CS 33: Introduction to Computer Organization

TA: Aalisha Dalal

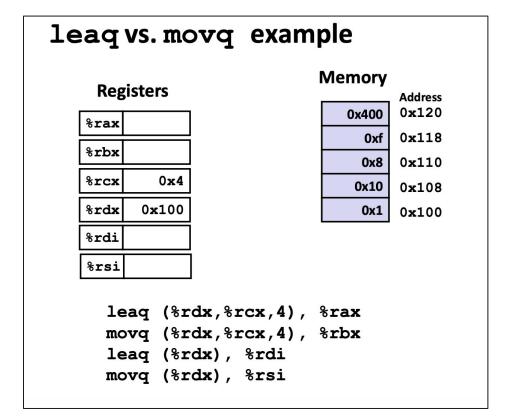
LA: Jonathan Myong

Office Hours: Friday, 9:30-11:30AM

Focus of the Discussion

- Queries from previous discussion
- Stack Trace
- Arrays memory representation
- Worksheet Problems

leaq vs. movq instruction



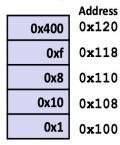
leaq vs. movq instruction

leaq vs. movq example (solution)

Registers

%rax	0x110
%rbx	0 x 8
%rcx	0 x 4
%rdx	0x100
%rdi	0x100
%rsi	0x1

Memory



```
leaq (%rdx,%rcx,4), %rax
movq (%rdx,%rcx,4), %rbx
leaq (%rdx), %rdi
movq (%rdx), %rsi
```

CMP Instruction

0x000000000004004f2 <+26>:

0x000000000004004f3 <+27>:

End of assembler dump.

```
(gdb) list
        int max(int a, int b)
          if(a<b)
            return b;
          else
                                                    0x000000000004004e5 <+13>:
                                                                                     cmp
                                                                                             -0x8(%rbp),%eax
            return a;
                                                    0x000000000004004e8 <+16>:
                                                                                             0x4004ef < max + 23 >
                                                                                     jge
                                                    0x000000000004004ea <+18>:
                                                                                     mov
                                                                                             -0x8(%rbp),%eax
                                                    0x000000000004004ed <+21>:
                                                                                             0x4004f2 < max + 26 >
                                                                                     jmp
10
                                                    0x000000000004004ef <+23>:
                                                                                             -0x4(\%rbp),%eax
                                                                                     mov
(adb) disassemble max
Dump of assembler code for function max:
   0x000000000004004d8 <+0>:
                                 push
                                        %rbp
   0x000000000004004d9 <+1>:
                                        %rsp,%rbp
                                 mov
   0x000000000004004dc <+4>:
                                        %edi,-0x4(%rbp)
                                 mov
   0x00000000004004df <+7>:
                                        %esi,-0x8(%rbp)
                                 mov
                                        -0x4(%rbp),%eax
   0x000000000004004e2 <+10>:
                                 mov
   0x000000000004004e5 <+13>:
                                        -0x8(%rbp),%eax
                                 cmp
   0x000000000004004e8 <+16>:
                                 jge
                                         0x4004ef < max + 23 >
   0x000000000004004ea <+18>:
                                         -0x8(%rbp),%eax
                                 mov
   0x000000000004004ed <+21>:
                                 qmj
                                         0x4004f2 < max + 26 >
   0x000000000004004ef <+23>:
                                 mov
                                         -0x4(%rbp),%eax
```

%rbp

pop

reta

CMP Instruction - II

retq

0x000000000004004f3 <+27>:

End of assembler dump.

```
(gdb) list
       int max(int a, int b)
                                       0x000000000004004e5 <+13>:
                                                                                         -0x8(%rbp),%eax
                                                                               cmp
         if(a>b)
           return a;
                                       0x000000000004004e8 <+16>:
                                                                               jle
                                                                                         0x4004ef < max + 23 >
         else
                                       0x000000000004004ea <+18>:
                                                                                         -0x4(%rbp),%eax
                                                                               mov
           return b;
                                       0x000000000004004ed <+21>:
                                                                                        0x4004f2 < max + 26 >
                                                                               jmp
       }
                                       0x000000000004004ef <+23>:
                                                                                        -0x8(%rbp),%eax
                                                                               mov
9
10
(adb) disassemble max
Dump of assembler code for function max:
   0x00000000004004d8 <+0>:
                              push
                                     %rbp
   0x000000000004004d9 <+1>:
                                     %rsp,%rbp
                              mov
   0x000000000004004dc <+4>:
                                     \%edi,-0x4(\%rbp)
                              mov
   0x00000000004004df <+7>:
                                     %esi,-0x8(%rbp)
                              mov
   0x000000000004004e2 <+10>:
                                     -0x4(\%rbp),%eax
                              mov
   0x000000000004004e5 <+13>:
                                     -0x8(%rbp),%eax
                              cmp
                                     0x4004ef <max+23>
   0x000000000004004e8 <+16>:
                              ile
   0x000000000004004ea <+18>:
                                     -0x4(%rbp),%eax
                              mov
   0x000000000004004ed <+21>:
                                     0x4004f2 <max+26>
                              ami
   0x000000000004004ef <+23>:
                                     -0x8(%rbp),%eax
                              mov
   0x000000000004004f2 <+26>:
                              gog
                                     %rbp
```

Understanding Function Stack Trace

```
11
         int main()
12
13
14
         int a = 14;
         int b = 17;
15
16
         int c = max(a, b);
17
18
19
         return 0;
20
```

```
(qdb) disassemble main
Dump of assembler code for function main:
   0x000000000004004f4 <+0>:
                                 push
                                        %rbp
   0x000000000004004f5 <+1>:
                                        %rsp,%rbp
                                 mov
   0x000000000004004f8 <+4>:
                                 sub
                                        $0x10,%rsp
                                        $0xe,-0x4(%rbp)
   0x00000000004004fc <+8>:
                                 movl
   0x00000000000400503 <+15>:
                                 mov1
                                        $0x11,-0x8(%rbp)
   0x0000000000040050a <+22>:
                                        -0x8(%rbp),%edx
                                 mov
   0x000000000040050d <+25>:
                                        -0x4(%rbp),%eax
                                 mov
   0x00000000000400510 <+28>:
                                        %edx,%esi
                                 mov
   0x00000000000400512 <+30>:
                                        %eax,%edi
                                 mov
                                 callo
                                        0x4004d8 <max>
   0x00000000000400514 <+32>:
   0x00000000000400519 <+37>:
                                        %eax,-0xc(%rbp)
                                 mov
   0x0000000000040051c <+40>:
                                        $0x0,%eax
                                 mov
   0x00000000000400521 <+45>:
                                 leaved
   0x00000000000400522 <+46>:
                                 reta
End of assembler dump.
```

Understanding Function Stack Trace

End of assembler dump.

```
(gdb)
           list
               int max(int a, int b)
                                                                                                                      Low
                                                                                                                     Address
                  if(a>b)
3
                      return a;
5
                  else
6
                      return b;
                                                                                                      arg 'a'
                            (gdb) disassemble max
                                                                                        %esp -
                            Dump of assembler code for function max:
8
                               0x00000000004004d8 <+0>:
                                                           push
                                                                  %rbp
                               0x000000000004004d9 <+1>:
                                                                  %rsp,%rbp
                                                           mov
                                                                                                      arg 'b'
9
                               0x000000000004004dc <+4>:
                                                                  %edi,-0x4(%rbp)
                                                           mov
                               0x00000000004004df <+7>:
                                                                  \%esi,-0x8(\%rbp)
                                                           mov
10
                               0x000000000004004e2 <+10>:
                                                                  -0x4(%rbp),%eax
                                                           mov
                               0x00000000004004e5 <+13>:
                                                                  -0x8(%rbp),%eax
                                                                                                    %ebp (old)
                                                           cmp
                               0x00000000004004e8 <+16>:
                                                           ile
                                                                  0x4004ef <max+23>
                                                                  -0x4(%rbp),%eax
                               0x000000000004004ea <+18>:
                                                           mov
                               0x000000000004004ed <+21>:
                                                           dmi
                                                                  0x4004f2 <max+26>
                                                                                                                      High
                                                                                                       %eip
                               0x000000000004004ef <+23>:
                                                                  -0x8(%rbp),%eax
                                                           mov
                                                                                                                    Address
                               0x000000000004004f2 <+26>:
                                                           pop
                                                                  %rbp
                               0x000000000004004f3 <+27>:
                                                           retq
```

Array Representation

$$A = \begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \\ x_{41} & x_{42} & x_{43} & x_{44} \end{bmatrix}$$

How would we access this location from the starting memory address

Worksheet Problems

https://tinyurl.com/y6lgkjcj

Questions