## Wrap-up of

# Machine-Level Programming II: Control

**15-213: Introduction to Computer Systems**
**Sept. 18, 2018**

# Switch Statement Example

```
long my_switch
   (long x, long y, long z)
{
    long w = 1;
    switch(x) {
    case 1:
.L3:    w = y*z;
        break;
    case 2:
.L5:    w = y/z;
        /* Fall Through */
    case 3:
.L9:    w += z;
        break;
    case 5:
    case 6:
.L7:    w -= z;
        break;
    default:
.L8:    w = 2;
    }
    return w;
}
```

```
my_switch:
   cmpq     $6, %rdi    # x:6
   ja       .L8    # if x > 6 jump
                   # to default
   jmp      *.L4(,%rdi,8)
```

```
.section    .rodata
  .align 8
.L4:
  .quad     .L8   # x = 0
  .quad     .L3   # x = 1
  .quad     .L5   # x = 2
  .quad     .L9   # x = 3
  .quad     .L8   # x = 4
  .quad     .L7   # x = 5
  .quad     .L7   # x = 6
```

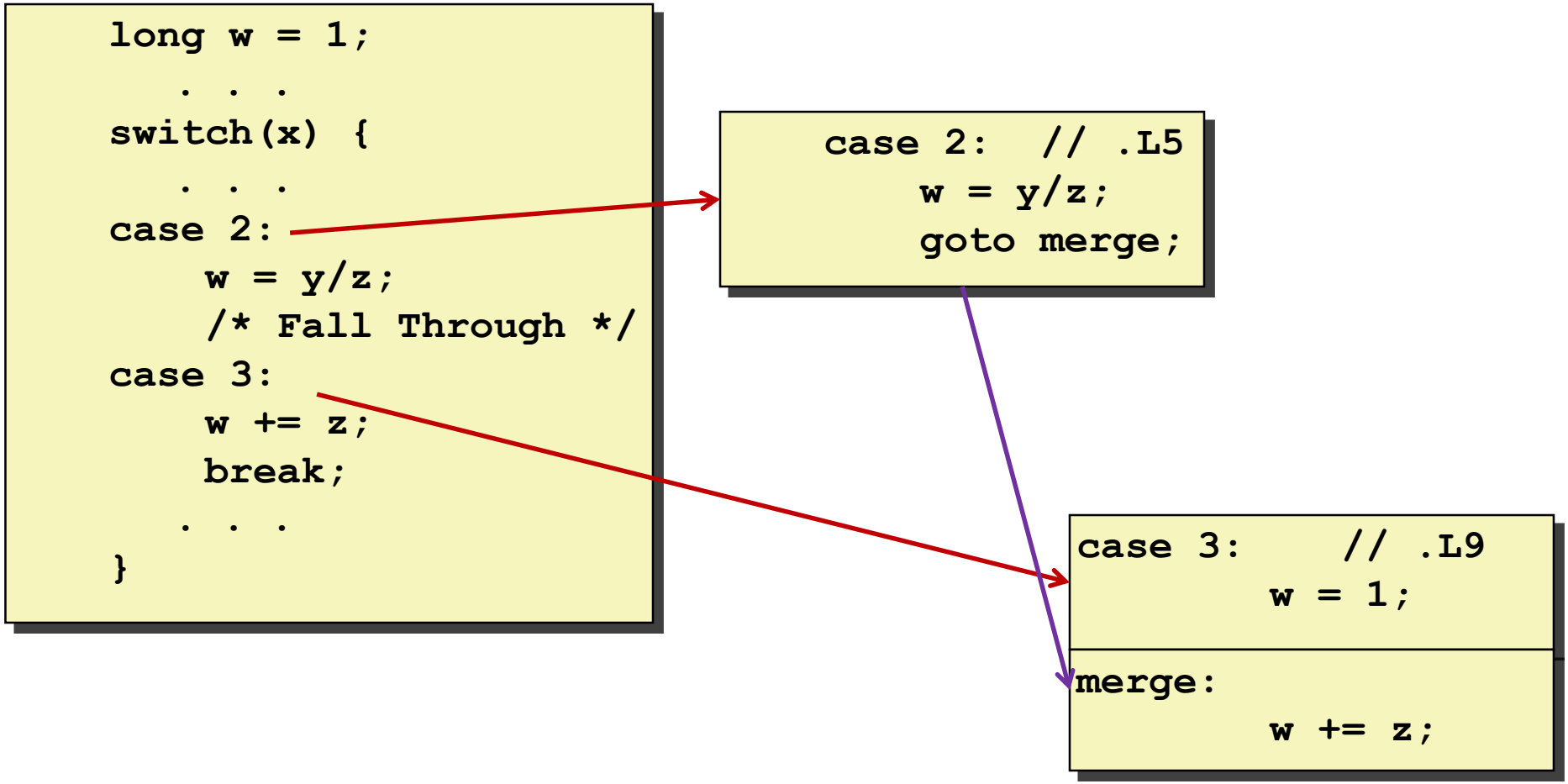
# Code Blocks (x == 1)

```
  switch(x) {
  case 1:         // .L3
        w = y*z;
        break;
    . . .
}
```

```
.L3:
   movq     %rsi, %rax   # y
   imulq    %rdx, %rax   # y*z
   ret
```

| Register | Use(s) |
|---|---|
| %rdi | Argument x |
| %rsi | Argument y |
| %rdx | Argument z |
| %rax | Return value |

# Handling Fall-Through (x ==2, x==3)



# Code Blocks (x == 5, x == 6, default)

```
 switch(x) {
   long w = 1;
   switch(x) {
   . . .
   case 5:  // .L7
   case 6:  // .L7
       w -= z;
       break;
   default: // .L8
       w = 2;
}
```

```
.L7:                    # Case 5,6
  movl  $1, %eax        #  w = 1
  subq  %rdx, %rax      #  w -= z
  ret
.L8:                    # Default:
  movl  $2, %eax        #  2
  ret
```

| Register | Use(s) |
|---|---|
| %rdi | Argument x |
| %rsi | Argument y |
| %rdx | Argument z |
| %rax | Return value |