1.חיפוש בmn

2.חיפוש בobjdump - בשביל המשתנים הלוקליים.(4,5,7 ו11 לא נמצאים בmn כי הם משתנים לוקליים)

<u>הסבר לקוד אסמבלי:</u>

rbp - is the base pointer, which points to the base of the current stack frame.
rsp - is the stack pointer, which points to the top of the current stack frame.
eax - is a 32-bit general-purpose register with two common uses: to store the return value of a function and as a special register for certain calculations. It is technically a volatile register, since the value isn't preserved. Instead, its value is set to the return value of a function before a function returns.

```
avital@avital-Vostro-5468:~/Documents/final-work/q1$ nm a.out
0000000000201024 B __bss_start
0000000000201040 b completed.7698
                 w __cxa_finalize@@GLIBC_2.2.5
0000000000201000 D __data_start
0000000000201000 W data_start
00000000000005b0 t deregister_tm_clones
00000000000006a0 t doCalc
0000000000000640 t __do_global_dtors_aux
0000000000200db8 t __do_global_dtors_aux_fini_array_entry
0000000000201008 D __dso_handle
0000000000200dc0 d _DYNAMIC
0000000000201024 D _edata
0000000000bd5060 B _end
                 U exit@@GLIBC_2.2.5
00000000000007a4 T _fini
0000000000000680 t frame_dummy
0000000000200db0 t __frame_dummy_init_array_entry
0000000000000974 r __FRAME_END__
0000000000200fb0 d _GLOBAL_OFFSET_TABLE_
0000000000bc5060 B globBuf
                 w __gmon_start__
00000000000007e4 r __GNU_EH_FRAME_HDR
0000000000000528 T _init
0000000000200db8 t __init_array_end
0000000000200db0 t __init_array_start
00000000000007b0 R _IO_stdin_used
                 w _ITM_deregisterTMCloneTable
                 w _ITM_registerTMCloneTable
0000000000201020 d key.2775
00000000000007a0 T __libc_csu_fini
0000000000000730 T __libc_csu_init
                 U __libc_start_main@@GLIBC_2.2.5
0000000000000702 T main
0000000000201060 b mbuf.2776
0000000000201010 D primes
                 U printf@@GLIBC_2.2.5
00000000000005f0 t register_tm_clones
000000000000068a t square
0000000000000580 T _start
0000000000201028 D __TMC_END__
```

char globBuf[65536]; /* 1. Uninitialized data segment */

B - The symbol is in the uninitialized data section (known as BSS).
Global bss symbol.

int primes[] = { 2, 3, 5, 7 }; /* 2. Initialized data segment */

D - The symbol is in the initialized data section.
Global data symbol.

3,4,5 הם בקוד אסמבלי של square ולכן הם בframe שלו:

```
0000000000000000 <square>:
   0:   55                      push   %rbp
   1:   48 89 e5                mov    %rsp,%rbp
   4:   89 7d ec                mov    %edi,-0x14(%rbp)
   7:   8b 45 ec                mov    -0x14(%rbp),%eax
   a:   0f af 45 ec             imul   -0x14(%rbp),%eax
   e:   89 45 fc                mov    %eax,-0x4(%rbp)
  11:   8b 45 fc                mov    -0x4(%rbp),%eax
  14:   5d                      pop    %rbp
  15:   c3                      retq
```

square(int x) /* 3. Allocated in frame for square() */

t - The symbol is in the text (code) section.
Local text symbol.

השורות קוד באסמלי square(int x) :
של

```
0:  55          push  %rbp
1:  48 89 e5    mov   %rsp,%rbp
```

int result; /* 4. Allocated in frame for square() */

השורות קוד באסמלי של int result; :

```
4:  89 7d ec    mov   %edi,-0x14(%rbp)
```

return result; /* 5. Return value passed via register */

השורות קוד באסמלי של return result;

```
14:  5d         pop   %rbp
15:  c3         retq
```

ערך ההחזרה מוחזר על ידי רגיסטר שמור.

6,7 הם בקוד אסמבלי של doCalc ולכן הם בframe שלו:

```
0000000000000016 <doCalc>:
  16:   55                      push   %rbp
  17:   48 89 e5                mov    %rsp,%rbp
  1a:   48 83 ec 20             sub    $0x20,%rsp
  1e:   89 7d ec                mov    %edi,-0x14(%rbp)
  21:   8b 45 ec                mov    -0x14(%rbp),%eax
  24:   89 c7                   mov    %eax,%edi
  26:   e8 d5 ff ff ff          callq  0 <square>
  2b:   89 c2                   mov    %eax,%edx
  2d:   8b 45 ec                mov    -0x14(%rbp),%eax
  30:   89 c6                   mov    %eax,%esi
  32:   48 8d 3d 00 00 00 00    lea    0x0(%rip),%rdi        # 39 <doCalc+0x23>
  39:   b8 00 00 00 00          mov    $0x0,%eax
  3e:   e8 00 00 00 00          callq  43 <doCalc+0x2d>
  43:   81 7d ec e7 03 00 00    cmpl   $0x3e7,-0x14(%rbp)
  4a:   7f 29                   jg     75 <doCalc+0x5f>
  4c:   8b 45 ec                mov    -0x14(%rbp),%eax
  4f:   0f af 45 ec             imul   -0x14(%rbp),%eax
  53:   8b 55 ec                mov    -0x14(%rbp),%edx
  56:   0f af c2                imul   %edx,%eax
  59:   89 45 fc                mov    %eax,-0x4(%rbp)
  5c:   8b 55 fc                mov    -0x4(%rbp),%edx
  5f:   8b 45 ec                mov    -0x14(%rbp),%eax
  62:   89 c6                   mov    %eax,%esi
  64:   48 8d 3d 00 00 00 00    lea    0x0(%rip),%rdi        # 6b <doCalc+0x55>
  6b:   b8 00 00 00 00          mov    $0x0,%eax
  70:   e8 00 00 00 00          callq  75 <doCalc+0x5f>
  75:   90                      nop
  76:   c9                      leaveq
  77:   c3                      retq
```

doCalc(int val) /* 6. Allocated in frame for doCalc() */

t - The symbol is in the text (code) section.

Local text symbol.

השורות קוד באסמלי של doCalc(int val) :

```
16:   55                push   %rbp
17:   48 89 e5          mov    %rsp,%rbp
```

int t; /* 7. Allocated in frame for doCalc() */

השורה הנ"ל אינה מאותחלת, ולכן היא ספציפית אינה מופיעה,

מפני שאחר כך יש בה שימוש (t = val*val*val) מופיעות ארבעת השורות הבאות באסמלי:

```
4c:   8b 45 ec        mov   -0x14(%rbp),%eax
4f:   0f af 45 ec     imul  -0x14(%rbp),%eax
53:   8b 55 ec        mov   -0x14(%rbp),%edx
```

56: 0f af c2 imul %edx,%eax

8,11 הם בקוד אסמבלי של main ולכן הם בframe שלו:

```
0000000000000078 <main>:
  78:   55                      push   %rbp
  79:   48 89 e5                mov    %rsp,%rbp
  7c:   48 83 ec 10             sub    $0x10,%rsp
  80:   89 7d fc                mov    %edi,-0x4(%rbp)
  83:   48 89 75 f0             mov    %rsi,-0x10(%rbp)
  87:   8b 05 00 00 00 00       mov    0x0(%rip),%eax        # 8d <main+0x15>
  8d:   89 c7                   mov    %eax,%edi
  8f:   e8 82 ff ff ff          callq  16 <doCalc>
  94:   bf 00 00 00 00          mov    $0x0,%edi
  99:   e8 00 00 00 00          callq  9e <main+0x26>
```

main(int argc, char* argv[]) /* 8. Allocated in frame for main() */

T - The symbol is in the text (code) section.

Global text symbol.

השורות קוד באסמלי של main(int argc, char* argv[]) :

```
78:  55            push  %rbp
79:  48 89 e5      mov   %rsp,%rbp
7c:  48 83 ec 10   sub   $0x10,%rsp
80:  89 7d fc      mov   %edi,-0x4(%rbp)
83:  48 89 75 f0   mov   %rsi,-0x10(%rbp)
```

static int key = 9973; /* 9. Initialized data segment */

d - The symbol is in the initialized data section.

Local data symbol.

static char mbuf[10240000]; /* 10. Uninitialized data segment */

b - The symbol is in the uninitialized data section (known as BSS).

Local bss symbol.

char* p; /* 11. Allocated in frame for main() */

השורה הנ"ל אינה מאותחלת, ולכן היא ספציפית אינה מופיעה בקוד אסמבלי, אך היא כתובה בפונקצית main ולכן היא Allocated in frame for main().