命令行参数解析函数 —— getopt()

getopt()函数声明如下：

|  |
| --- |
| #include <unistd.h>  int getopt(int argc, char \* const argv[], const char \*optstring);  extern char \*optarg; extern int optind, opterr, optopt; |

该函数的argc和argv参数通常直接从main()的参数直接传递而来。optstring是选项字母组成的字串。如果该字串里的任一字符后面有冒号，那么这个选项就要求有选项参数。

当给定getopt()命令参数的数量 (argc)、指向这些参数的数组 (argv) 和选项字串 (optstring) 后，getopt() 将返回第一个选项，并设置一些全局变量。使用相同的参数再次调用该函数时，它将返回下一个选项，并设置相应的全局变量。如果不再有可识别的选项，将返回 -1，此任务就完成了。

getopt() 所设置的全局变量包括：

* char \*optarg——当前选项参数字串（如果有）。
* int optind——argv的当前索引值。当getopt()在while循环中使用时，循环结束后，剩下的字串视为操作数，在argv[optind]至argv[argc-1]中可以找到。
* int opterr——这个变量非零时，getopt()函数为“无效选项”和“缺少参数选项，并输出其错误信息。
* int optopt——当发现无效选项字符之时，getopt()函数或返回'?'字符，或返回':'字符，并且optopt包含了所发现的无效选项字符。

以下面的程序为例：  
选项：

* -n —— 显示“我的名字”。
* -g —— 显示“我女朋友的名字”。
* -l —— 带参数的选项.

**清单2：**

|  |
| --- |
| #include <stdio.h> #include <unistd.h>  int main (int argc, char \*\*argv) {     int oc;                     /\*选项字符 \*/     char \*b\_opt\_arg;            /\*选项参数字串 \*/      while((oc = getopt(argc, argv, "ngl:")) != -1)     {         switch(oc)         {             case 'n':                 printf("My name is Lyong.\n");                 break;             case 'g':                 printf("Her name is Xxiong.\n");                 break;             case 'l':                 b\_opt\_arg = optarg;                 printf("Our love is %s\n", optarg);                 break;         }     }    return 0; } |

运行结果：

|  |
| --- |
| $ ./opt\_parse\_demo -n My name is Lyong. $ ./opt\_parse\_demo -g Her name is Xxiong. $ ./opt\_parse\_demo -l forever Our love is forever $ ./opt\_parse\_demo -ngl forever My name is Lyong. Her name is Xxiong. Our love is forever |

**6、改变getopt()对错误命令行参数信息的输出行为**

不正确的调用程序在所难免，这种错误要么是命令行选项无效，要么是缺少选项参数。正常情况下，getopt()会为这两种情况输出自己的出错信息，并且返回'?'。为了验证此事，可以修改一下上面的清单2中的代码。

**清单3：**

|  |
| --- |
| #include <stdio.h> #include <unistd.h>  int main (int argc, char \*\*argv) {     int oc;                     /\*选项字符 \*/     char \*b\_opt\_arg;            /\*选项参数字串 \*/      while((oc = getopt(argc, argv, "ngl:")) != -1)     {         switch(oc)         {             case 'n':                 printf("My name is Lyong.\n");                 break;              case 'g':                 printf("Her name is Xxiong.\n");                 break;             case 'l':                 b\_opt\_arg = optarg;                 printf("Our love is %s\n", optarg);                 break;             case '?':                 printf("arguments error!\n");                 break;         }     }     return 0; } |

输入一个错误的命令行，结果如下：

|  |
| --- |
| $ ./opt\_parse\_demo -l ./opt\_parse\_demo: option requires an argument -- l arguments error! |

如果不希望输出任何错误信息，或更希望输出自定义的错误信息。可以采用以下两种方法来更改getopt()函数的出错信息输出行为：

1. 在调用getopt()之前，将opterr设置为0，这样就可以在getopt()函数发现错误的时候强制它不输出任何消息。
2. 如果optstring参数的第一个字符是冒号，那么getopt()函数就会保持沉默，并根据错误情况返回不同字符，如下：
   * “无效选项” —— getopt()返回'?'，并且optopt包含了无效选项字符（这是正常的行为）。
   * “缺少选项参数” —— getopt()返回':'，如果optstring的第一个字符不是冒号，那么getopt()返回'?'，这会使得这种情况不能与无效选项的情况区分开。

**清单4：**

|  |
| --- |
| #include <stdio.h> #include <unistd.h>  int main (int argc, char \*\*argv) {     int oc;                     /\*选项字符 \*/     char ec;                             /\*无效的选项字符\*/     char \*b\_opt\_arg;            /\*选项参数字串 \*/      while((oc = getopt(argc, argv, ":ngl:")) != -1)     {         switch(oc)         {             case 'n':                 printf("My name is Lyong.\n");                 break;              case 'g':                 printf("Her name is Xxiong.\n");                 break;             case 'l':                 b\_opt\_arg = optarg;                 printf("Our love is %s\n", optarg);                 break;             case '?':                 ec = (char)optopt;                 printf("无效的选项字符 \' %c \'!\n", ec);                 break;             case ':':                 printf("缺少选项参数！\n");                 break;         }     }     return 0; } |

测试结果：

$ ./opt\_parse\_demo -a  
无效的选项字符 ' a '!  
$ ./opt\_parse\_demo -l  
缺少选项参数！