

```

long fact_while(long n)
n in %rdi
fact_while:
1  cmpq    $1, %rdi      Compare n:1
2  jle     .L7           If <=, goto done
3  movl    $1, %eax      Set result = 1
4  .L6:                                loop:
5  imulq   %rdi, %rax     Compute result *= n
6  subq    $1, %rdi      Decrement n
7  cmpq    $1, %rdi      Compare n:1
8  jne     .L6           If !=, goto loop
9  rep; ret              Return
10 .L7:                                done:
11 movl    $1, %eax      Compute result = 1
12 ret                                Return
13

```

c) 对应的汇编代码

图 3-21 (续)



练习题 3.25 对于如下 C 代码：

```

long loop_while2(long a, long b)
{
    long result = _____;
    while (_____) {
        result = _____;
        b = _____;
    }
    return result;
}

```

以命令行选项 -O1 运行 GCC，产生如下代码：

```

a in %rdi, b in %rsi
1  loop_while2:
2  testq   %rsi, %rsi
3  jle     .L8
4  movq    %rsi, %rax
5  .L7:
6  imulq   %rdi, %rax
7  subq    %rdi, %rsi
8  testq   %rsi, %rsi
9  jg      .L7
10 rep; ret
11 .L8:
12 movq    %rsi, %rax
13 ret

```

可以看到编译器使用了 guarded-do 的翻译方法，在第 3 行使用了 jle 指令使得当初测试不成立时，忽略循环代码。填写缺失的 C 代码。注意汇编语言中的控制结构不一定与根据翻译规则直接翻译 C 代码得到的完全一致。特别地，它有两个不同的 ret 指令（第 10 行和第 13 行）。不过，你可以根据等价的汇编代码行为填写 C 代码中缺失的部分。