

1 递归中的栈帧进出

请按照要求，产生可执行文件，并给出每一个步骤的截图。以下要求使用 Linux 或者 Windows 的 GCC 工具。

1.1 阶乘运算的代码文件

以下包括主函数和用于自定义的函数。

1.1.1 bar.h 文件

```
1 long long factr(long long);
```

1.1.2 bar.c 文件

```
1 #include <stdio.h>
2 long long factr(long long n)
3 {
4     printf("calculating factr(%lld), addr of its single
5           parameter: %p\n", n, &n);
6     return (n == 1) ? 1 : n * factr(n - 1);
7 }
```

1.1.3 main1.c 文件

```
1 #include <stdio.h>
2 #include "bar.h"
3
4 int main()
5 {
6     long long m = 4;
7     int f = factr(m);
8     printf("factorial res: %lld\n", f);
9     return 0;
10 }
```

1.1.4 步骤

1. 根据以上代码产生可执行文件，并运行得到输出结果。
2. 按开始被调用的先后顺序给出写出各个被调用的函数并指出它们被调用时传入的参数值（main() 函数可以不写）。
3. 按调用结束的先后顺序给出写出各个被调用的函数指出它们被调用时传入的参数值（main() 函数可以不写）。
4. 以上两个顺序有何关系？
5. 用文字或者示意图，表示各个函数（包括传入的参数值）开始被调用以及调用结束的先后发生流程。
6. 在 factr() 函数被调用的过程中，参数 n 的值，以及它的地址，有何规律或者有何变化规律？
7. 以上实验结果说明什么？

1.2 斐波那契数的代码文件

以下包括主函数和用于自定义的函数。

1.2.1 foo.h 文件

```
1 long long fibonnaci(long long);
```

1.2.2 foo.c 文件

```
1 #include <stdio.h>
2 int fibonnaci(long long n)
3 {
4     printf("calculating fabonnaci(%lld), addr of its single
5           parameter: %p\n", n, &n);
6     return (n == 1) || (n == 2) ? 1 : fibonnaci(n - 1) +
7           fibonnaci(n - 2);
8 }
```

1.2.3 main2.c 文件

```
1 #include <stdio.h>
2 #include "bar.h"
3
4 int main()
5 {
6     long long m = 5;
7     long long f = fibonnaci(m);
8     printf("factorial res: %lld\n", f);
9     return 0;
10 }
```

1.2.4 步骤

1. 根据以上代码产生可执行文件，并运行得到输出结果。
2. 按开始被调用的先后顺序给出写出各个被调用的函数并指出它们被调用时传入的参数值（main() 函数可以不写）。
3. 按调用结束的先后顺序给出写出各个被调用的函数指出它们被调用时传入的参数值（main() 函数可以不写）。
4. 以上两个顺序有何关系？
5. 用文字或者示意图，表示各个函数（包括传入的参数值）开始被调用以及调用结束的先后发生流程。
6. 在 fibonacci() 函数被调用的过程中，参数 n 的值，以及它的地址，有何规律或者有何变化规律？
7. 以上实验结果说明什么？

1.3 实验报告写作要求

1. 步骤详细；
2. 表述简明；
3. 图文并茂；

4. 逻辑流畅。