

## 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
```

*Dikkat:* koyarken breakpoint at line 5 diyor ancak çalıştırılıp breakpoint noktasında durunca line 8'de duruyor. Baştaki function template kısmını atlıyor.

```
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 : hepsi aynı işi yapıyor burada -> GÖSTER  
continue

clear russian

start // stops at beginning of main function  
next  
...  
next // does not go into function, skips over it

start  
nexti  
...  
nexti // does go into function but does not show on debug screen (even goes into scanf) ????

start  
step  
...  
step // steps through the lines in source code (does not go into scanf)

start  
stepi  
...  
stepi // steps through all instructions, even scanf, printf etc

*Not:* "next 5", "step 5", "nexti 5" ve "stepi 5" aynı işlemi 5 kere tekrarlamak demektir.

## 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* : adresi göstermez

*info frame*

*backtrace*

*frame ...*

## Examining Registers

*info registers* : list of integer registers and their contents

Scenario:

break russian

run

info registers

info registers edx

//step ile devam edip her adımda saklayıcı içeriklerinin değişimini göster

## 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

ebp ve eip adreslerine bak

ebp+8 ve ebp+12 hesapla

x/d [ebp+8]

x/d [ebp+12]

x/2d [ebp+8]

x/i [eip]

x/2i [eip]