

# Translation of Loop

## While loop

Take an example of this C code. This code is inside some function.

```
int a = 4;
int b = 8;
int d = 0;
while(b > a)
{
    d = a + 2;
    b--;
}
```

Generated assembly code:

```
movl  $4, -4(%ebp)
movl  $8, -8(%ebp)
movl  $0, -12(%ebp)
jmp   .L2
.L3:
movl  -4(%ebp), %eax
addl  $2, %eax
movl  %eax, -12(%ebp)
subl  $1, -8(%ebp)
.L2:
movl  -8(%ebp), %eax
cmpl  -4(%ebp), %eax
jg    .L3
```

Location of local variables of the stack (local variables are explained here ([memorymanagement.html](http://memorymanagement.html)))

```
a => -4(%ebp)
b => -8(%ebp)
d => -12(%ebp)
```

The use of registers as temporary memory is described here ([arithmeticop.html#tempVaribaleUsage](http://arithmeticop.html#tempVaribaleUsage))

Comments on generated assembly code

```

# a = 4
    movl    $4, -4(%ebp)

# b = 8
    movl    $8, -8(%ebp)

# d = 0
    movl    $0, -12(%ebp)

# jump to label .L2. The condition for the while loop is evaluated at .L2
    jmp     .L2

# The label .L3. This is the start instruction inside while {}
.L3:

# tmp = a
    movl    -4(%ebp), %eax

# tmp = tmp + 2
    addl    $2, %eax

# d = tmp
    movl    %eax, -12(%ebp)

# b = b - 1
    subl    $1, -8(%ebp)

# The instruction for evaluating the condition of while starts here
.L2:

# tmp = b
    movl    -8(%ebp), %eax

# compare a to tmp
    cmpl    -4(%ebp), %eax

# jump to start of loop block if the above tmp greater than a in above comparison
    jg      .L3

```

## For Loop

Take example of this C code:

```
int a = 4;
int b = 8;
int d = 0;
for(b = 9; b > a; b--)
{
    d = a + 2;
}
```

Generated assembly code:

```
movl    $4, -4(%ebp)
movl    $8, -8(%ebp)
movl    $0, -12(%ebp)
movl    $9, -8(%ebp)
jmp     .L2
.L3:
movl    -4(%ebp), %eax
addl    $2, %eax
movl    %eax, -12(%ebp)
subl    $1, -8(%ebp)
.L2:
movl    -8(%ebp), %eax
cmpl    -4(%ebp), %eax
jg      .L3
```

The generated code is almost the same as the while loop. Here are few things to notice:

- The init condition of the loop is generated before the code for if-block.
- The update statement of the loop is generated at the end of the code of the if-block.

---

◀ [Translation of Branch Statement \(/cin/branchstmt.html\)](/cin/branchstmt.html)

[up \(/cin/cin.html\)](/cin/cin.html)

[Pointer Dereferencing \(/cin/pointer.html\)](/cin/pointer.html) ▶

---

**Do you collaborate using whiteboard? Please try Lekh Board - An Intelligent Collaborate Whiteboard App (<https://lekh.app>)**