

# Comparación de códigos ensamblador: TASM 16-bit vs GCC x86-64 Linux

Característica	TASM (DOS, 16-bit)	GCC x86-64 Linux (Intel)
<b>Modelo de memoria</b>	.MODEL small (16-bit)	No aplica (64-bit)
<b>Segmentos</b>	.DATA, .STACK, .CODE	.data (.rodata), .text
<b>Variable para el número</b>	num DW 17	-8[rbp] = 17
<b>Variable divisor</b>	i DW 2	-16[rbp] = 2
<b>Bucle principal</b>	check_loop:	.L4 / .L6
<b>Comprobación de primos</b>	div cx → cmp dx, 0	idiv QWORD PTR -16[rbp] → test rdx, rdx
<b>Condición de salida</b>	cmp bx, ax / jge es_primo	cmp rax, -8[rbp] / jl .L6
<b>Incremento divisor</b>	inc bx	add -16[rbp], 1
<b>Impresión de resultado</b>	DOS interrupt 21h ( mov ah,09h; int 21h )	call puts@PLT con puntero a .rodata
<b>Finalización del programa</b>	mov ah,4Ch; int 21h	leave; ret
<b>Registros principales usados</b>	AX, BX, CX, DX	RAX, RDX, RBP, RDI
<b>Manejo de pila</b>	.STACK 100h	push rbp / sub rsp, 16
<b>Longitud de instrucciones</b>	Muy corta, cada instrucción realiza operaciones simples	Más larga, incluye setup de stack frame y manejo de registros de 64-bit
<b>Compatibilidad</b>	MS-DOS / TASM	Linux x86-64 / GCC
<b>Sintaxis</b>	Intel	Intel (x86-64)

## Observaciones clave:

1. TASM es un ensamblador 16-bit orientado a DOS, por lo que depende de interrupciones de DOS para I/O y usa registros de 16-bit.
2. GCC genera un ensamblador x86-64 moderno para Linux, con manejo de stack frame estándar, registros de 64-bit y llamadas a funciones del sistema.

3. La lógica de verificación de primos es similar, pero el manejo de la división, resto y control de bucles difiere debido a los registros y arquitectura.
  4. La salida en TASM se hace vía int 21h, mientras que en Linux se hace mediante `puts` de la libc.
-