

```

loop:
    body-statement
    update-expr;
test:
    t = test-expr;
    if (t)
        goto loop;

```

而 guarded-do 策略得到:

```

    init-expr;
    t = test-expr;
    if (!t)
        goto done;
loop:
    body-statement
    update-expr;
    t = test-expr;
    if (t)
        goto loop;

```

done:

作为一个示例, 考虑用 for 循环写的阶乘函数:

```

long fact_for(long n)
{
    long i;
    long result = 1;
    for (i = 2; i <= n; i++)
        result *= i;
    return result;
}

```

如上述代码所示, 用 for 循环编写阶乘函数最自然的方式就是从 2 一直到 n 的因子乘起来, 因此, 这个函数与我们使用 while 或者 do-while 循环的代码很不一样。

这段代码中的 for 循环的不同组成部分如下:

```

init-expr      i = 2
test-expr      i <= n
update-expr    i++
body-statement result *= i;

```

用这些部分替换前面给出的模板中相应的位置, 就把 for 循环转换成了 while 循环, 得到下面的代码:

```

long fact_for_while(long n)
{
    long i = 2;
    long result = 1;
    while (i <= n) {
        result *= i;
        i++;
    }
    return result;
}

```