After examining the examples of switch cases and creating a few of our own, we found a couple things to be true about how switch cases work. To begin, the compiler first takes the value given to the switch function (call this x), and moves it to a register. From there, the compiler could do a couple different things.

If there are more than 3 cases specified separately from the default case, the compiler will then take the switch value, x, and compare it to the lowest case value. If x is determined to be equal to the case value, the compiler will then receive a jump execution to a given label where it will continue with the execution. However, if x is not equal to the case value, the compiler will then get the next lowest case value and repeat the same process with this new case value. If the switch value is equal to the case value for any of the remaining case value, the compiler will jump the correct label and continue the execution. On the other hand, if the compiler gets through all the case value without any matching the switch value, it will then do an unconditional jump to the default case of the switch statement. When the compiler has finished either the default case, or the corresponding code to a matching case value, the compiler will jump to the label that contains the end of the switch statement.

For example if we examine the code given from switch\_int1.c :

*void main() {*

*int x = 4;*

*int i;*

*switch (x) {*

*case 3:*

*i += 3;*

*break;*

*case -4:*

*i -= 4;*

*break;*

*case 50:*

*i += 50;*

*break;*

*default:*

*i += 1;*

*}*

*}*

The compiler will first take the value of x, 4, and place it in a register (eax). \*\*\*\*\*tv64\*\*\*\*\* - we will call this value the switch value. The first case that the compiler compares the switch value to will be the -4 case (the second case listed in the code). Since 4 is not equal to -4, the compiler does not jump to the case -4 label and instead moves on to the 3 case (the first case listed in the code). Since 4 is not equal to 3, the compiler does not jump to the case 3 label.