

## 西邮 Linux 兴趣小组 2019 纳新试题

小伙伴们：

你们好！欢迎来到西邮 Linux 兴趣小组。无论你是刚刚进入大学，立志奋发图强但却不知从何做起、亦或是在过去的一年里熟悉了大学的节奏，想要找到志同道合的小伙伴们一同前行，加入小组都是你的不二之选。小组的学长学姐们都在毕业后感叹，小组改变了他们的人生。

或许你曾经听说，小组的面试题目很难，一度让你失去了信心。但既然你拿到了纳新试题，何不自己做一做尝试一下呢？对技术的热爱，对知识的渴求，将比你现在所了解的一星半点知识更能使你受益终生。何况，我们对这些题目的考察仅仅作为面试的有限参考，并且在面试前，你亦可以使用手边的资源进行学习。

最后，下面的题目都有一些共同的前提条件：

- ① 为了代码的简洁，我们略去了大部分不影响理解的预处理指令；
- ② 所有代码均假设在 Linux x86\_64 下使用 gcc 编译器编译；
- ③ 题目的难度与序号并无直接关系；

我们在小组等你！

西邮 Linux 兴趣小组  
2018 年 11 月 28 日

1. 解释下面 C 语言代码段中每一句的语义。

```
int main(int argc, char *argv[])
{
    int val = 2018;
    int *pi = 2019;
    pi = &val;
    *pi = 0;
    return 0;
}
```

2. 下面的代码段将打印出多少个“=”？运用相关知识解释该输出。

```
int main(int argc, char *argv[])
{
    for (unsigned int i = 3; i >= 0; i--)
    {
        putchar('=');
    }
    return 0;
}
```

3. 下面是三种交换整数的方式，它们是怎样实现交换的？

```
/* (1) */ int c = a; a = b ; b = c ;
/* (2) */ a = a - b; b = b + a; a = b - a;
/* (3) */ a = a ^ b; b = b ^ a; a = a ^ b;
```

4. 有如下代码段所示的函数  $f$ ，当我们执行该函数时，会产生什么样的输出结果？在同一程序中多次执行该函数，输出结果是否一致？

```
void f()
{
    static int a = 0;
    int b = 0;
    printf("%d, %d\n", ++a, ++b);
    return 0;
}
```

5. 下面程序段的输出是什么？请解释该现象并说出与之相关尽可能多的知识。

```
int main(int argc, char *argv[])
{
    printf("%d\n", printf("Xiyou Linux Group2%d", printf("")));
    return 0;
}
```

6. 执行下面的代码段，会输出什么？请试着解释其原因，并叙述相关知识。

```
int main(int argc, char *argv[])
{
    char ch = 255;
    int d = ch + 1;
    printf("%d %d", ch, d);
    return 0;
}
```

7. 执行以下代码段，将产生什么样的输出？请对输出加以解释，并手动计算代码中  $t$  的值。

```
int main(int argc, char *argv[])
{
    char x = -2, y = 3;
    char t = (++x) | (y++);
    printf("x = %d, y = %d, t = %d\n", x, y, t);
    t = (++x) || (y++);
    printf("x = %d, y = %d, t = %d\n", x, y, t);
    return 0;
}
```

8. 下面代码段的输出结果是什么？输出该结果的原因是？

```
#define X a + b

int main(int argc, char *argv[])
{
    int a = 1, b = 1;
    printf("%d\n", X * X);
    return 0;
}
```

9. 执行下列程序段，并输入“Xiyou Linux”（不含引号），那么程序的输出结果是什么？请解释其原因。该程序中有一处隐患，请找出并说明。

```
int main(int argc, char *argv[])
{
    char *p = (char *)malloc(sizeof(char) * 20);
    char *q = p;
    scanf("%s %s", p, q);
    printf("%s %s\n", p, q);
    return 0;
}
```

10. 执行下面的程序段，整理并解释输出结果。

```
int main(int argc, char *argv[])
{
    int a[4] = { 2, 0, 1, 9 };
    printf("%p, %p\n", a, &a);
    printf("%p, %p\n", a + 1, &a + 1);
    return 0;
}
```

11. 斐波那契数列是指这样的一串数列：第一项、第二项为 1，从第三项开始，每一项为前两项之和，该数列的第  $x$  项可以表示如下所示的函数。请根据描述，写出一个程序，使之输入  $x$  后，能够输出斐波那契数列的第  $x$  项（其中  $x \leq 30$ ）。当你完成之后，你可以尝试使用递归解决这个问题。

$$f(x) = \begin{cases} 1, & x = 1 \text{ or } x = 2 \\ f(x-1) + f(x-2), & x \geq 3 \end{cases} (x \in N^+)$$

12. 下面代码段是某一种排序算法的简单实现，你知道它是什么吗？请讲解其原理，并尝试改进它。你还知道哪些排序算法？试着详细描述这些算法。

```
void sort(int nums[], int n) {
    for (int i = 0; i < n; i++) {
        for (int j = i; j < n; j++) {
            if (nums[i] > nums[j]) {
                int c = nums[i];
                nums[i] = nums[j];
                nums[j] = c;
            }
        }
    }
}
```

13. 请简单叙述两种字节序（大端、小端）的概念，你的机器是什么字节序？试着写一个 C 语言程序来验证，如果你没有思路，你可以尝试使用联合体或者指针。

14. 以下是在某机器下执行 Linux 命令 `ls` 的部分输出（有删节），参考该输出，你可以说出哪些关于该命令以及其他 Linux 的相关知识？

```
[root@xiyoulinux /]# ls -al
total 36
drwxr-xr-x 17 root root 4096 Sep 21 23:45 .
drwxr-xr-x 17 root root 4096 Sep 21 23:45 ..
lrwxrwxrwx 1 root root 7 Aug 21 22:21 bin -> usr/bin
drwxr-xr-x 4 root root 2048 Jan 1 1970 boot
drwxr-xr-x 21 root root 3580 Nov 21 21:16 dev
drwxr-xr-x 83 root root 4096 Nov 21 22:12 etc
drwxr-xr-x 4 root root 4096 Sep 22 00:07 home
drwxr-xr-x 2 root root 4096 Aug 21 22:21 mnt
drwxr-x--- 9 root root 4096 Nov 19 19:15 root
dr-xr-xr-x 13 root root 0 Nov 21 21:15 sys
drwxrwxrwt 10 root root 380 Nov 21 22:30 tmp
drwxr-xr-x 9 root root 4096 Nov 21 22:12 usr
```