CS 354 Machine Organization and Programming

Lecture 20

Michael Doescher Summer 2020 **Function Call Practice**

Arrays

Structures

Security

royal-13.cs.wisc.edu - PuTTY - 0 X 1 int increment(int x){ .file "increment1.c" return x+1; .text .glob1 increment increment, @function .type 5 int main(){ increment: int f = 10; endbr32 f = increment(f); pushl %ebp return 0; movl %esp, %ebp 8(%ebp), %eax movl 10 addl \$1, %eax 11 popl %ebp 12 ret 13 .size increment, .-increment 14 .globl main 15 main, @function .type 16 main: 17 endbr32 18 pushl %ebp %esp, %ebp 19 movl 20 subl \$16, %esp 21 movl \$10, -4(%ebp) 22 -4 (%ebp) pushl 23 call increment 24 addl \$4, %esp 25 movl %eax, -4(%ebp) 26 movl \$0, %eax 27 leave 28 ret 29 .size main, .-main .ident "GCC: (Ubuntu 9.3.0-10ubuntu2) 9.3.0"

31 .section .note.GNU-stack,"",@progbits 9,1 All increment1.s 31, 1-4increment1.c Top

30

```
royal-13.cs.wisc.edu - PuTTY
```

```
- 0 X
 1 void increment (int *x) {
                                                                      .file
                                                                              "increment2.c"
       *x = *x + 1;
                                                                      .text
                                                                      .glob1
                                                                              increment
                                                                              increment, @function
                                                                      .type
 5 int main(){
                                                                  increment:
       int f = 10;
                                                                      endbr32
       increment(&f);
                                                                      pushl
                                                                               %ebp
       return 0;
                                                                      movl
                                                                              %esp, %ebp
                                                                      movl
                                                                              8(%ebp), %eax
10
                                                               10
                                                                              (%eax), %eax
                                                                      movl
11
                                                               11
                                                                              1(%eax), %edx
                                                                      leal
12
                                                               12
                                                                      movl
                                                                              8(%ebp), %eax
13
                                                               13
                                                                              %edx, (%eax)
                                                                      movl
14
                                                               14
                                                                      nop
15
                                                               15
                                                                              %ebp
                                                                      popl
16
                                                               16
                                                                      ret
17
                                                               17
                                                                      .size
                                                                              increment, .-increment
18
                                                               18
                                                                      .globl
                                                                              main
19
                                                               19
                                                                              main, @function
                                                                      .type
20
                                                               20 main:
21
                                                               21
                                                                      endbr32
22
                                                               22
                                                                      pushl
                                                                               %ebp
23
                                                               23
                                                                      movl
                                                                              %esp, %ebp
24
                                                               24
                                                                      subl
                                                                              $16, %esp
25
                                                               25
                                                                              $10, -4(%ebp)
                                                                      movl
26
                                                               26
                                                                      leal
                                                                              -4(%ebp), %eax
27
                                                               27
                                                                      pushl
                                                                               %eax
                                                               28
                                                                      call
                                                                              increment
                                                               29
                                                                      addl
                                                                              $4, %esp
                                                                      movl
                                                               30
                                                                              $0, %eax
                                                               31
                                                                      leave
```

increment2.c 4,0-1 All increment2.s 13,1-4 Top

```
1 int strange(int *p1, int p2) {
                                                               13
                                                                       imull
                                                                                %edx, %eax
       int res;
                                                               14
                                                                               ext{%eax}, -4(ext{%ebp})
                                                                       movl
       res = *p1 * p2;
                                                               15
                                                                       movl
                                                                               8(%ebp), %eax
       *p1 = *p1 + p2;
                                                               16
                                                                       movl
                                                                                (%eax), %edx
                                                               17
                                                                                12(%ebp), %eax
       return res;
                                                                       movl
                                                               18
                                                                       addl
                                                                                %eax, %edx
                                                               19
                                                                       movl
                                                                               8(%ebp), %eax
 8 int main() {
                                                               20
                                                                               %edx, (%eax)
                                                                       movl
       int x=3;
                                                               21
                                                                       movl
                                                                                -4(%ebp), %eax
10
                                                               22
       int y=4;
                                                                       leave
11
       int z = strange(&x, y);
                                                               23
                                                                       ret
12 }
                                                               24
                                                                       .size
                                                                               strange, .-strange
13
                                                               25
                                                                       .globl
                                                                               main
                                                               26
                                                                       .type
                                                                               main, @function
                                                               27
                                                                  main:
                                                               28
                                                                       endbr32
                                                               29
                                                                       pushl
                                                                                %ebp
                                                               30
                                                                       movl
                                                                               %esp, %ebp
                                                               31
                                                                       subl
                                                                               $16, %esp
                                                               32
                                                                               $3, -12(%ebp)
                                                                       movl
                                                               33
                                                                       movl
                                                                               $4, -4(%ebp)
                                                               34
                                                                               -4 (%ebp)
                                                                       pushl
                                                                35
                                                                       leal
                                                                                -12(%ebp), %eax
                                                               36
                                                                       pushl
                                                                               %eax
                                                               37
                                                                       call
                                                                               strange
                                                               38
                                                                       addl
                                                                               $8, %esp
                                                               39
                                                                       movl
                                                                               %eax, -8 (%ebp)
                                                               40
                                                                       movl
                                                                                $0, %eax
                                                               41
                                                                       leave
                                                               42
                                                                       ret
                                                               43
                                                                       .size
                                                                               main, .-main
```

strange.c 13,0-1 All strange.s 31,1-4 40%