



查看: 4044 | 回复: 5

wuwenhua

白手起家
帖子 41
主题 11
精华 0
可用积分 68
专家积分 0
在线时间 133 小时
注册时间 2010-01-30
最后登录 2011-04-30
问答 好友
博客 消息
论坛徽章: 0

getopt和getopt_long函数使用介绍 [复制链接]

0 0 打印 分享

发表于 2010-11-20 20:03:22 | 只看该作者 | 倒序浏览 [报告] [收藏(0)] 1楼 电梯直达

Linux选项解释-getopt和getopt_long函数

一、命令行简介

解释分析命令行通常是所以程序的第一个任务，C语言通过argc和argv参数来访问它的命令行参数。

最简单的命令行处理技术可以通过if判断来表示，如下例：

if(argc>1 &&argv[1][0] == ‘ - ‘ &&argv[1][1] == ‘ h ’) //判断命令行参数是否为-n { do _ some thing(); }

这样处理简单有序的命令行还可以，对于复杂的命令行处理显得有心无力，于是GNU提供两个函数专门用来处理命令行参数：getopt和getopt_long。

二、getopt函数

getopt()函数声明如下：

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

说明：函数中的argc和argv通常直接从main()到两个参数传递而来。optsting是选项参数组成的字符串，如果该字符串里任一字母后有冒号，那么这个选项就要求有参数，optarg就是选项参数。optind是当前索引，optopt用于当发现无效选项字符的时候，getopt函数或者返回“？”或者返回“：”字符，并且optopt包含了所发现的无效选项字符。

如果optstring参数的第一个字符是冒号，那么getopt会根据错误情况返回不同的字符，当错误是无效选项，getopt返回“？”，当错误是缺少选项参数，getopt返回“：”。

注：GNU getopt()第三个特点是optstring中的选项字符后面接两个冒号，就允许该选项有可选的选项参数。在选项参数不存在的情况下，GNU getopt()返回选项字符并将optarg设置为NULL。

例子：

C代码#include <stdio.h>

```
#include <unistd.h>

#include <getopt.h>

char *para = ":ab:c";

int main(int argc, char *argv[])

{

    int oc = -1;

    t = NULL;

    getopt(argc, argv, para)) != -1)

        x(oc)

    {

        case 'a':

            printf("input para is a\n");

            break;

        case 'b':

            b_input = optarg;

            printf("input para is b,and optarg is %s\n", b_input);

            break;

        case 'c':

            printf("input para is c\n");

            break;

        case ':':

            printf("option %c requires an argument\n",optopt);

            break;

        case '?':

        default:

            printf("option %c is invalid:ignored\n",optopt);

            break;

    }

}

return 0;

}

#include <stdio.h>

#include <unistd.h>

#include <getopt.h>
```

perl 有没有类似C语言getopt_long(...

getopt和getopt_long函数使用介绍

Linux选项解释-getopt和getopt_l...

Linux下C语言的各种信息获取

使用 getopt()进行命令行处理__轻松处理

+ 更多"C语言getopt_long"相关帖子

```
char *para = ":ab:c";

int main(int argc, char *argv[])

{

    int oc = -1;

    char *b_input = NULL;

    while((oc = getopt(argc, argv, para)) != -1)

    {

        switch(oc)

        {

            case 'a':

                printf("input para is a\n"😊);

                break;

            case 'b':

                b_input = optarg;

                printf("input para is b,and optarg is %s\n", b_input);

                break;

            case 'c':

                printf("input para is c\n"😊);

                break;

            case ':':

                printf("option %c requires an argument\n",optopt);

                break;

            case '?':

            default:

                printf("option %c is invalid:ignored\n",optopt);

                break;

        }

    }

    return 0;

}
```

编译:

```
[root@heguangwu projects]# gcc -o getopt_ex getopt_ex.c
```

运行:

```
[root@heguangwu projects]# ./getopt_ex -a
```

```
input para is a
```

```
[root@heguangwu projects]# ./getopt_ex -a -b
```

input para is a

option b requires an argument

```
[root@heguangwu projects]# ./getopt_ex -d
```

option d is invalid:ignored

三、 getopt_long函数

getopt_long用来处理长选项，使用 man 3 getopt_long，得到其声明如下：

C代码

```
#include <getopt.h>
```

```
int getopt_long(int argc, char * const argv[], const char *optstring,
```

```
const struct option *longopts, int *longindex);
```

```
int getopt_long_only(int argc, char * const argv[], const char *optstring,
```

```
const struct option *longopts, int *longindex);
```

```
#include <getopt.h>
```

```
int getopt_long(int argc, char * const argv[], const char *optstring,
```

```
const struct option *longopts, int *longindex);
```

```
int getopt_long_only(int argc, char * const argv[], const char *optstring,
```

```
const struct option *longopts, int *longindex);
```

前三个参数与getopt相同，下一个参数是指向数组的指针，这个数组是option结构数组，option结构称为长选项表，其声明如下：

C代码

```
struct option
```

```
{
```

```
    const char *name;
```

```
    int has_arg;
```

```
    int *flag;
```

```
    int val;
```

```
};
```

```
struct option
```

```
{
```

```
    const char *name;
```

```
    int has_arg;
```

```
    int *flag;
```

```
    int val;
```

```
};
```

结构中的元素解释如下：

const char *name：选项名，前面没有短横线

int has_arg：描述长选项是否有参数，其值见下表

符号常量

数值

含义

no_argument

required_argument

optional_argument

0

1

2

选项没有参数

选项需要参数

选项参数是可选的

int *flag：

如果该指针为NULL，那么getopt_long返回val字段的值；

如果该指针不为NULL，那么会使得它所指向的结构填入val字段的值，同时getopt_long返回0

int val：

如果flag是NULL，那么val通常是个字符常量，如果短选项和长选项一致，那么该字符就应该与optstring中出现的这个选项的参数相同；

C代码

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

```
#include <unistd.h>
```

```
#include <getopt.h>
```

```
char *para = ":ab:cf:v";
```

```
int do_all = 0;
```

```
int do_help = 0;
```

```
int do_version = 0;
```

```
char *file = NULL;
```

```
struct option longopt[] =
```

```
{
```

```
    {"all", no_argument, &do_all, 1},
```

```
    {"file", required_argument, NULL, 'f'},
```

```
    {"help", no_argument, &do_help, 1},
```

```
    {"version", no_argument, &do_version, 1},
```

```
    {"bob", required_argument, NULL, 'b'},
```

```
{0, 0, 0, 0},

};

int main(int argc, char *argv[])

{

    int oc = -1;

    char *b_input = NULL;

    while((oc = getopt_long(argc, argv, para, longopt, NULL)) != -1)

    {

        switch(oc)

        {

            case 'a':

                printf("input para is a\n"😊);

                break;

            case 'b':

                b_input = optarg;

                printf("input para is b,and optarg is %s\n", b_input);

                break;

            case 'c':

                printf("input para is c\n"😊);

                break;

            case 'v':

                printf("input para is v\n"😊);

                break;

            case 'f':

                printf("input para is f\n"😊);

                file = "hello world";

                break;

            case 0:

                break;

            case ':':

                printf("option %c requires an argument\n",optopt);

                break;

            case '?':

            default:

                printf("option %c is invalid:ignored\n",optopt);
```

```
        break;

    }

}

printf("do_all is %d\n",do_all);

printf("do_help is %d\n",do_help);

printf("do_version is %d\n",do_version);

printf("do_file is %s\n",file);

printf("bob is %s\n", b_input);

return 0;

}

#include <stdio.h>

#include <unistd.h>

#include <getopt.h>

char *para = ":ab:cf:v";

int do_all = 0;

int do_help = 0;

int do_version = 0;

char *file = NULL;

struct option longopt[] =

{

    {"all", no_argument, &do_all, 1},

    {"file", required_argument, NULL, 'f'},

    {"help", no_argument, &do_help, 1},

    {"version", no_argument, &do_version, 1},

    {"bob", required_argument, NULL, 'b'},

    {0, 0, 0, 0},

};

int main(int argc, char *argv[])

{

    int oc = -1;

    char *b_input = NULL;

    while((oc = getopt_long(argc, argv, para, longopt, NULL)) != -1)

    {

        switch(oc)

        {

            case 'a':
```

```
        printf("input para is a\n");  
  
        break;  
  
    case 'b':  
  
        b_input = optarg;  
  
        printf("input para is b,and optarg is %s\n", b_input);  
  
        break;  
  
    case 'c':  
  
        printf("input para is c\n");  
  
        break;  
  
    case 'v':  
  
        printf("input para is v\n");  
  
        break;  
  
    case 'f':  
  
        printf("input para is f\n");  
  
        file = "hello world";  
  
        break;  
  
    case 0:  
  
        break;  
  
    case ':':  
  
        printf("option %c requires an argument\n",optopt);  
  
        break;  
  
    case '?':  
  
    default:  
  
        printf("option %c is invalid:ignored\n",optopt);  
  
        break;  
  
    }  
}  
  
printf("do_all is %d\n",do_all);  
  
printf("do_help is %d\n",do_help);  
  
printf("do_version is %d\n",do_version);  
  
printf("do_file is %s\n",file);  
  
printf("bob is %s\n", b_input);  
  
return 0;  
}
```



```
执行的结果： 只显示关键结果

[root@heguangwu projects]# ./opt_ex2 -a

input para is a


[root@heguangwu projects]# ./opt_ex2 --all

do_all is 1


[root@heguangwu projects]# ./opt_ex2 -f h

input para is f

do_file is hello world


[root@heguangwu projects]# ./opt_ex2 --bob aa

input para is b,and optarg is aa

bob is aa


[root@heguangwu projects]# ./opt_ex2 -b aa

input para is b,and optarg is aa
```

本版精华

热门专题

文库|博客

关于redhat 9.0 打开进程打开最大文件数
fluke
DirectFB 之一：编译与安装(PC)
Linux下如何Enable阵列卡(Raid)的Write Back...
java程序的发布，JRE——请教高手

YUM认识
GDB调试多文件问题
freebsd下的ADSL上网设置
如何查看本机开放端口
建Oracle RAC数据库时报错ora27041

初学UNIX环境高级编程的，关于cat的疑问 | chinaunix博客什么时候可以设置背景啊，感觉 ... | 一个访问量较大网站的服务器。配置那些文件 ... | 收音机驱动与v4l2架构的关系，如何交互以及 ... |

PCliangtao



稍有积蓄

☆

帖子 206

主题 26

精华 1

可用积分 418

专家积分 0

在线时间 151 小时

注册时间 2009-09-21

最后登录 2012-07-26

问答 好友

博客 消息

论坛徽章: 0

发表于 2010-11-20 21:50:08 | 只看该作者

[报告] 2楼

内核源码版块怎么出现这个东西...

企业级Linux应用痛点难点剖析|国产桌面虚拟化产品发展现状|Mac OS X和iOS内核初探有奖交流|开发实战之Python与我不得不说的故事

wuwenhua

发表于 2010-11-21 08:42:01 | 只看该作者

[报告] 3楼

<div></div> <div>白手起家</div> <div><div>帖子</div><div>41</div></div> <div><div>主题</div><div>11</div></div> <div><div>精华</div><div>0</div></div> <div><div>可用积分</div><div>68</div></div> <div><div>专家积分</div><div>0</div></div> <div><div>在线时间</div><div>133 小时</div></div> <div><div>注册时间</div><div>2010-01-30</div></div> <div><div>最后登录</div><div>2011-04-30</div></div> <div><div>问答</div><div>好友</div></div> <div><div>博客</div><div>消息</div></div> <div>论坛徽章: 0</div>	<div>哦 我只想说明一下 netfilter防火墙中，能用到这个函数，实现用户输入命令的解析</div> <div>如果不是这 呵呵 那就发错了</div> <div>企业级Linux应用痛点难点剖析 国产桌面虚拟化产品发展现状 Mac OS X和iOS内核初探有奖交流 开发实战之Python与我不得不说的故事</div>
<div>Godbach</div> <div>To be 千里马！</div> <div></div> <div>版主</div> <div><div>帖子</div><div>15418</div></div> <div><div>主题</div><div>371</div></div> <div><div>精华</div><div>21</div></div> <div><div>可用积分</div><div>3969</div></div> <div><div>专家积分</div><div>79</div></div> <div><div>在线时间</div><div>7397 小时</div></div> <div><div>注册时间</div><div>2007-03-09</div></div> <div><div>最后登录</div><div>2014-05-31</div></div> <div><div>问答</div><div>好友</div></div> <div><div>博客</div><div>消息</div></div> <div>论坛徽章: 4</div> <div></div>	<div><div> 发表于 2010-11-21 10:54:28 只看该作者</div><div>[报告] 4楼</div></div> <div>哦 我只想说明一下 netfilter防火墙中，能用到这个函数，实现用户输入命令的解析</div> <div>如果不是这 呵呵 那就发错 ...</div> <div>wuwenhua 发表于 2010-11-21 08:42</div> <div>LZ 说的是 iptables 命令行吧。</div> <div>如果能偶结合 iptables 来分析就更好了。</div> <div>欢迎光临Godbach的博客： HAProxy 研究笔记</div> <div>明犯我强汉天威者，穷搜天下，万里追杀，覆其巢，断其苗裔，戮其身，追其魂，屠其魄，虽远必诛！</div> <div>企业级Linux应用痛点难点剖析 国产桌面虚拟化产品发展现状 Mac OS X和iOS内核初探有奖交流 开发实战之Python与我不得不说的故事</div>
<div>Godbach</div> <div>To be 千里马！</div> <div></div> <div>版主</div> <div><div>帖子</div><div>15418</div></div> <div><div>主题</div><div>371</div></div> <div><div>精华</div><div>21</div></div> <div><div>可用积分</div><div>3969</div></div> <div><div>专家积分</div><div>79</div></div> <div><div>在线时间</div><div>7397 小时</div></div> <div><div>注册时间</div><div>2007-03-09</div></div> <div><div>最后登录</div><div>2014-05-31</div></div> <div><div>问答</div><div>好友</div></div> <div><div>博客</div><div>消息</div></div> <div>论坛徽章: 4</div> <div></div>	<div><div> 发表于 2010-11-21 11:01:55 只看该作者</div><div>[报告] 5楼</div></div> <div>内核源码版块怎么出现这个东西...</div> <div>内核版也欢迎讨论一些和内核有关系比较直接的应用代码的交流。比如 iptables 之类的源码等等。</div> <div>欢迎光临Godbach的博客： HAProxy 研究笔记</div> <div>明犯我强汉天威者，穷搜天下，万里追杀，覆其巢，断其苗裔，戮其身，追其魂，屠其魄，虽远必诛！</div> <div>企业级Linux应用痛点难点剖析 国产桌面虚拟化产品发展现状 Mac OS X和iOS内核初探有奖交流 开发实战之Python与我不得不说的故事</div>
<div>L_kernel</div>	<div><div> 发表于 2010-11-21 19:50:51 只看该作者</div><div>[报告] 6楼</div></div> <div>○~○b汗，我还以为是shell脚本中的那个getopts</div>



巨富豪门
😄😄
帖子 3680
主题 101
精华 0
可用积分 26721
专家积分 0
在线时间 949 小时
注册时间 2010-04-05
最后登录 2014-03-05
[问答](#) [好友](#)
[博客](#) [消息](#)
论坛徽章： 0

=====GNU/Linux FreeBSD=====

GNU's not Unix, but Unix is a beast; its plural form is Unixen.

Freedom. Friends. Features. First.

[企业级Linux应用痛点难点剖析](#) | [国产桌面虚拟化产品发展现状](#) | [Mac OS X和iOS内核初探有奖交流](#) | [开发实战之Python与我不得不说的故事](#)

[发表主题](#)

返回列表

论坛 程序设计 内核源码 [getopt和getopt_long函数使用介绍](#)

高级模式

您需要登录后才可以回帖 登录 | 注册  [用QQ帐号登录](#)

[发表回复](#) ☐ 回帖后跳转到最后一页