C语言使用中疑难问题的解决

中国矿业大学 胡继普

C语言具有语言简洁，使用灵活方便，易于维护、易于移植，可读性强及数据结构丰富等特点，它不但广泛应用于编制系统软件，而且，随着微机上C语言的普及，更多的人在使用C语言开发各种应用软件。由于C语言本身语法限制不严，在使用中，很多不合理的语句编译器不能发现，这就增加了程序调试的难度，对于初学者更是如此。本文是几年来使用C语言的经验总结，希望能起到抛砖引玉的作用。

一、字符串使用方面的问题

无论开发研制何种软件，字符串的使用都占据非常重要的地位，正确地使用字符串就显得特别具有现实意义。

1.字符串赋初值问题

```
main( ){char str1[8 ], str2[8 ];
        str1[ ]="abcd"; str2="efg ";
        printf("%s %s ", str1, str2); }
```

编译时出现如下错误：

```
illegal lhs of assignment operator。
```

无论 str1，还是 str2，这二种赋值都不成立，正确的赋值方法如下：

(1)将字符数组定义成外部字符数组，并同时赋以初值，示意如下：

```
char str [8 ]={"胡继普"};
main( ){……}
```

本例中，字符串两边的大括号可以省去，但请注意，如下使用仍然不对。

```
char str [8 ];
main( ){…str ="工程师"; …}
```

如果需要字符串赋值也只能采取下标循环的方式，如：

```
char str [10]="计算中心";
main(){char sttr2 [10]; int i;
  for ci=0 ; i < 10; i ++)str2[i ]= str [i ];
  printf("%s %s ", str , str2); }
```

(2)将字符数组定义成静态数组

```
static ehar strr[ ]={"中国"};
```

(3)使用字符指针变量

```
char * str ="胡继普";
```

或者在使用时赋初值，如下例：

```
char * str ;
main( ) {…str ="工程师+; …}
```

(4)使用字符指针数组

```
main ( ) {
static char * str [ ]={"ab","c ","d "};
printf(%s %s %s , "str [0 ], str [1 ], str [2 ]; }
```

2.多维数组使用中的问题

(1)定义一个二维字符数组，如10个字符串，每个串长15个字符，应定义成：

```
char str [10][15];
```

而不能定义成：char str [15][10];

(2)缺省下标的数组定义。正确定义字意如下：

```
char str [ ], str1[ ][3 ], str2[ ][4 ][5 ];
```

下面的定义都会产生错误：

```
char str [ ][ ], str1[ ][ ][4 ];
```

因为多维数组的定义只能是最左边的一维的下标可以缺省，同理其它多维数组的定义也必须遵守此项规定。

3.字符数组如何清零

在很多应用软件的开发中，经常有字符数组的重复使用问题，如果在第二次用之前不清零会带来错误，因为字符串是以'/0'为结束标志的，则只需将字符数组最后一维的首字符赋为'/0'即可，如下都为正确的清零表达式。

```
str [0 ] = ‘ / 0 ’; str [i ][o ]= ‘ / o ’;
str [i ][j ][o ]= ‘ / o ’;
```

二、scanf 函数使用方面的问题

1. Memory fault － Core dumped

这是由于 scanf (“%d ”, &i); 即为正确。

2.无论输入何值，打印输出皆为零

这是由于: scanf (“c = %d ”, &i); 造成的。这种使用试图在输入时，给出提示信息，实际上，格式转换符“%”之前只可以出现空格、制表及换行符，其它符号皆为非法，这种使用方法如果取地址运算符“&”漏写，也不会出现上例错误，编译执行皆通过，只是无论输入为何值打印输出皆为零值。

3.输入为一句话，输出为一个词

请看下例:

```
main( ) {char str [15];
scanf (“%s ”, str );
printf(“%s ”, str ); }
```

如这时的输入为: How do you do ?

输出只一个词: How

这是因为 scanf 函数输入多个字符串时，其分隔符为空格，，How 其后的字符皆舍去，若想完整打印这句话，可以如下使用:

```
main( ){char strr1 [ ], str2[ ], str3[ ], str4[ ];
scanf (“%s %s %s %s ”, str1, str2, str3, str4;
printf(“%s %s %s %s ”, str1, str2, str3, str4);
```

同理如: scanf (“%d %d ”, &i , &j); 这时输出时，两个整数应由空格分隔，但如果这样使用: scannf(“%d , %d ”, &i , &j); 则输入时，两个整数以逗号“,”分隔，如不按这要求输入，则会出现错误。

4.scanf (%s ”, &str)

如这时 str 定义为字符数组则出错。因为字符数组名本身就是代表着地址的指针，而加上“&”则为多余，故出错。

5.指针未赋值，即用 scanf 输入。如下:

```
main( ){intt * c ; scanf (“%d ”, c ); }
```

因为 c 不确定，使用中会出现不可预料的错误，正确使用应为:

```
main( ) {int * c , ab[4 ];
c = ab; scanf (“%d ”, c ); }
```

这样使用就不会出错，因为 C 具有的确的值，它为整数组 ab 的起始地址。

三、预处理出现的问题

编译时，出错信息如下:

```
1 : illegal charaeter : 043 (octal )
1 : cannot recoverr from earlierr erors : goodby!
```

这是由于#include 或#define 没有顶头写造成的，将#号前的空格删去，则错误消失。

四、Buss error __ Core dumped

请看下例:

```
main( ) {char ln[4 ];
sprintf (ln, “abcefgh …xyz ”); }
```

这里欲向 ln 数组中放的字符太多，以致造成该错，解决办法: 定义是够长的字符数组即可。因为 C 语言的编译器对数组的下标不作检查，因此，在使用数组时，一定要注意下标的合理取值范围，否则会出现不可意料的结果。

五、Memory fault __ Core duwped

欲读一个不存在的文件，或欲向一个以只读方式打开的文件中写数据。

```
fp = fopen (“sf”, “r ”);
fscannf (fp, “%d ”, &i );
```

如果 sf 文件不存在则出上述错误。

加入出错处理，即可避免这类错误的产生

```
if (fp = fopnen(“sf”, “rr); )
fscanf (fp, “%d ”, &i );
else printf(“文件不存在，退出”);
```

同时，请注意，在这种方式下，打开文件的方式的字符“r ”和“w ”只能用小写，用大写则会出错。

六、当你完全发现不了错误时

程序员在编写调试程序时，最头痛的就是这种问题，有时很简单的一点小问题也会浪费很长时间，下面分别予以说明:

1.编译指出某行有错，而这一行看上去完全不存在

错误。

建议通过如下方法，予以排除故障。

(1)将该行重新输入一遍，因为有时程序录入时，会混入不可见字符。

(2)检查一下前方的大括号是否配对。

(3)静态变量赋初值时，语句最后是否遗忘分号“;”。

(4)注释语句是否有重迭，如：

/ * …… / * …… * / …… *

/不合法，错应该该为：/ * …… * / / * …… * /

(5)预处理后是否多分号“;”。

(6)前方语句是否遗失分号“;”。

2.编译运行皆通过，只是结果不正确

(1)将赋值运算符“=”当成关系运算符的等于，即恒等于运算符，“==”。如：

```
for (i = o ; ; itt ){…… if (i = 6 )brdak ; ……}
```

本段的原意是，当 i = 6 时，跳出循环，但因为关系运算符使用不对，实际上为赋值运算符，i = 6 大于 o，恒为真，所以只执行一次就跳出循环体。

同理，while (c = ‘q ’){……break ; }

因为赋给 c 的是一个大于 o 的数，恒为真，所以，并不是 c = ‘q ’时才跳出，而是直接跳出。

(2)运算符优先级校不清造成的错误。

这类错误，通过适当的增加括号即可以避免这类错误的发生。

(3)在 if，for，while 等语句后，多加分号“;”，相当于执行空语句，而不能正常完成预期工作，应防止如下示意的错误发生

```
if(……); {……}
for (; ; ); {……}
while (o < n ); {……}
```

(4)将分隔符逗号“,”，误敲成句号“。”。

(5)该用复合语句的地方，忘记了大括号。

(6)输入、输出的数据类型与所有的格式说明不一致，如：

```
int a ; float b ;
scanf (“%f %d ”, &a , &b );
printf (“%f %d ”, &a , &b );
```

(7)整型变量(int)超出取值范围、改成长整型即可，，即 int 改成 long。

(8)表达式存在二义性。如：

```
mx[i ]+++i ; mx[c ]= ++i ;
```

3.其它可能出现错误的情况

(1)do…while 语句结尾处少分号“;”

```
do {……}while (i < n )
```

在 i < n 括号后加上分号才为正确。

(2)在函数名及“{”之间定义非函数参数变量，如：

```
main () int i ; {……}或
main (x , y )int x , y , z ; {……}
```

这二例中，int i 和 int z 都是非法的。

(3)关键字敲错，出错提示为变量未定义。

(4)定义数组使用小括号，如 innt abc ();

(5)使用的小括号不配对。

(6)有些无法解释的错误，改变一下变量名试试，有时，变量名同系统保留字成数据库字段名重名，会出现无法解释的错误。

(7)试图用一个括号定义多维数组，如：

```
int i [5 , 6 ], j [5 , 6 , 7 ]。
```

▲