx	0	3er arg



STACK

OIAGIC			
RBP+44			
RBP+40	fff1000		
RBP+32	Ptr fs:0x28		
RBP+24	0x0000		
RBP+16	0x55551df		
RBP+8	fffOff8	_	RBP
RBP+0	0x0		
RBP-8			
RBP-16	0x4 0x8		
RBP-24	0xff0fe8	_	RSP
RBP-32			ROP
RBP-40			
RBP-48			
RBP-56			
RBP-64			
RBP-72			
RBP-80			

```
endbr64
  0x555555555174 <div>
                                    push
  0x5555555555178 <div+4>
                                           грр
  0x5555555555179 <div+5>
                                           rbp,rsp
                                    MOV
                                           rsp,0x20
  0x555555555517c <div+8>
                                    sub
                                           DWORD PTR [rbp-0x14],edi
  0x5555555555180 <div+12>
                                    MOV
                                           DWORD PTR [rbp-0x18],esi
  0x5555555555183 <div+15>
                                    MOV
                                           OWORD PTR [rbp-0x20],rdx
  0x5555555555186 <div+18>
                                    MOV
+ 0x55555555518a <div+22>
                                           DWORD PTR [rbp-0x4],0x0
                                    MOV
                                           eax, DWORD PTR [rbp-0x18]
  0x5555555555191 <div+29>
                                    MOV
                                           eax DWORD PTR [rbp-0x4]
  0x5555555555194 <div+32>
                                    CMP
  0x5555555555197 <div+35>
                                           0x5555555551a0 <div+44>
                                    jq
  0x5555555555199 <div+37>
                                           eax, 0xffffffff
                                    MOV
                                           0x55555555551be <div+74>
  0x555555555519e <div+42>
                                    jmp
  0x5555555551a0 <div+44>
                                           eax, DWORD PTR [rbp-0x14]
                                    MOV
  0x5555555551a3 <div+47>
                                    cda
                                    idiv
                                           DWORD PTR [rbp-0x18]
  0x55555555551a4 <div+48>
  0x55555555551a7 <div+51>
                                           edx, eax
                                    MOV
  0x5555555551a9 <div+53>
                                    MOV
                                           rax, OWORD PTR [rbp-0x20]
  0x5555555551ad <div+57>
                                           DWORD PTR [rax],edx
                                    MOV
  0x5555555551af <div+59>
                                           eax,0x0
                                    MOV
  0x5555555551b4 <div+64>
                                    call
                                           0x5555555555169 <another>
  0x5555555551b9 <div+69>
                                           eax.0x0
                                    MOV
  0x55555555551be <div+74>
                                    leave
  0x55555555551bf <div+75>
                                    ret
```

rip	55551a7	inst pointer						
rsp	fff0fb8	stack pointer	rax	0x0	return	rsi	0x8	2ndo arg
rbp	fff0fd8	base pointer	rdi	0x4	1er arg	rdx	0	3er arg



SIACK			
RBP+44			
RBP+40	fff1000		
RBP+32	Ptr fs:0x28		
RBP+24	0x0000		
RBP+16	0x55551df		
RBP+8	fff0ff8	_	RBP
RBP+0	0x0		
RBP-8			
RBP-16	0x4 0x8		
RBP-24	0xff0fe8	_	RSP
RBP-32			KOP
RBP-40			
RBP-48			
RBP-56			
RBP-64			
RBP-72			
RBP-80			

1		
0x0000	Direcciones	bajas

0x5555555555174 <div></div>	endbr6	4	
0x5555555555178 <div+4></div+4>	push	грр	
0x5555555555179 <div+5></div+5>	MOV	rbp,rsp	
0x555555555517c <div+8></div+8>	sub	rsp,0x20	
0x555555555180 <div+12></div+12>	MOV	DWORD PTR [rbp-0x14],edi	
0x5555555555183 <div+15></div+15>	mov	DWORD PTR [rbp-0x18],esi	
0x555555555186 <div+18></div+18>	MOV	QWORD PTR [rbp-0x20],rdx	
+ 0x55555555518a <div+22></div+22>	MOV	DWORD PTR [rbp-0x4],0x0	
0x5555555555191 <div+29></div+29>	MOV	eax, DWORD PTR [rbp-0x18]	
0x5555555555194 <div+32></div+32>	cmp	eax, DWORD PTR [rbp-0x4]	
0x5555555555197 <div+35></div+35>	jg	0x5555555551a0 <div+44></div+44>	
0x5555555555199 <div+37></div+37>	mov	eax,0xffffffff	
0x555555555519e <div+42></div+42>	jmp	0x5555555551be <div+74></div+74>	
0x5555555551a0 <div+44></div+44>	mov	eax, DWORD PTR [rbp-0x14]	
0x5555555551a3 <div+47></div+47>	cdq		
0x5555555551a4 <div+48></div+48>	idiv	DWORD PTR [rbp-0x18]	
0x5555555551a7 <div+51></div+51>	MOV	edx,eax	
0x5555555551a9 <div+53></div+53>	MOV	<pre>rax,QWORD PTR [rbp-0x20]</pre>	←
0x5555555551ad <div+57></div+57>	MOV	DWORD PTR [rax],edx	
0x5555555551af <div+59></div+59>	MOV	eax,0x0	
0x5555555551b4 <div+64></div+64>	call	0x5555555555169 <another></another>	
0x5555555551b9 <div+69></div+69>	mov	eax,0x0	
0x5555555551be <div+74></div+74>	leave		
0x5555555551bf <div+75></div+75>	ret		

rip	55551a9	inst pointer						
rsp	fff0fb8	stack pointer	rax	0x0	return	rsi	0x8	2ndo arg
rbp	fff0fd8	base pointer	rdi	0x4	1er arg	rdx	0x0	3er arg



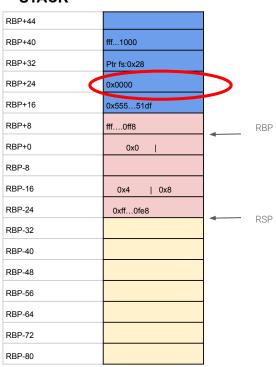
STACK

OIAGIC			
RBP+44			
RBP+40	fff1000		
RBP+32	Ptr fs:0x28		
RBP+24	0x0000		
RBP+16	0x55551df		
RBP+8	fffOff8	_	RBP
RBP+0	0x0		
RBP-8			
RBP-16	0x4 0x8		
RBP-24	0xff0fe8	_	RSP
RBP-32			ROP
RBP-40			
RBP-48			
RBP-56			
RBP-64			
RBP-72			
RBP-80			

```
endbr64
  0x555555555174 <div>
  0x5555555555178 <div+4>
                                    push
                                           грр
  0x5555555555179 <div+5>
                                           rbp,rsp
                                    MOV
                                           rsp,0x20
  0x555555555517c <div+8>
                                    sub
                                           DWORD PTR [rbp-0x14],edi
  0x5555555555180 <div+12>
                                    MOV
                                           DWORD PTR [rbp-0x18],esi
  0x5555555555183 <div+15>
                                    MOV
                                           OWORD PTR [rbp-0x20],rdx
  0x5555555555186 <div+18>
                                    MOV
+ 0x55555555518a <div+22>
                                           DWORD PTR [rbp-0x4],0x0
                                    MOV
  0x5555555555191 <div+29>
                                           eax, DWORD PTR [rbp-0x18]
                                    MOV
                                           eax DWORD PTR [rbp-0x4]
  0x5555555555194 <div+32>
                                    CMP
  0x5555555555197 <div+35>
                                           0x55555555551a0 <div+44>
                                    jq
  0x5555555555199 <div+37>
                                           eax, 0xffffffff
                                    MOV
  0x555555555519e <div+42>
                                    jmp
                                           0x55555555551be <div+74>
                                           eax, DWORD PTR [rbp-0x14]
  0x5555555551a0 <div+44>
                                    MOV
  0x5555555551a3 <div+47>
                                    cda
                                    idiv
                                           DWORD PTR [rbp-0x18]
  0x55555555551a4 <div+48>
  0x55555555551a7 <div+51>
                                           edx, eax
                                    MOV
  0x5555555551a9 <div+53>
                                    MOV
                                           rax, OWORD PTR [rbp-0x20]
  0x5555555551ad <div+57>
                                           DWORD PTR [rax],edx
                                    MOV
  0x5555555551af <div+59>
                                           eax,0x0
                                    MOV
  0x5555555551b4 <div+64>
                                    call
                                           0x5555555555169 <another>
  0x5555555551b9 <div+69>
                                           eax.0x0
                                    MOV
  0x55555555551be <div+74>
                                    leave
  0x55555555551bf <div+75>
                                    ret
```

rip	55551ad	inst pointer						
rsp	fff0fb8	stack pointer	rax	0xff0fe8	return	rsi	0x8	2ndo arg



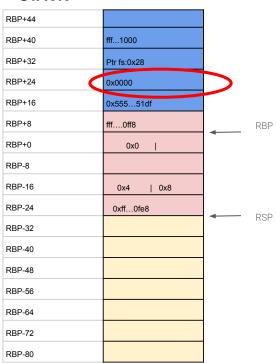


0x0000	Direcciones bajas
--------	-------------------

0x55555555174	<div></div>	endbr6	4
0x55555555178	<div+4></div+4>	push	гЬр
0x55555555179	<div+5></div+5>	MOV	rbp,rsp
0x5555555517c	<div+8></div+8>	sub	rsp,0x20
0x55555555180	<div+12></div+12>	MOV	DWORD PTR [rbp-0x14],edi
0x55555555183	<div+15></div+15>	MOV	DWORD PTR [rbp-0x18],esi
0x55555555186	<div+18></div+18>	MOV	QWORD PTR [rbp-0x20],rdx
0x5555555518a	<div+22></div+22>	MOV	DWORD PTR [rbp-0x4],0x0
0x555555555191	<div+29></div+29>	MOV	eax, DWORD PTR [rbp-0x18]
0x55555555194	<div+32></div+32>	cmp	eax, DWORD PTR [rbp-0x4]
0x555555555197	<div+35></div+35>	jg	0x5555555551a0 <div+44></div+44>
0x55555555199	<div+37></div+37>	mov	eax,0xffffffff
0x5555555519e	<div+42></div+42>	jmp	0x5555555551be <div+74></div+74>
0x555555551a0	<div+44></div+44>	mov	eax, DWORD PTR [rbp-0x14]
0x555555551a3	<div+47></div+47>	cdq	
0x555555551a4	<div+48></div+48>	idiv	DWORD PTR [rbp-0x18]
0x555555551a7	<div+51></div+51>	MOV	edx,eax
0x555555551a9	<div+53></div+53>	MOV	rax, QWORD PTR [rbp-0x20]
0x555555551ad	<div+57></div+57>	MOV	DWORD PTR [rax],edx
0x555555551af	<div+59></div+59>	MOV	eax,0x0 ◀
0x555555551b4	<div+64></div+64>	call	0x5555555555169 <another></another>
0x555555551b9	<div+69></div+69>	mov	eax,0x0
0x555555551be	<div+74></div+74>	leave	
0x555555551bf	<div+75></div+75>	ret	

rip	55551af	inst pointer						
rsp	fff0fb8	stack pointer	rax	0xff0fe8	return	rsi	0x8	2ndo arg
rbp	fff0fd8	base pointer	rdi	0x4	1er arg	rdx	0x0	3er arg







	0x555555555174 <div></div>	endbré	54	
	0x555555555178 <div+4></div+4>	push	rbp	
	0x5555555555179 <div+5></div+5>	MOV	rbp,rsp	
	0x55555555517c <div+8></div+8>	sub	rsp,0x20	
	0x555555555180 <div+12></div+12>	MOV	DWORD PTR [rbp-0x14],edi	
	0x555555555183 <div+15></div+15>	mov	DWORD PTR [rbp-0x18],esi	
	0x555555555186 <div+18></div+18>	MOV	QWORD PTR [rbp-0x20],rdx	
F	0x55555555518a <div+22></div+22>	MOV	DWORD PTR [rbp-0x4],0x0	
	0x5555555555191 <div+29></div+29>	MOV	eax, DWORD PTR [rbp-0x18]	
	0x5555555555194 <div+32></div+32>	cmp	eax, DWORD PTR [rbp-0x4]	
	0x5555555555197 <div+35></div+35>	jg	0x5555555551a0 <div+44></div+44>	
	0x555555555199 <div+37></div+37>	mov	eax,0xffffffff	
	0x55555555519e <div+42></div+42>	jmp	0x5555555551be <div+74></div+74>	
	0x5555555551a0 <div+44></div+44>	mov	eax, DWORD PTR [rbp-0x14]	
	0x5555555551a3 <div+47></div+47>	cdq		
	0x5555555551a4 <div+48></div+48>	idiv	DWORD PTR [rbp-0x18]	
	0x5555555551a7 <div+51></div+51>	MOV	edx,eax	Considerar no op
	0x5555555551a9 <div+53></div+53>	mov	rax, QWORD PTR [rbp-0x20]	(sin operacion)
	0x5555555551ad <div+57></div+57>	MOV	DWORD PTR [rax],edx	(SIII operación)
	0x5555555551af <div+59></div+59>	MOV	eax,0x0	
	0x5555555551b4 <div+64></div+64>	call	0x5555555555169 <another> <</another>	
	0x5555555551b9 <div+69></div+69>	mov	eax,0x0	
	0x5555555551be <div+74></div+74>	leave	With a Tangent of	
	0x5555555551bf <div+75></div+75>	ret		

rip	55551b4	inst pointer						
rsp	fff0fb8	stack pointer	rax	0	return	rsi	0x8	2ndo arg
rbp	fff0fd8	base pointer	rdi	0x4	1er arg	rdx	0x0	3er arg



SIACK			
RBP+44			
RBP+40	fff1000		
RBP+32	Ptr fs:0x28		
RBP+24	0x0000		
RBP+16	0x55551df		
RBP+8	fff0ff8	_	RBP
RBP+0	0x0		
RBP-8			
RBP-16	0x4 0x8		
RBP-24	0xff0fe8	_	RSP
RBP-32			KOP
RBP-40			
RBP-48			
RBP-56			
RBP-64			
RBP-72			
RBP-80			

1		
0x0000 \	Direcciones	bajas

	0x555555555174	<div></div>	endbr6	14
	0x555555555178	<div+4></div+4>	push	rbp
	0x555555555179	<div+5></div+5>	mov	rbp,rsp
	0x5555555517c	<div+8></div+8>	sub	rsp,0x20
	0x55555555180	<div+12></div+12>	MOV	DWORD PTR [rbp-0x14],edi
	0x55555555183	<div+15></div+15>	mov	DWORD PTR [rbp-0x18],esi
	0x55555555186	<div+18></div+18>	mov	QWORD PTR [rbp-0x20],rdx
۲	0x5555555518a	<div+22></div+22>	mov	DWORD PTR [rbp-0x4],0x0
	0x555555555191	<div+29></div+29>	MOV	eax, DWORD PTR [rbp-0x18]
	0x555555555194	<div+32></div+32>	cmp	eax, DWORD PTR [rbp-0x4]
	0x555555555197	<div+35></div+35>	jg	0x5555555551a0 <div+44></div+44>
	0x55555555199	<div+37></div+37>	mov	eax,0xffffffff
	0x55555555519e	<div+42></div+42>	jmp	0x5555555551be <div+74></div+74>
	0x555555551a0	<div+44></div+44>	mov	eax, DWORD PTR [rbp-0x14]
	0x555555551a3	<div+47></div+47>	cdq	
	0x555555551a4	<div+48></div+48>	idiv	DWORD PTR [rbp-0x18]
	0x555555551a7	<div+51></div+51>	MOV	edx,eax
	0x555555551a9	<div+53></div+53>	MOV	rax, QWORD PTR [rbp-0x20]
	0x555555551ad	<div+57></div+57>	mov	DWORD PTR [rax],edx
	0x555555551af	<div+59></div+59>	MOV	eax,0x0
	0x555555551b4	<div+64></div+64>	call	0x5555555555169 <another></another>
	0x555555551b9	<div+69></div+69>	mov	eax,0x0 ←
	0x555555551be	<div+74></div+74>	leave	
	0x555555551bf	<div+75></div+75>	ret	

rip	55551b9	inst pointer						
rsp	fff0fb8	stack pointer	rax	0	return	rsi	0x8	2ndo arg
rbp	fff0fd8	base pointer	rdi	0x4	1er arg	rdx	0x0	3er arg



STACK

SIACK			
RBP+44			
RBP+40	fff1000		
RBP+32	Ptr fs:0x28		
RBP+24	0x0000		
RBP+16	0x55551df		
RBP+8	fff0ff8	_	RBP
RBP+0	0x0		
RBP-8			
RBP-16	0x4 0x8		
RBP-24	0xff0fe8	_	RSP
RBP-32			KOP
RBP-40			
RBP-48			
RBP-56			
RBP-64			
RBP-72			
RBP-80			

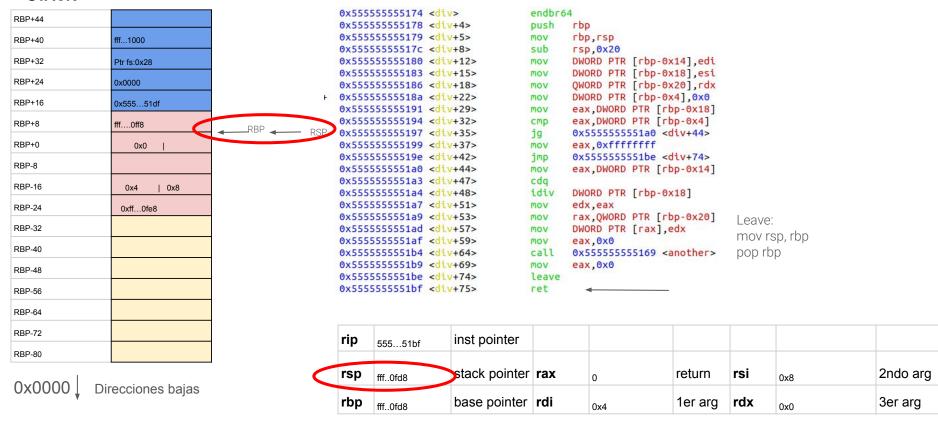


	0x555555555174	<div></div>	endbr6	4
	0x555555555178	<div+4></div+4>	push	rbp
	0x555555555179	<div+5></div+5>	mov	rbp,rsp
	0x5555555517c	<div+8></div+8>	sub	rsp,0x20
	0x55555555180	<div+12></div+12>	MOV	DWORD PTR [rbp-0x14],edi
	0x55555555183	<div+15></div+15>	mov	DWORD PTR [rbp-0x18],esi
	0x55555555186	<div+18></div+18>	mov	QWORD PTR [rbp-0x20],rdx
F	0x5555555518a	<div+22></div+22>	mov	DWORD PTR [rbp-0x4],0x0
	0x55555555191	<div+29></div+29>	MOV	eax, DWORD PTR [rbp-0x18]
	0x55555555194	<div+32></div+32>	cmp	eax, DWORD PTR [rbp-0x4]
	0x555555555197	<div+35></div+35>	jg	0x5555555551a0 <div+44></div+44>
	0x55555555199	<div+37></div+37>	MOV	eax,0xffffffff
	0x55555555519e	<div+42></div+42>	jmp	0x5555555551be <div+74></div+74>
	0x555555551a0	<div+44></div+44>	mov	eax, DWORD PTR [rbp-0x14]
	0x555555551a3	<div+47></div+47>	cdq	A POLY THE STORY OF THE STORY O
	0x555555551a4	<div+48></div+48>	idiv	DWORD PTR [rbp-0x18]
	0x555555551a7	<div+51></div+51>	MOV	edx,eax
	0x555555551a9	<div+53></div+53>	MOV	rax, QWORD PTR [rbp-0x20]
	0x555555551ad	<div+57></div+57>	MOV	DWORD PTR [rax],edx
	0x555555551af	<div+59></div+59>	MOV	eax,0x0
	0x555555551b4	<div+64></div+64>	call	0x5555555555169 <another></another>
	0x555555551b9	<div+69></div+69>	mov	eax,0x0
	0x555555551be	<div+74></div+74>	leave	Proceedings to the process of the pr
	0x555555551bf	<div+75></div+75>	ret	

Leave: mov rsp, rbp pop rbp

rip	55551be	inst pointer						
rsp	fff0fb8	stack pointer	rax	0	return	rsi	0x8	2ndo arg
rbp	fff0fd8	base pointer	rdi	0x4	1er arg	rdx	0x0	3er arg







STACK		
RBP+44		
RBP+0	fff1000	RBP
RBP-8	Ptr fs:0x28	
RBP-16	0x0000	
RBP-24	0x55551df	
RBP-32	fff0ff8	RSP
RBP-40	0x0	
RBP-48		
RBP-56	0x4 0x8	
RBP-64	0xff0fe8	
RBP-72		
RBP-80		
RBP-88		
RBP-96		
RBP-104		
RBP-112		
RBP-120		

0x0000	Direcciones	bajas
0,10000	D11000101100	~ ajac

	0x55555555174	add	and bac		
			endbr6		
	0x55555555178		push	грр	
	0x55555555179	<div+5></div+5>	MOV	rbp,rsp	
	0x5555555517c	<div+8></div+8>	sub	rsp,0x20	
	0x55555555180	<div+12></div+12>	MOV	DWORD PTR [rbp-0x14],edi	
	0x55555555183	<div+15></div+15>	mov	DWORD PTR [rbp-0x18],esi	
	0x55555555186	<div+18></div+18>	mov	OWORD PTR [rbp-0x20],rdx	
F	0x5555555518a	<div+22></div+22>	mov	DWORD PTR [rbp-0x4],0x0	
	0x555555555191	<div+29></div+29>	mov	eax DWORD PTR [rbp-0x18]	
	0x555555555194	<div+32></div+32>	CMD	eax DWORD PTR [rbp-0x4]	
	0x555555555197	<div+35></div+35>	jg	0x5555555551a0 <div+44></div+44>	
	0x555555555199		mov	eax,0xffffffff	
	0x55555555519e		jmp	0x5555555551be <div+74></div+74>	
	0x5555555551a0	100	mov	eax, DWORD PTR [rbp-0x14]	
	0x5555555551a3		cdq	conjument in [rep exti]	
	0x5555555551a4		idiv	DWORD PTR [rbp-0x18]	
	0x5555555551a7		MOV	edx.eax	
	0x5555555551a9		mov	rax, OWORD PTR [rbp-0x20]	
					Leave:
	0x555555551ad		MOV	DWORD PTR [rax],edx	mov rsp, rbp
	0x555555551af		MOV	eax,0x0	. ' '
	0x555555551b4		call	0x5555555555169 <another></another>	pop rbp
	0x555555551b9		mov	eax,0x0	
	0x555555551be	<div+74></div+74>	leave		
	0x555555551bf	<div+75></div+75>	ret	←	

rip	55551bf	inst pointer						
rsp	fff0fd8	stack pointer	rax	0	return	rsi	0x8	2ndo arg
rbp	fff0ff8	base pointer	rdi	0x4	1er arg	rdx	0x0	3er arg



SIACK		
RBP+44		
RBP+0	fff1000	RBP
RBP-8	Ptr fs:0x28	
RBP-16	0x0000	
RBP-24	0x55551df	
RBP-32	fffOff8	RSF
RBP-40	0x0	
RBP-48		
RBP-56	0x4 0x8	
RBP-64	0xff0fe8	
RBP-72		
RBP-80		
RBP-88		
RBP-96		
RBP-104		
RBP-112		
RBP-120		

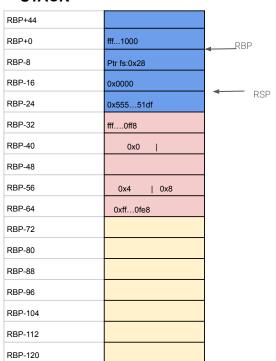
0x0000 J	, Direcciones	baias
	, Direcciones	Dajas

	0x55555555174	<div></div>	endbr6	14	
	0x555555555178	<div+4></div+4>	push	rbp	
	0x555555555179	<div+5></div+5>	MOV	rbp,rsp	
	0x5555555517c	<div+8></div+8>	sub	rsp,0x20	
	0x55555555180	<div+12></div+12>	MOV	DWORD PTR [rbp-0x14],edi	
	0x55555555183	<div+15></div+15>	MOV	DWORD PTR [rbp-0x18],esi	
	0x55555555186	<div+18></div+18>	MOV	QWORD PTR [rbp-0x20],rdx	
F	0x5555555518a	<div+22></div+22>	MOV	DWORD PTR [rbp-0x4],0x0	
	0x555555555191	<div+29></div+29>	MOV	eax, DWORD PTR [rbp-0x18]	
	0x55555555194	<div+32></div+32>	стр	eax, DWORD PTR [rbp-0x4]	
	0x555555555197	<div+35></div+35>	jg	0x5555555551a0 <div+44></div+44>	
	0x55555555199	<div+37></div+37>	MOV	eax,0xffffffff	
	0x5555555519e	<div+42></div+42>	jmp	0x5555555551be <div+74></div+74>	
	0x555555551a0	<div+44></div+44>	MOV	eax, DWORD PTR [rbp-0x14]	
	0x555555551a3	<div+47></div+47>	cdq		
	0x555555551a4	<div+48></div+48>	idiv	DWORD PTR [rbp-0x18]	
	0x555555551a7	<div+51></div+51>	MOV	edx,eax	
	0x555555551a9	<div+53></div+53>	MOV	rax,QWORD PTR [rbp-0x20]	ret:
	0x555555551ad	<div+57></div+57>	MOV	DWORD PTR [rax],edx	
	0x555555551af	<div+59></div+59>	MOV	eax,0x0	Pop %rip
	0x555555551b4	<div+64></div+64>	call	0x5555555555169 <another></another>	
	0x555555551b9	<div+69></div+69>	MOV	eax,0x0	
	0x555555551be	<div+74></div+74>	leave		
	0x555555551bf	<div+75></div+75>	ret	←	

rip	55551bf	inst pointer						
rsp	fff0fd8	stack pointer	rax	0	return	rsi	0x8	2ndo arg
rbp	fff0ff8	base pointer	rdi	0x4	1er arg	rdx	0x0	3er arg



STACK



0x0000 Direcciones bajas

		Ou!
0x5555555551a7 <main></main>	endbr	64
0x5555555551ab <main+< th=""><th>+4> push</th><th>гbр</th></main+<>	+4> push	гbр
0x5555555551ac <main+< th=""><th>+5> mov</th><th>rbp,rsp</th></main+<>	+5> mov	rbp,rsp
0x5555555551af <main+< th=""><th>+8> sub</th><th>rsp,0x10</th></main+<>	+8> sub	rsp,0x10
0x5555555551b3 <main+< th=""><th>+12> mov</th><th>rax,QWORD PTR fs:0x28</th></main+<>	+12> mov	rax,QWORD PTR fs:0x28
0x5555555551bc <main+< td=""><td>+21> mov</td><td>QWORD PTR [rbp-0x8],rax</td></main+<>	+21> mov	QWORD PTR [rbp-0x8],rax
0x5555555551c0 <main+< td=""><td>+25> xor</td><td>eax,eax</td></main+<>	+25> xor	eax,eax
0x5555555551c2 <main+< td=""><td>+27> mov</td><td>DWORD PTR [rbp-0x10],0x0</td></main+<>	+27> mov	DWORD PTR [rbp-0x10],0x0
0x5555555551c9 <main+< td=""><td>+34> lea</td><td>rax,[rbp-0x10]</td></main+<>	+34> lea	rax,[rbp-0x10]
0x5555555551cd <main+< td=""><td>+38> mov</td><td>rdx,rax</td></main+<>	+38> mov	rdx,rax
0x5555555551d0 <main+< td=""><td>+41> mov</td><td>esi,0x8</td></main+<>	+41> mov	esi,0x8
0x5555555551d5 <main+< td=""><td>+46> mov</td><td>edi,0x4</td></main+<>	+46> mov	edi,0x4
0x5555555551da <main+< td=""><td>+51> call</td><td>0x555555555169 <div></div></td></main+<>	+51> call	0x555555555169 <div></div>
0x5555555551df <main+< td=""><td>+56> mov</td><td>DWORD PTR [rbp-0xc],eax ←</td></main+<>	+56> mov	DWORD PTR [rbp-0xc],eax ←
0x5555555551e2 <main+< td=""><td>+59> cmp</td><td>DWORD PTR [rbp-0xc],0x0</td></main+<>	+59> cmp	DWORD PTR [rbp-0xc],0x0
0x5555555551e6 <main+< td=""><td>+63> je</td><td>0x555555555201 <main+90></main+90></td></main+<>	+63> je	0x555555555201 <main+90></main+90>
0x5555555551e8 <main+< td=""><td>+65> mov</td><td>eax,DWORD PTR [rbp-0x10]</td></main+<>	+65> mov	eax,DWORD PTR [rbp-0x10]
0x5555555551eb <main+< td=""><td>+68> mov</td><td>esi,eax</td></main+<>	+68> mov	esi,eax
0x5555555551ed <main+< td=""><td>+70> lea</td><td>rax,[rip+0xe10] # 0x55555556004</td></main+<>	+70> lea	rax,[rip+0xe10] # 0x55555556004
0x5555555551f4 <main+< td=""><td>+77> mov</td><td>rdi,rax</td></main+<>	+77> mov	rdi,rax
0x5555555551f7 <main+< td=""><td>+80> mov</td><td>eax,0x0</td></main+<>	+80> mov	eax,0x0
0x5555555551fc <main+< td=""><td>+85> call</td><td>0x555555555070 <printf@plt></printf@plt></td></main+<>	+85> call	0x555555555070 <printf@plt></printf@plt>
0x555555555201 <main+< td=""><td>+90> mov</td><td>eax,0x0</td></main+<>	+90> mov	eax,0x0
0x555555555206 <main+< td=""><td>+95> mov</td><td>rdx,QWORD PTR [rbp-0x8]</td></main+<>	+95> mov	rdx,QWORD PTR [rbp-0x8]
0x55555555520a <main+< td=""><td>+99> sub</td><td>rdx,QWORD PTR fs:0x28</td></main+<>	+99> sub	rdx,QWORD PTR fs:0x28
0x5555555555213 <main+< td=""><td>108> je</td><td>0x55555555521a <main+115></main+115></td></main+<>	108> je	0x55555555521a <main+115></main+115>
0x5555555555215 <main+< td=""><td>-110> call</td><td>0x555555555060 <stack_chk_fail@plt></stack_chk_fail@plt></td></main+<>	-110> call	0x555555555060 <stack_chk_fail@plt></stack_chk_fail@plt>
0x555555555521a <main+< td=""><td>115> leave</td><td></td></main+<>	115> leave	

rip	55551df	inst pointer						
rsp	fff0fd8	stack pointer	rax	0	return	rsi	0x8	2ndo arg
rbp	fff Off8	base pointer	rdi	0x4	1er arg	rdx	0x0	3er arg

Temas del Stack



- Stack Attack
- Dirección de retorno? (Return Address)
- Mas de 6 parametros?

Extra Slides



https://www.cs.virginia.edu/~evans/cs216/guides/x86.html