p n #printea n

p /t n #printea n en binario

p /x n #printea n en hexadecimal

ptype n #printea el tipo (entero,char…) de variable que es n

whatis n #es lo mismo que la anterior

p &n #printea la dirección de memoria en la que está n

p $eax #printea el registro eax (sin el $ no va ya que le estás pidiendo que printee la

#variable eax no el registro)

info sources #te da info de donde está el fichero en ejecución y de donde se ha compilado

info registers #printea todos los registros en hexadecimal y decimal

layout split #para desplegar tres ventanas (una desensambla el codigo)

x /1db &sum #examina 1 byte de la dirección de memoria &sum en decimal (te pasa la

#dirección de memoria en hexadecimal y el valor que guarda en decimal

x /1tb &sum #lo mismo que la anterior pero en binario el valor en la dirección de memoria

x /1ob &sum #lo mismo que la anterior pero en octal el valor en la dirección de memoria

x /1xb &sum #lo mismo que la anterior pero en hexadecimal el valor en la dirección de

#memoria

help x #explica las letras para examinar en distintos lenguajes y distintos tamaños

#de dirección de memoria

shell lo que sea #desde el GDB para utilizar comandos de la terminal como si no

#estuvieras en GDB

layout regs #es como el layout split pero además de desensamblar te hace un

#display de todos los registros (o casi todos)

A partir del .s si lo ensamblas con as te da el .o y si se lo pasas al linker ld te devuelve el binario ejecutable. gcc directamente te lo pasa de .s al binario ejecutable.

x /xh &da2 #printear la dirección de memoria del array da2 y el primer elemento

x /4xh &da2 #lo mismo que el anterior pero printea 4 elementos del array

p (short[5])lista #printea un byte de los primeros 5 elementos del array lista

p /c (char[5])saludo #printea los cinco primeros caracteres del array de caracteres saludo

x /5c &saludo #mismo que anterior pero encima te printea la direccion de memoria

#de saludo

p /c \*(char \*)&saludo #printea el primer elemento del array saludo

p /s (char \*)&saludo #printea toda la string en saludo

p \*(short\*)(&lista+1) #printea el segundo elemento del array lista compuesto por shorts

x $esp #printea el stack pointer de la pila

x $sp #lo mismo que el anterior

x $ebp #printea el final de la pila

x $fp #lo mismo que el anterior

x /i &\_start #desensambla la etiqueta \_start