

Linux C 5 深入理解函数

Subtitle

2022/10/05

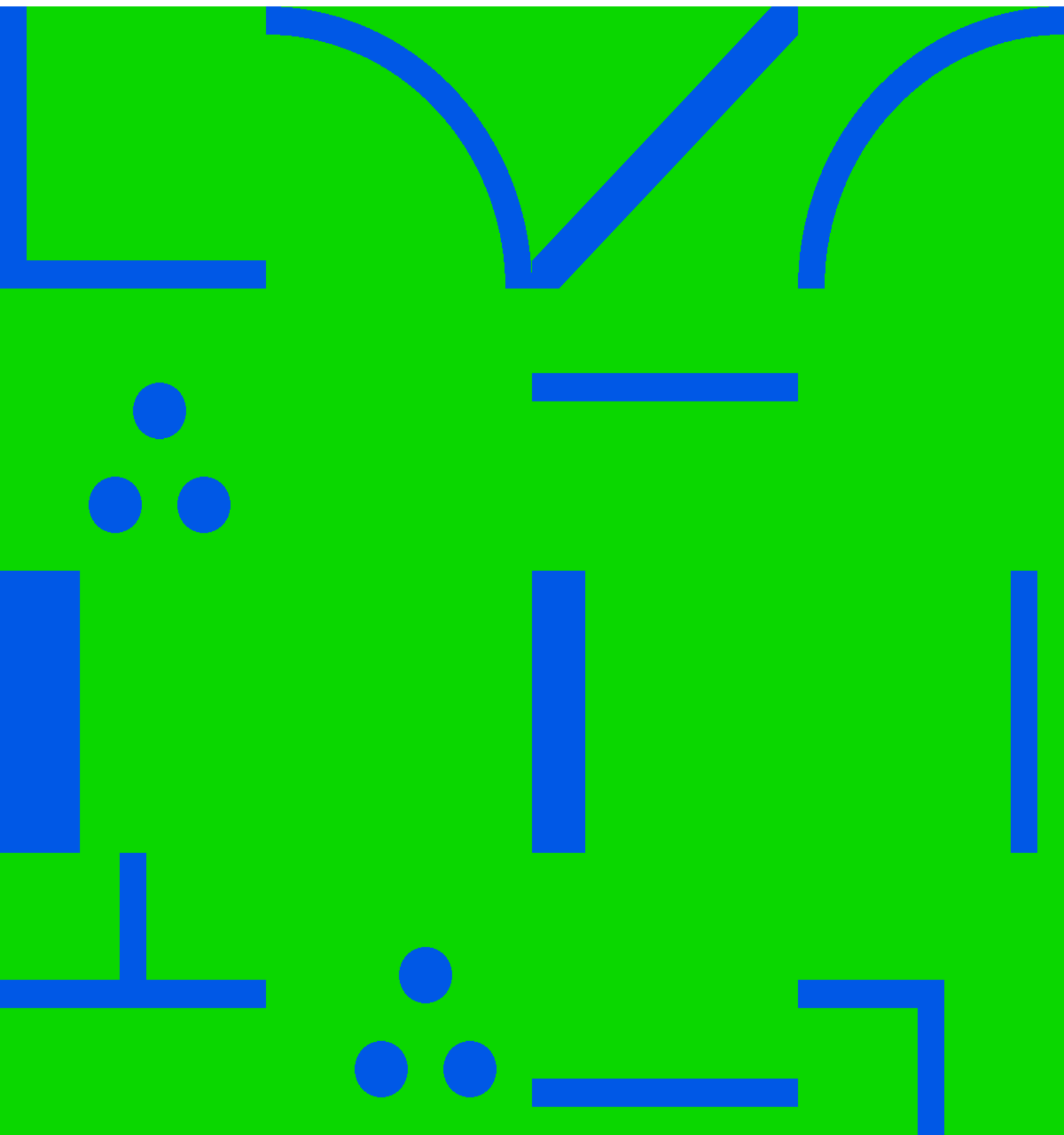


Table of Contents

- Linux C 5 深入理解函数 1
 - 递归 1
 - 求阶乘 1
 - 最大公约数 1
 - Fibonacci数列 2

Linux C 5 深入理解函数

递归

求阶乘

factorial.c

```
#include <stdio.h>
#include <stdlib.h>

int factorial (int n)
{
    if(n==0)
        return 1;
    else
        return n*factorial(n-1);
}

int main(int argc, char* argv[])
{
    int n;
    printf("input n: ");
    scanf("%d",&n);
    printf("%d",factorial(n));
    system("pause");
    return 0;
}
```

最大公约数

1、编写递归函数求两个正整数a和b的最大公约数（GCD□Greatest Common Divisor），使用Euclid算法(🐼欧几里得算法)：

1. 如果a除以b能整除，则最大公约数是b
2. 否则，最大公约数等于b和a%b的最大公约数。

Euclid算法是很容易证明的，请读者自己证明一下为什么这么算就能算出最大公约数。最后，修改你的程序使之适用于所有整数，而不仅仅是正整数。

gcd.c

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
int gcd (int a,int b)
{
    if(b==0)
        return abs(a); //约数为正？
}
```

```
    else
        return gcd(b,a%b);
}

int main(int argc, char* argv[])
{
    printf("input a and b: ");
    int a,b;
    scanf("%d%d",&a,&b);
    printf("%d",gcd(a,b));
    system("pause");
}
```

Fibonacci数列

2、编写递归函数求Fibonacci数列的第n项，这个数列是这样定义的：

$\text{fib}(0)=1$ $\text{fib}(1)=1$ $\text{fib}(n)=\text{fib}(n-1)+\text{fib}(n-2)$

上面两个看似毫不相干的问题之间却有一个有意思的联系：

Lamé定理 如果Euclid算法需要k步来计算两个数的GCD，那么这两个数之中较小的一个必然大于等于Fibonacci数列的第k项。

[fibonacci.c](#)

```
#include <stdio.h>
#include <stdlib.h>

int factorial (int n)
{
    if(n==0)
        return 1;
    else
        return n*factorial(n-1);
}

int main(int argc, char* argv[])
{
    int n;
    printf("input n: ");
    scanf("%d",&n);
    printf("%d",factorial(n));
    system("pause");
    return 0;
}
```

Printed on: 2022/10/05 17:12

Convert to img Failed!