

Ejecución: Next y Step

- s | step | stepi [count]
ejecuta [count] instrucciones
- n | next | p
Ejecuta instrucciones sin entrar a las subrutinas
- c | cont | continue
Continúa la ejecución
- q | quit | exit
Sale del debugger y del emulador
- Ctrl-C
Detiene la ejecución y retorna al prompt

Registros de propósito general

- r | reg | regs | registers

Lista los registros del CPU y sus contenidos

```
<bochs:12> registers
```

```
eax: 0x00000000 0
```

```
ecx: 0x00000000 0
```

```
edx: 0x00000543 1347
```

```
ebx: 0x00000000 0
```

```
esp: 0x00000000 0
```

```
ebp: 0x00000000 0
```

```
esi: 0x00000000 0
```

```
edi: 0x00000000 0
```

```
eip: 0x0000e05d
```

```
eflags 0x00000046
```

```
id vip vif ac vm rf nt IOPL=0 of df if tf sf ZF af PF cf
```

Memory Dump

- x /nuf [addr]

Muestra el contenido de la dirección [addr]

- xp /nuf [addr]

Muestra el contenido de la dirección física [addr]; nuf es número que indica cuantos valores se mostrarán, seguido de uno o más de los indicadores de formato.

x : hex	d : decimal	u : sin signo
o : octal	t : binario	c : char
s : ascii	i : instrucción	

y de tamaño

b : byte	h : word = half-word	w : doubleword = word
----------	----------------------	-----------------------

Memory Disassemble

- u | disasm | disassemble [count] [start] [end]
Desensambla instrucciones desde la dirección lineal [start] hasta [end] exclusive.
- u | disasm | disassemble switch-mode
Selecciona la sintaxis Intel o AT&T de assembler.
- u | disasm | disassemble size = n
Setea el tamaño del segmento a desensamblar.

Breakpoints

- p | pb | break | pbreak [addr] Crea un breakpoint en la dirección física [addr]
- vb | vbreak [seg:offset] Crea un breakpoint en la dirección virtual [addr]
- lb | lbreak [addr] Crea un breakpoint en la dirección lineal [addr]
- d | del | delete [n] Borra el breakpoint número [n]
- bpe [n] Activa el breakpoint número [n]
- bpd [n] Desactiva el breakpoint número [n]

Watches

- watch Muestra el estado actual de los watches
- watch stop Detiene la simulación cuando un watch es encontrado
- watch continue No detiene la simulación si un watch es encontrado
- watch r | read [addr] Agrega un watch de lectura en la dirección física [addr]
- watch w | write [addr] Agrega un watch de escritura en la dirección física [addr]

Infos

- info break Muestra los Breakpoint creados
- info eflags Muestra el registro EEFLAGS
- info idt Muestra el descriptor de interrupciones (idt)
- info ivt Muestra la tabla de vectores de interrupción
- info gdt Muestra la tabla global de descriptores (gdt)
- info tss Muestra el segmento de estado de tarea actual (tss)
- info tab Muestra la tabla de paginas

Registros de Segmento

- sreg Muestra los registros de segmento

```
<bochs:5> sreg
cs:s=0xf000, dh=0xff0093ff, dl=0x0000ffff, valid=7
ds:s=0x0000, dh=0x00009300, dl=0x0000ffff, valid=7
ss:s=0x0000, dh=0x00009300, dl=0x0000ffff, valid=7
es:s=0x0000, dh=0x00009300, dl=0x0000ffff, valid=7
fs:s=0x0000, dh=0x00009300, dl=0x0000ffff, valid=7
gs:s=0x0000, dh=0x00009300, dl=0x0000ffff, valid=7
ldtr:s=0x0000, dh=0x00008200, dl=0x0000ffff, valid=1
tr:s=0x0000, dh=0x00008b00, dl=0x0000ffff, valid=1
gdtr:base=0x00000000, limit=0xffff
idtr:base=0x00000000, limit=0xffff
```


Registros de Control

- creg Muestra los registros de control

```
<bochs:10> creg
```

```
CR0=0x60000010: pg CD NW ac wp ne ET ts em mp pe
```

```
CR2=page fault laddr=0x00000000
```

```
CR3=0x00000000
```

```
PCD=page-level cache disable=0
```

```
PWT=page-level writes transparent=0
```

```
CR4=0x00000000: osxsave smx vmx osxmmexcpt osfxsr pce pge mce pae pse de tsd pvi vme
```

Magic Breakpoint

- `xchg bx, bx` Magic breakpoint

Esta instrucción detiene el flujo del programa y nos devuelve al prompt de bochs.

Magic Breakpoint

- `xchg bx, bx` Magic breakpoint

Esta instrucción detiene el flujo del programa y nos devuelve al prompt de bochs.

