

关于指针函数的使用

首先是指针函数变量的使用和注意事项

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

int fast(void)
{
    return rand();
}

int main()
{
    int (*a)(void) = fast; /*声明一个函数指针a,int表示函数应该返回的值, void表示函数需要的值,
int, void必须——对*/

    /*(*a)表示a是一个函数指针。*/
    /*fast不需要(), 因为fast()返回的是一个int类型的值*/

    int i;
    scanf("%d",&i);
    while (i--)
    {
        printf("%d ",a());
    }
    return 0;
}
```

声明一个函数指针a,int表示函数应该返回的值, void表示函数需要的值, int, void必须——对应(*a)表示a是一个函数指针。例如声明函数指针变量int (*a)(int ,int),那么只能给他赋值形如int Funtion(int the_a,int the_b);的函数。因为优先级的原因, 所以*a必须用括号()括起来。C Primer Plus中的14.14节有更详细的解释。

关于回调函数的的使用

个人认为回调函数是在函数中需要函数指针变量, 然后在函数中调用那个函数指针。以下面举例:

```

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

int randon(int a)
{
    srand(a);
    return rand();
}

int callbackFuntion(int (*a)(int),int seed) //回调函数的一个特征就是需要一个指针函数变量
{
    int number = a(seed);
    printf("the number is %d",number);
    return number;
}

int main()
{
    int seed;
    scanf("%d",&seed);
    callbackFuntion(randon,seed);
    return 0;
}

```

int callbackFuntion(int (*a)(int),int seed) 就是一个回调函数，他需要提供提供一个形如 int Func(int a) 的函数，

关于 typedef void (*Func)(void);

```
#include <stdio.h>

typedef void (*fun)(void); //将fun作为一个类型，可以声明函数指针

void one(void)
{
    printf("one\n");
}

void two(void)
{
    printf("two\n");
}

void three(void)
{
    printf("three\n");
}

void print(int a)
{
    fun arr[] = {one,two,three}; //声明函数指针变量可以变得更简洁
    fun out = arr[a];
    out();
}

int main()
{
    print(0);
    print(1);
    print(2);
    return 0;
}
```

由示例可以知道typedef void (*Func)(void) 可以使代码变得更加简洁，提高可读性。

by 马上回来 2024/3/17