

C语言正确使用extern关键字

 blog.csdn.net/xingjiarong/article/details/47656339

利用关键字extern，可以在一个文件中引用另一个文件中定义的变量或者函数，下面就结合具体的实例，分类说明一下。

一、引用同一个文件中的变量

```
#include<stdio.h>

int func();

int main()
{
    func();
    printf("%d",num);
    return 0;
}

int num = 3;

int func()
{
    printf("%d\n",num);
}
```

如果按照这个顺序，变量 num在main函数的后边进行声明和初始化的话，那么在main函数中是不能直接引用num这个变量的，因为当编译器编译到这一句话的时候，找不到num这个变量的声明，但是在func函数中是可以正常使用，因为func对num的调用是发生在num的声明和初始化之后。

如果我不想改变num的声明的位置，但是想在main函数中直接使用num这个变量，怎么办呢？可以使用extern这个关键字。像下面这一段代码，利用extern关键字先声明一下num变量，告诉编译器num这个变量是存在的，但是不是在这之前声明的，你到别的地方找找吧，果然，这样就可以顺利通过编译啦。但是你要是想欺骗编译器也是不行的，比如你声明了extern int num；但是在后面却没有真正的给出num变量的声明，那么编译器去别的地方找了，但是没找到还是不行的。

下面的程序就是利用extern关键字，使用在后边定义的变量。

```

#include<stdio.h>

int func();

int main()
{
    func();
    extern int num;
    printf("%d",num);
    return 0;
}

int num = 3;

int func()
{
    printf("%d\n",num);
}

```

二、引用另一个文件中的变量

如果extern这个关键字就这点功能，那么这个关键字就显得多余了，因为上边的程序可以通过将num变量在main函数的上边声明，使得在main函数中也可以使用。extern这个关键字的真正的作用是引用不在同一个文件中的变量或者函数。

main.c

```

#include<stdio.h>

int main()
{
    extern int num;
    printf("%d",num);
    return 0;
}

```

b.c

```

#include<stdio.h>

int num = 5;

void func()
{
    printf("fun in a.c");
}

```

例如，这里b.c中定义了一个变量num，如果main.c中想要引用这个变量，那么可以使用extern这个关键字，注意这里能成功引用的原因是，num这个关键字在b.c中是一个全局变量，也就是说只有当一个变量是一个全局变量时，extern变量才会起作用，向下面这样是不行的。

mian.c

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

```
int main()
{
    extern int num;
    printf("%d",num);
    return 0;
}
```

b.c

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

```
void func()
{
    int num = 5;
    printf("fun in a.c");
}
```

另外，extern关键字只需要指明类型和变量名就行了，不能再重新赋值，初始化需要在原文件所在处进行，如果不进行初始化的话，全局变量会被编译器自动初始化为0。像这种写法是不行的。

```
extern int num=4;
    1
```

但是在声明之后就可以使用变量名进行修改了，像这样：

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

```
int main()
{
    extern int num;
    num=1;
    printf("%d",num);
    return 0;
}
```

如果不想这个变量被修改可以使用const关键字进行修饰，写法如下：

mian.c

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

```
int main()
{
    extern const int num;
    printf("%d",num);
    return 0;
}
```

b.c

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

```
const int num=5;
void func()
{
    printf("fun in a.c");
}
```

使用include将另一个文件全部包含进去可以引用另一个文件中的变量，但是这样做的结果就是，被包含的文件中的所有的变量和方法都可以被这个文件使用，这样就变得不安全，如果只是希望一个文件使用另一个文件中的某个变量还是使用extern关键字更好。

三、引用另一个文件中的函数

extern除了引用另一个文件中的变量外，还可以引用另一个文件中的函数，引用方法和引用变量相似。

mian.c

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

```
int main()
{
    extern void func();
    func();
    return 0;
}
```

b.c

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

```
const int num=5;
void func()
{
    printf("fun in a.c");
}
```

这里main函数中引用了b.c中的函数func。因为所有的函数都是全局的，所以对函数的extern用法

和对全局变量的修饰基本相同，需要注意的就是，需要指明返回值的类型和参数。