Compiling and Linking

nasm -f elf -g russian.asm
gcc -c -g rusmain.c
gcc -g russian.o rusmain.o -o russian

Running the Debugger

gdb ./russian

Getting Help

help : get list of classes of commands

help breakpoints / breakpoints
help running / running the program
help data / examining data
help info / list of info subcommands
info: generic command for showing things about the program being debugged

Breakpoints

break *function_name*
break *line_number*

clear *function_name*
clear *line_number*

Scenario:
break russian
run
continue

clear russian
break main
run
continue

clear main
break 8 // breakpoint at russian.asm line 8.

run
continue

Running

start : run the debugged program until the beginning of the main procedure
run : start debugged program
continue : continue program being debugged
next : step program
nexti : step one instruction
step : step program until it reaches a different source line
stepi : step one instruction exactly

Scenario
break russian
run
next / nexti / step / stepi
continue

clear russian

start // stops at beginning of main function
//see effects of next, nexti, step and stepi

Note: "next 5", "step 5", "nexti 5" and "stepi 5" repeat same 5 times

Info

info breakpoints
Scenario:
  break russian
  break 12
  info breakpoints
  clear russian
  clear 12

info address russian : describe what symbol is at location ADDR. Only for symbols with fixed locations
info address main
info address even
info address y : remember that y is not a symbol with a fixed location

info frame
backtrace
frame ...

Examining Registers

info registers : list of integer registers and their contents
Scenario:
```
  break russian
  run
  info registers
  info registers edx
//see how the register contents change after each operation
```

Examining Data

Examine memory: x/FMT ADDRESS
FMT is: a repeat count followed by a format letter and a size letter.
Format letters: o(octal), x(hex), d(decimal), u(unsigned decimal), t(binary), f(float), a(address), i(instruction), c(char) and s(string).
Size letters: b(byte), h(halfword), w(word), g(giant, 8 bytes).

Scenario:
```
info breakpoints
// clear all previous breakpoints
break russian
run  // breakpointte durur
info registers   // i r
   look at ebp and eip
   calculate ebp+8  and ebp+12
x/d [ebp+8]
x/d [ebp+12]
x/2d [ebp+8]
x/i [eip]
x/2i [eip]
```