

Próxima instrucción: 13

```
1  #include <stdio.h>
2  int f(char a, int b, char c, long d, char e, short f, int g, int h) {
3      printf("a": %p\n", &a);
4      printf("b": %p\n", &b);
5      printf("c": %p\n", &c);
6      printf("d": %p\n", &d);
7      printf("e": %p\n", &e);
8      printf("f": %p\n", &f);
9      printf("g": %p\n", &g);
10     printf("h": %p\n", &h);
11     return 0;
12 }
13 int main() {
14     return f('1',2,'3',4,'5',6,7,8);
15 }
```

Registros

- RDI: argc
- RSI: argv
- RDX: ?
- RCX: ?
- R8: ?
- R9: ?
- RAX: ?

Pila

| | |
|-----------|--------|
| | |
| R.A. main | ← %rsp |
| | |
| | |
| | |
| | |
| | |
| | |

- Los argumentos a, c y e van en los registros RDI, RDX, R8; y ocupan 8 bits.
- Los argumentos b, d y f van en los registros RSI, RCX, R9; y ocupan 32 bits, 64 bits, 16 bits.
- Los argumentos g y h van en la pila y ocupan 64 bits en memoria.
- Todo esto dependerá de la arquitectura, compilador y convención de llamada.
- Como los printf van a imprimir direcciones de memoria pero esos valores se encuentran en registros, el compilador cargará dichos valores en la pila para poder imprimir esa dirección.

Próxima instrucción: 14

```
1  #include <stdio.h>
2  int f(char a, int b, char c, long d, char e, short f, int g, int h) {
3      printf("a": %p\n", &a);
4      printf("b": %p\n", &b);
5      printf("c": %p\n", &c);
6      printf("d": %p\n", &d);
7      printf("e": %p\n", &e);
8      printf("f": %p\n", &f);
9      printf("g": %p\n", &g);
10     printf("h": %p\n", &h);
11     return 0;
12 }
13 int main() {
14     return f('1',2,'3',4,'5',6,7,8);
15 }
```

Registros

- RDI: argc
- RSI: argv
- RDX: ?
- RCX: ?
- R8: ?
- R9: ?
- RAX: ?

Pila

| | |
|-----------|---------------|
| | ← %rsp = %rbp |
| R.A. main | |
| Ex %rbp | |
| | |
| | |
| | |

- Los argumentos a, c y e van en los registros RDI, RDX, R8; y ocupan 8 bits.
- Los argumentos b, d y f van en los registros RSI, RCX, R9; y ocupan 32 bits, 64 bits, 16 bits.
- Los argumentos g y h van en la pila y ocupan 64 bits en memoria.
- Todo esto dependerá de la arquitectura, compilador y convención de llamada.
- Como los printf van a imprimir direcciones de memoria pero esos valores se encuentran en registros, el compilador cargará dichos valores en la pila para poder imprimir esa dirección.

Próxima instrucción: 2

```

1  #include <stdio.h>
2  int f(char a, int b, char c, long d, char e, short f, int g, int h) {
3      printf("a": %p\n", &a);
4      printf("b": %p\n", &b);
5      printf("c": %p\n", &c);
6      printf("d": %p\n", &d);
7      printf("e": %p\n", &e);
8      printf("f": %p\n", &f);
9      printf("g": %p\n", &g);
10     printf("h": %p\n", &h);
11     return 0;
12 }
13 int main() {
14     return f('1',2,'3',4,'5',6,7,8);
15 }

```

Registros

- RDI: 49
- RSI: 2
- RDX: 51
- RCX: 4
- R8: 53
- R9: 6
- RAX: ?

Pila

| | |
|-----------|--------|
| | |
| R.A. main | ← %rbp |
| Ex %rbp | |
| 8 | |
| 7 | |
| R.A. | ← %rsp |
| | |
| | |

- Los argumentos a, c y e van en los registros RDI, RDX, R8; y ocupan 8 bits.
- Los argumentos b, d y f van en los registros RSI, RCX, R9; y ocupan 32 bits, 64 bits, 16 bits.
- Los argumentos g y h van en la pila y ocupan 64 bits en memoria.
- Todo esto dependerá de la arquitectura, compilador y convención de llamada.
- Como los printf van a imprimir direcciones de memoria pero esos valores se encuentran en registros, el compilador cargará dichos valores en la pila para poder imprimir esa dirección.

Próxima instrucción: 3

```

1  #include <stdio.h>
2  int f(char a, int b, char c, long d, char e, short f, int g, int h) {
3      printf("a": %p\n", &a);
4      printf("b": %p\n", &b);
5      printf("c": %p\n", &c);
6      printf("d": %p\n", &d);
7      printf("e": %p\n", &e);
8      printf("f": %p\n", &f);
9      printf("g": %p\n", &g);
10     printf("h": %p\n", &h);
11     return 0;
12 }
13 int main() {
14     return f('1',2,'3',4,'5',6,7,8);
15 }

```

Registros

- RDI: 49
- RSI: 2
- RDX: 51
- RCX: 4
- R8: 53
- R9: 6
- RAX: ?

Pila

| | |
|-----------|---------------|
| | |
| R.A. main | |
| Ex %rbp | |
| 8 | |
| 7 | |
| R.A. | |
| Ex %rbp | ← %rsp = %rbp |
| | |

- Los argumentos a, c y e van en los registros RDI, RDX, R8; y ocupan 8 bits.
- Los argumentos b, d y f van en los registros RSI, RCX, R9; y ocupan 32 bits, 64 bits, 16 bits.
- Los argumentos g y h van en la pila y ocupan 64 bits en memoria.
- Todo esto dependerá de la arquitectura, compilador y convención de llamada.
- Como los printf van a imprimir direcciones de memoria pero esos valores se encuentran en registros, el compilador cargará dichos valores en la pila para poder imprimir esa dirección.

Próxima instrucción: 11

```

1  #include <stdio.h>
2  int f(char a, int b, char c, long d, char e, short f, int g, int h) {
3      printf("a": %p\n", &a);
4      printf("b": %p\n", &b);
5      printf("c": %p\n", &c);
6      printf("d": %p\n", &d);
7      printf("e": %p\n", &e);
8      printf("f": %p\n", &f);
9      printf("g": %p\n", &g);
10     printf("h": %p\n", &h);
11     return 0;
12 }
13 int main() {
14     return f('1',2,'3',4,'5',6,7,8);
15 }

```

Registros

- RDI: ?
- RSI: ?
- RDX: ?
- RCX: ?
- R8: ?
- R9: ?
- RAX: 18

Pila

| | |
|-----------|---------------|
| | |
| R.A. main | |
| Ex %rbp | |
| 8 | |
| 7 | |
| R.A. | |
| Ex %rbp | ← %rsp = %rbp |
| | |

- Los argumentos a, c y e van en los registros RDI, RDX, R8; y ocupan 8 bits.
- Los argumentos b, d y f van en los registros RSI, RCX, R9; y ocupan 32 bits, 64 bits, 16 bits.
- Los argumentos g y h van en la pila y ocupan 64 bits en memoria.
- Todo esto dependerá de la arquitectura, compilador y convención de llamada.
- Como los printf van a imprimir direcciones de memoria pero esos valores se encuentran en registros, el compilador cargará dichos valores en la pila para poder imprimir esa dirección.

Próxima instrucción: 12

```

1  #include <stdio.h>
2  int f(char a, int b, char c, long d, char e, short f, int g, int h) {
3      printf("a": %p\n", &a);
4      printf("b": %p\n", &b);
5      printf("c": %p\n", &c);
6      printf("d": %p\n", &d);
7      printf("e": %p\n", &e);
8      printf("f": %p\n", &f);
9      printf("g": %p\n", &g);
10     printf("h": %p\n", &h);
11     return 0;
12 }
13 int main() {
14     return f('1',2,'3',4,'5',6,7,8);
15 }

```

Registros

- RDI: ?
- RSI: ?
- RDX: ?
- RCX: ?
- R8: ?
- R9: ?
- RAX: 0

Pila

| | |
|-----------|---------------|
| | |
| R.A. main | |
| Ex %rbp | |
| 8 | |
| 7 | |
| R.A. | |
| Ex %rbp | ← %rsp = %rbp |
| | |

- Los argumentos a, c y e van en los registros RDI, RDX, R8; y ocupan 8 bits.
- Los argumentos b, d y f van en los registros RSI, RCX, R9; y ocupan 32 bits, 64 bits, 16 bits.
- Los argumentos g y h van en la pila y ocupan 64 bits en memoria.
- Todo esto dependerá de la arquitectura, compilador y convención de llamada.
- Como los printf van a imprimir direcciones de memoria pero esos valores se encuentran en registros, el compilador cargará dichos valores en la pila para poder imprimir esa dirección.