Linux 下类 Shell 的实现

用户手册

计算机科学与技术系 191220008 陈南曈

目录：
一、简要介绍 --

二、cp --

三、cmp ---

四、wc --

五、cat --

六、man ---

七、其它 ---

一、简要介绍

程序名为“ MyShell ”。该程序实现了在 Linux 下的类 Shell，主要功能有 cp（复制文件或目录）、cmp（比较文件）、wc（统计文件数据）、cat（显示内容）、man（说明文档）等。

二、cp

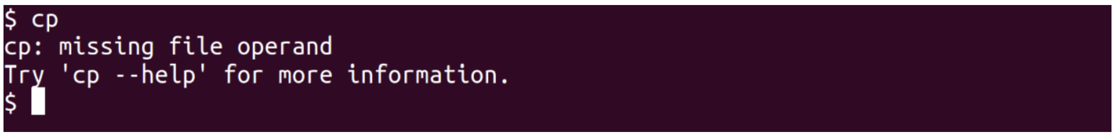
1、cp

格式：cp file1 file2

功能：将文件 file1 复制成文件 file2

失败示例：

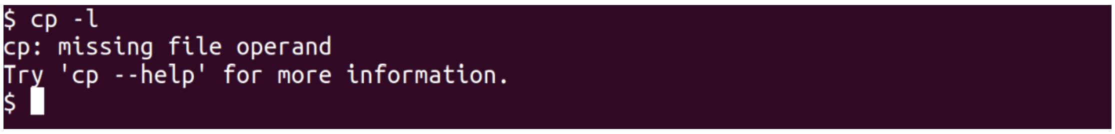
（1） 无参数：



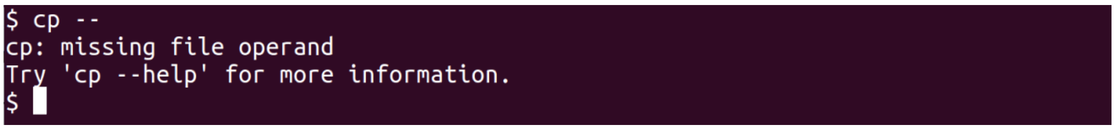
```
$ cp
cp: missing file operand
Try 'cp --help' for more information.
$
```

（2） 一个参数：

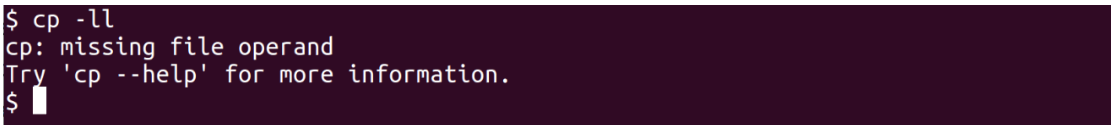
无效参数：



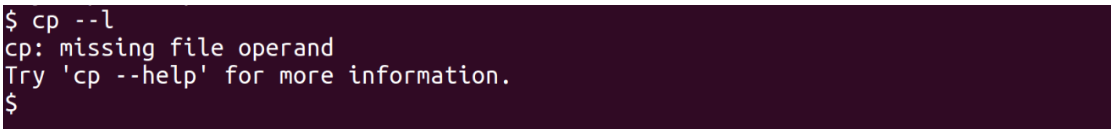
```
$ cp -l
cp: missing file operand
Try 'cp --help' for more information.
$
```



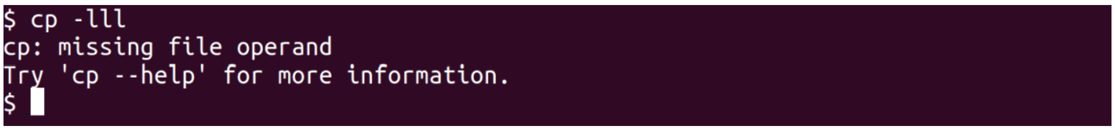
```
$ cp --
cp: missing file operand
Try 'cp --help' for more information.
$
```



```
$ cp -ll
cp: missing file operand
Try 'cp --help' for more information.
$
```

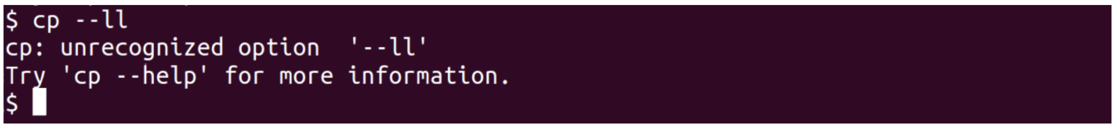


```
$ cp --l
cp: missing file operand
Try 'cp --help' for more information.
$
```

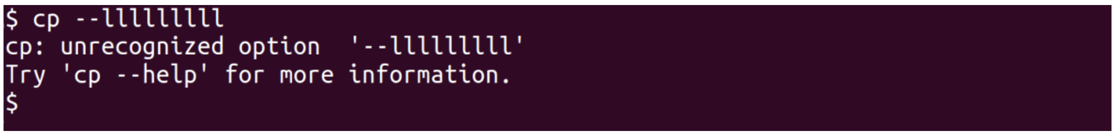


```
$ cp -lll
cp: missing file operand
Try 'cp --help' for more information.
$
```

无法识别的命令：

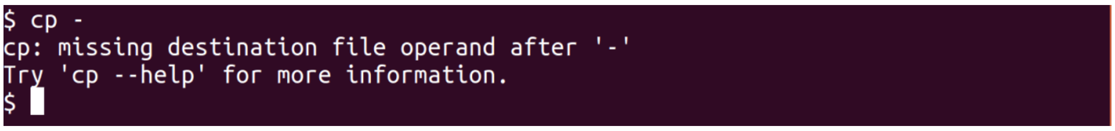


```
$ cp --ll
cp: unrecognized option  '--ll'
Try 'cp --help' for more information.
$
```

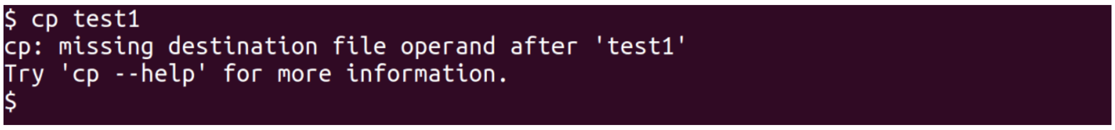


```
$ cp --lllllllll
cp: unrecognized option  '--lllllllll'
Try 'cp --help' for more information.
$
```

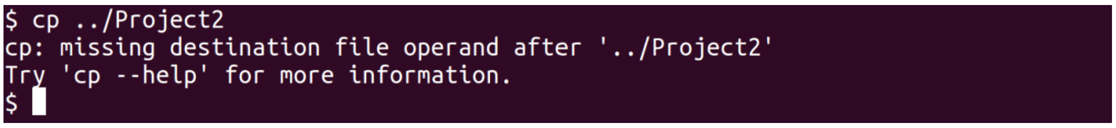
有效参数:



```
$ cp -
cp: missing destination file operand after '-'
Try 'cp --help' for more information.
$
```



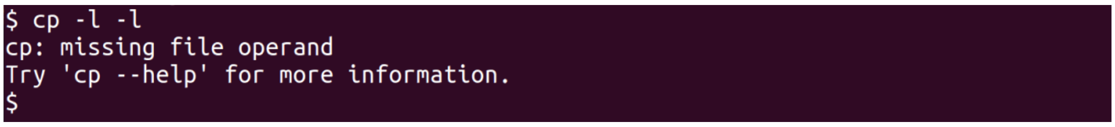
```
$ cp test1
cp: missing destination file operand after 'test1'
Try 'cp --help' for more information.
$
```



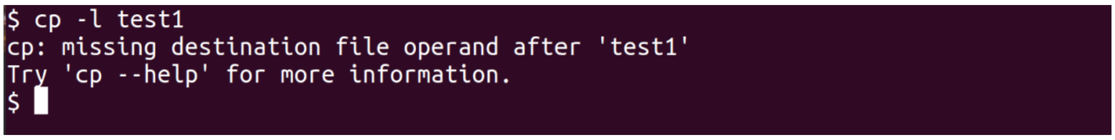
```
$ cp ../Project2
cp: missing destination file operand after '../Project2'
Try 'cp --help' for more information.
$
```

（3）两个参数:

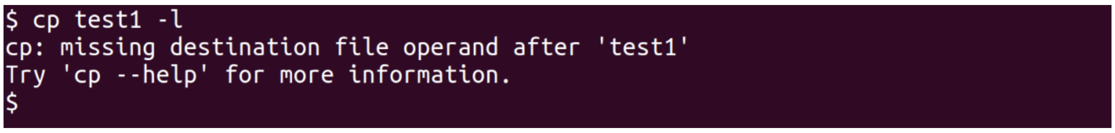
含有无效参数或无法识别的命令（分别以 “-l” 和 “--ll” 为例）:



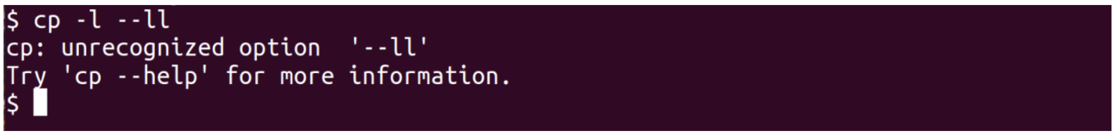
```
$ cp -l -l
cp: missing file operand
Try 'cp --help' for more information.
$
```



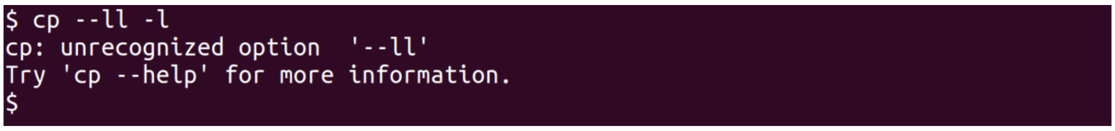
```
$ cp -l test1
cp: missing destination file operand after 'test1'
Try 'cp --help' for more information.
$
```



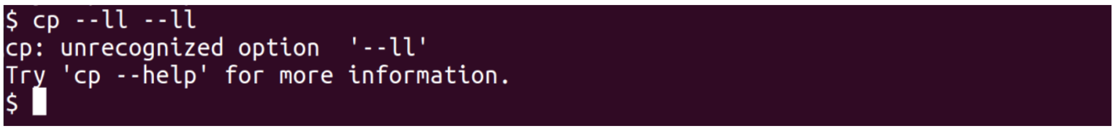
```
$ cp test1 -l
cp: missing destination file operand after 'test1'
Try 'cp --help' for more information.
$
```



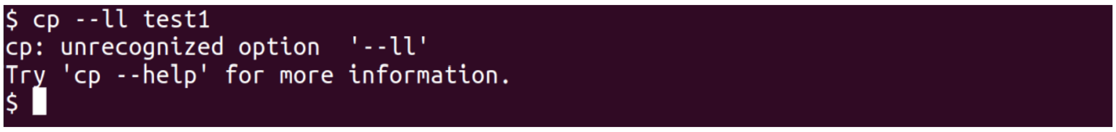
```
$ cp -l --ll
cp: unrecognized option  '--ll'
Try 'cp --help' for more information.
$
```



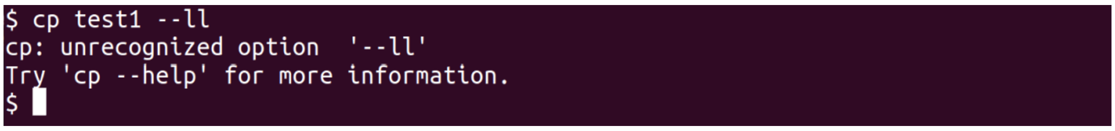
```
$ cp --ll -l
cp: unrecognized option  '--ll'
Try 'cp --help' for more information.
$
```



```
$ cp --ll --ll
cp: unrecognized option  '--ll'
Try 'cp --help' for more information.
$ █
```



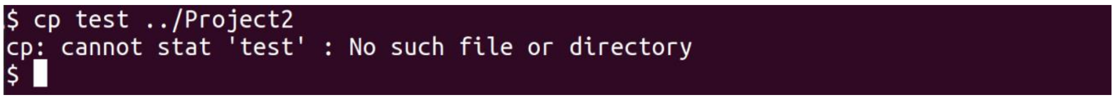
```
$ cp --ll test1
cp: unrecognized option  '--ll'
Try 'cp --help' for more information.
$ █
```



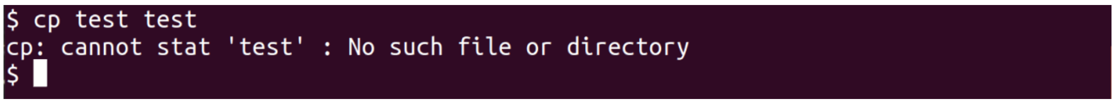
```
$ cp test1 --ll
cp: unrecognized option  '--ll'
Try 'cp --help' for more information.
$ █
```

只含有有效参数:

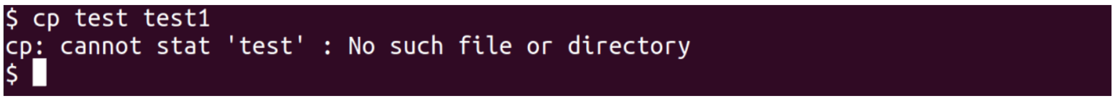
A．　file1 不存在



```
$ cp test ../Project2
cp: cannot stat 'test' : No such file or directory
$ █
```

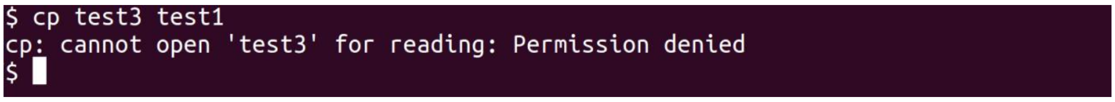


```
$ cp test test
cp: cannot stat 'test' : No such file or directory
$ █
```

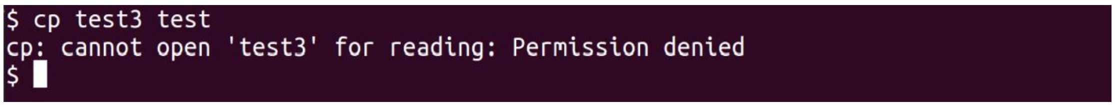


```
$ cp test test1
cp: cannot stat 'test' : No such file or directory
$ █
```

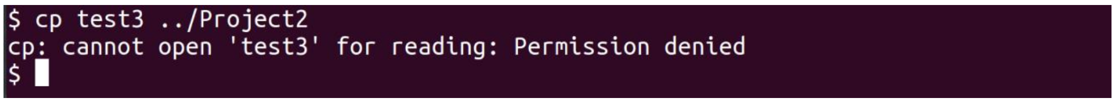
B．　file1 没有读权限



```
$ cp test3 test1
cp: cannot open 'test3' for reading: Permission denied
$ █
```

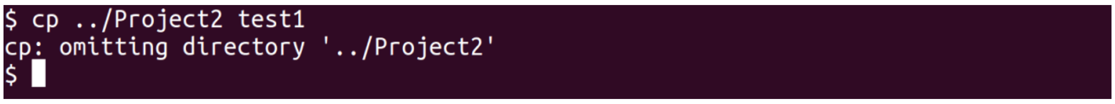


```
$ cp test3 test
cp: cannot open 'test3' for reading: Permission denied
$ █
```

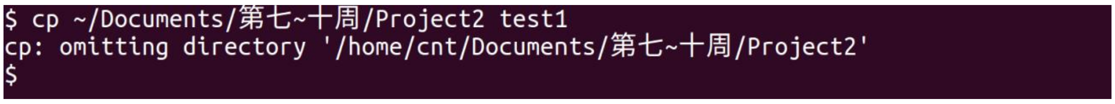


```
$ cp test3 ../Project2
cp: cannot open 'test3' for reading: Permission denied
$ █
```

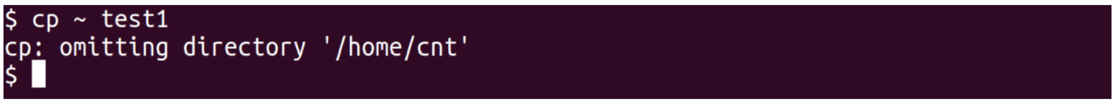
C．　file1 为目录



```
$ cp ../Project2 test1
cp: omitting directory '../Project2'
$ █
```

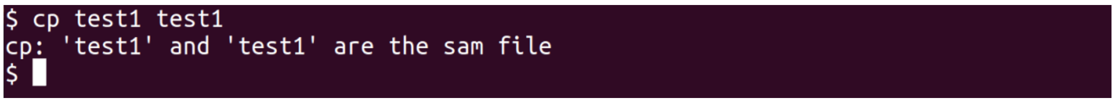


```
$ cp ~/Documents/第七~十周/Project2 test1
cp: omitting directory '/home/cnt/Documents/第七~十周/Project2'
$
```



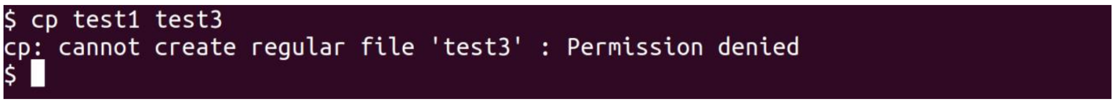
```
$ cp ~ test1
cp: omitting directory '/home/cnt'
$
```

D．　file2 与 file1 在同一子目录下，且重名

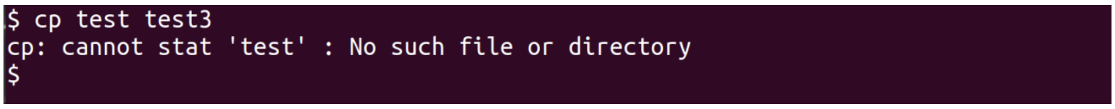


```
$ cp test1 test1
cp: 'test1' and 'test1' are the sam file
$
```

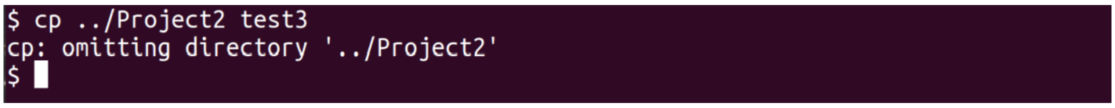
E．file2 没有写权限



```
$ cp test1 test3
cp: cannot create regular file 'test3' : Permission denied
$
```

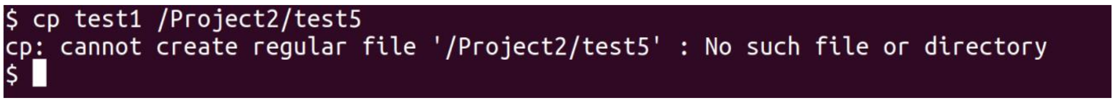


```
$ cp test test3
cp: cannot stat 'test' : No such file or directory
$
```

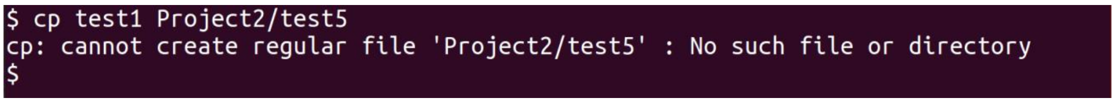


```
$ cp ../Project2 test3
cp: omitting directory '../Project2'
$
```

F．file2 路径格式错误



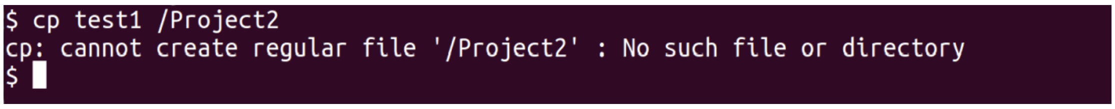
```
$ cp test1 /Project2/test5
cp: cannot create regular file '/Project2/test5' : No such file or directory
$
```



```
$ cp test1 Project2/test5
cp: cannot create regular file 'Project2/test5' : No such file or directory
$
```

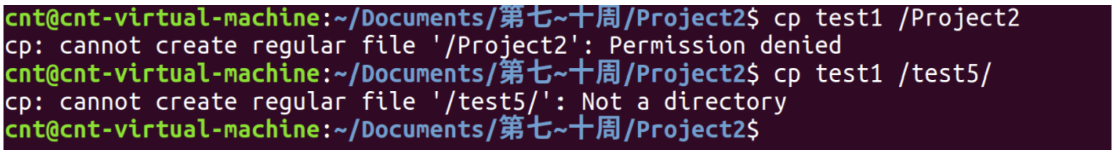
G．file2 为目录且路径错误

该种情况过于复杂，未能实现，采用与 E 中相同报错。



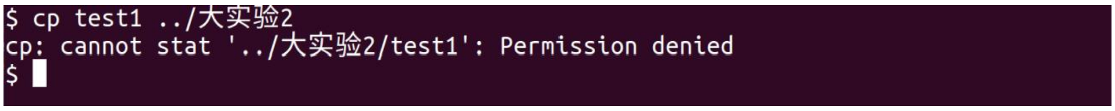
```
$ cp test1 /Project2
cp: cannot create regular file '/Project2' : No such file or directory
$
```

真实 shell 反馈如下：



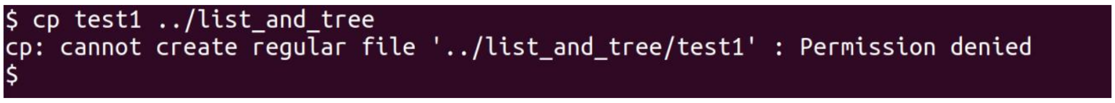
```
cnt@cnt-virtual-machine:~/Documents/第七~十周/Project2$ cp test1 /Project2
cp: cannot create regular file '/Project2': Permission denied
cnt@cnt-virtual-machine:~/Documents/第七~十周/Project2$ cp test1 /test5/
cp: cannot create regular file '/test5/': Not a directory
cnt@cnt-virtual-machine:~/Documents/第七~十周/Project2$
```

H．file2 为目录且无写权限或执行权限



```
$ cp test1 ../大实验2
cp: cannot stat '../大实验2/test1': Permission denied
$ 
```

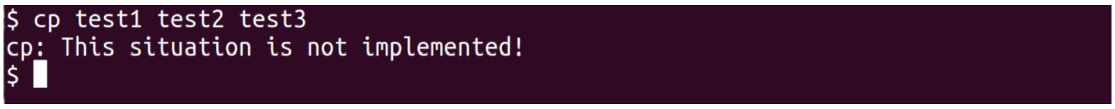
I．file2 为目录，且该目录下有与 file1 重名的文件，同时该文件无写权限



```
$ cp test1 ../list_and_tree
cp: cannot create regular file '../list_and_tree/test1' : Permission denied
$
```

（3） 多个参数

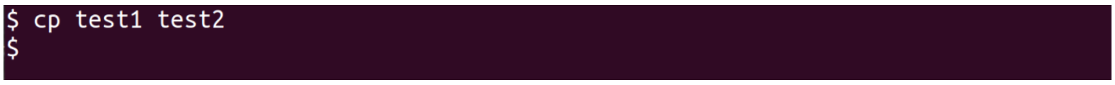
不支持大于等于三个参数的情况



```
$ cp test1 test2 test3
cp: This situation is not implemented!
$ 
```

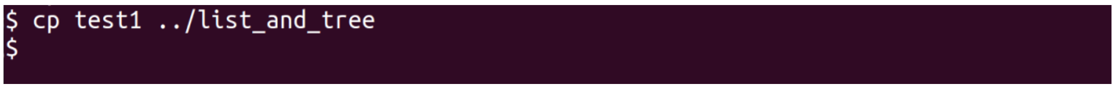
成功示例：

A．file2 为文件



```
$ cp test1 test2
$
```

B．file2 为目录



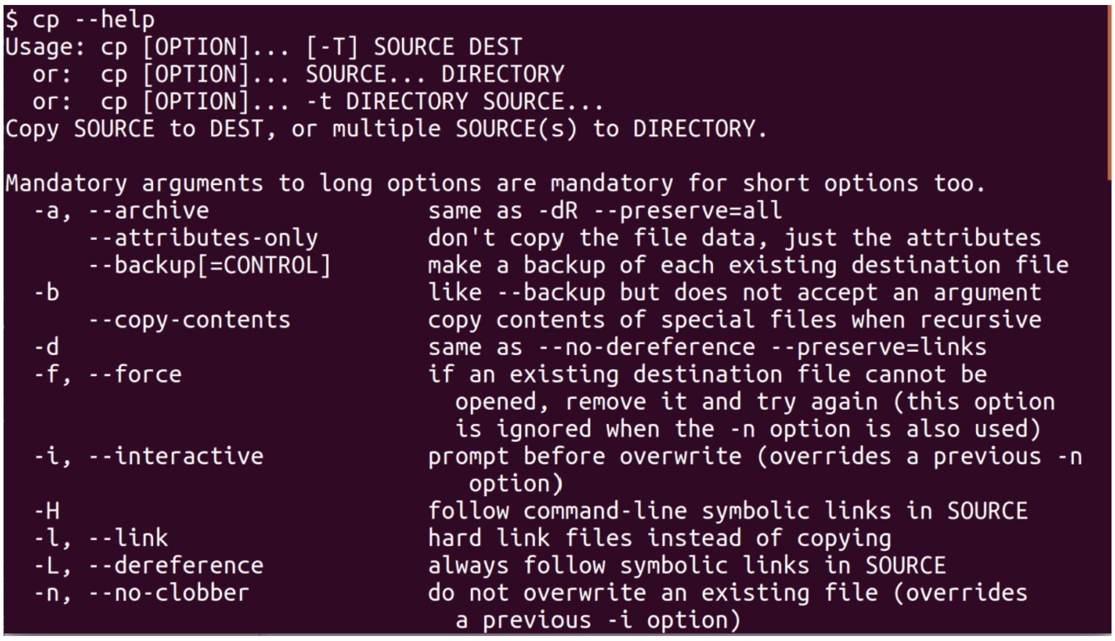
```
$ cp test1 ../list_and_tree
$
```

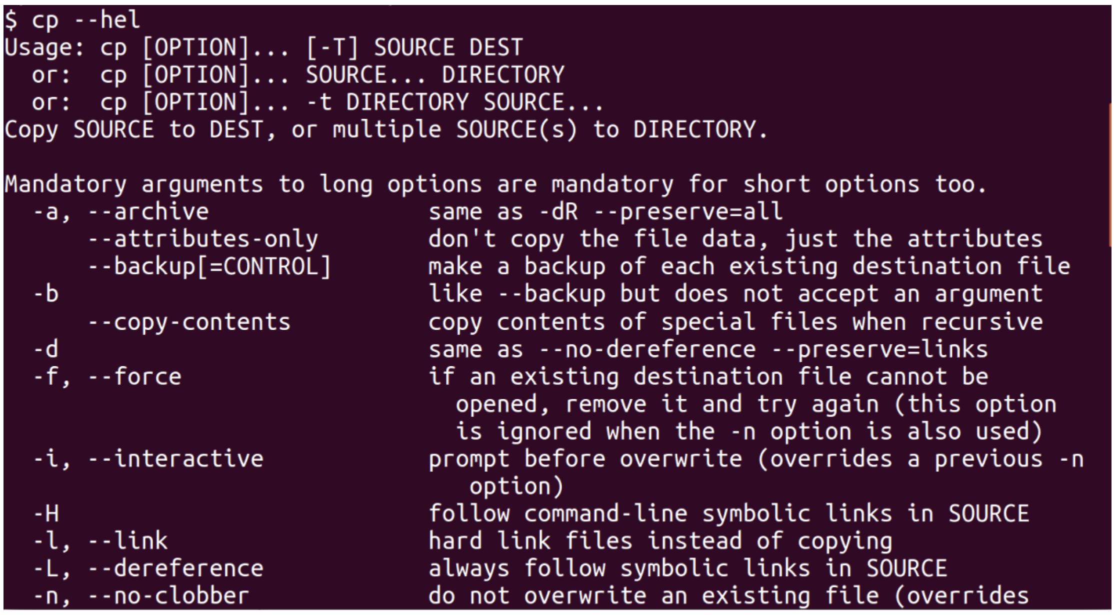
2、cp －r

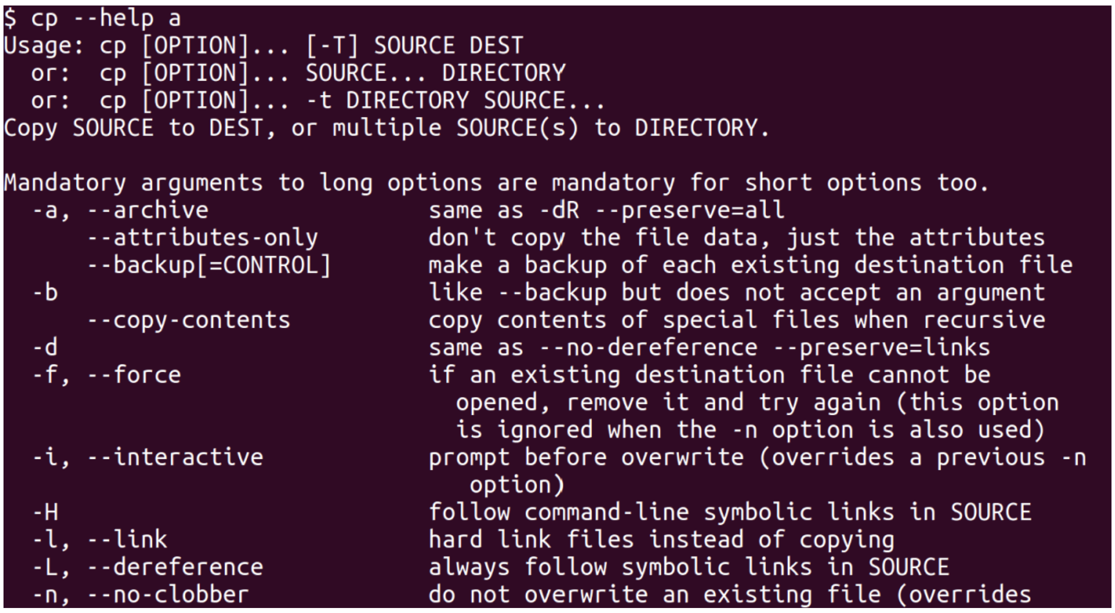
3、cp --help

格式：cp - - help

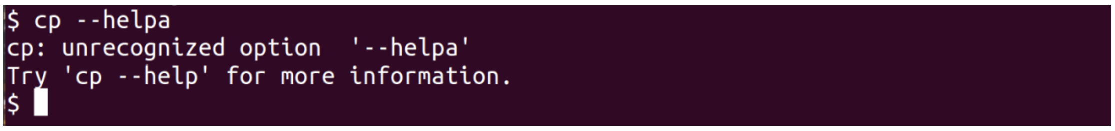
输入--help，--hel，--he，--h 均可被识别，与第三个参数无关；位于第二个参数时亦可被识别，如：cp a --help 等；不支持大于等于三个参数的情况


```
$ cp --help
Usage: cp [OPTION]... [-T] SOURCE DEST
  or:  cp [OPTION]... SOURCE... DIRECTORY
  or:  cp [OPTION]... -t DIRECTORY SOURCE...
Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY.

Mandatory arguments to long options are mandatory for short options too.
  -a, --archive                same as -dR --preserve=all
      --attributes-only        don't copy the file data, just the attributes
      --backup[=CONTROL]       make a backup of each existing destination file
  -b                           like --backup but does not accept an argument
      --copy-contents          copy contents of special files when recursive
  -d                           same as --no-dereference --preserve=links
  -f, --force                  if an existing destination file cannot be
                                 opened, remove it and try again (this option
                                 is ignored when the -n option is also used)
  -i, --interactive            prompt before overwrite (overrides a previous -n
                                  option)
  -H                           follow command-line symbolic links in SOURCE
  -l, --link                   hard link files instead of copying
  -L, --dereference            always follow symbolic links in SOURCE
  -n, --no-clobber             do not overwrite an existing file (overrides
                                 a previous -i option)
```


```
$ cp --hel
Usage: cp [OPTION]... [-T] SOURCE DEST
  or:  cp [OPTION]... SOURCE... DIRECTORY
  or:  cp [OPTION]... -t DIRECTORY SOURCE...
Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY.

Mandatory arguments to long options are mandatory for short options too.
  -a, --archive                same as -dR --preserve=all
      --attributes-only        don't copy the file data, just the attributes
      --backup[=CONTROL]       make a backup of each existing destination file
  -b                           like --backup but does not accept an argument
      --copy-contents          copy contents of special files when recursive
  -d                           same as --no-dereference --preserve=links
  -f, --force                  if an existing destination file cannot be
                                 opened, remove it and try again (this option
                                 is ignored when the -n option is also used)
  -i, --interactive            prompt before overwrite (overrides a previous -n
                                  option)
  -H                           follow command-line symbolic links in SOURCE
  -l, --link                   hard link files instead of copying
  -L, --dereference            always follow symbolic links in SOURCE
  -n, --no-clobber             do not overwrite an existing file (overrides
```


```
$ cp --help a
Usage: cp [OPTION]... [-T] SOURCE DEST
  or:  cp [OPTION]... SOURCE... DIRECTORY
  or:  cp [OPTION]... -t DIRECTORY SOURCE...
Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY.

Mandatory arguments to long options are mandatory for short options too.
  -a, --archive                same as -dR --preserve=all
      --attributes-only        don't copy the file data, just the attributes
      --backup[=CONTROL]       make a backup of each existing destination file
  -b                           like --backup but does not accept an argument
      --copy-contents          copy contents of special files when recursive
  -d                           same as --no-dereference --preserve=links
  -f, --force                  if an existing destination file cannot be
                                 opened, remove it and try again (this option
                                 is ignored when the -n option is also used)
  -i, --interactive            prompt before overwrite (overrides a previous -n
                                  option)
  -H                           follow command-line symbolic links in SOURCE
  -l, --link                   hard link files instead of copying
  -L, --dereference            always follow symbolic links in SOURCE
  -n, --no-clobber             do not overwrite an existing file (overrides
```

但--help 后还有字符时，则无法识别。



```
$ cp --helpa
cp: unrecognized option  '--helpa'
Try 'cp --help' for more information.
$ 
```

其余失败示例请参见 cp 只有一个参数的失败示例。

二、cmp

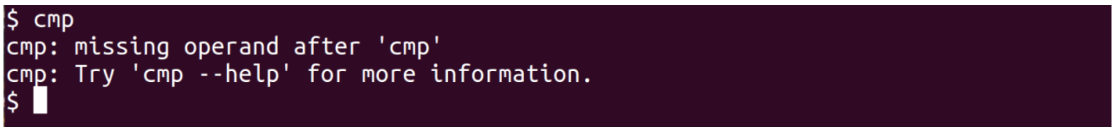
1、 cmp

格式：cmp file1 file2

功能：比较文件 1 和文件 2。当相互比较的两个文件完全一样时，该指令不会显示任何信息。若发现有所差异，命令会显示出第一个不同处的字符编号和行数编号。

失败示例：

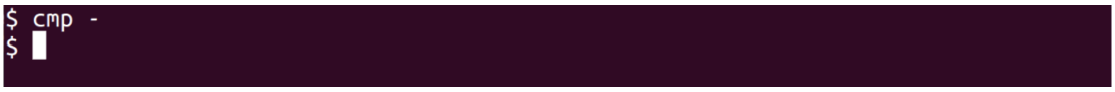
（1） 无参数



```
$ cmp
cmp: missing operand after 'cmp'
cmp: Try 'cmp --help' for more information.
$ 
```

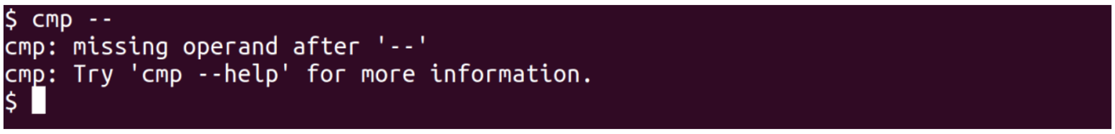
（2） 一个参数

无回显：



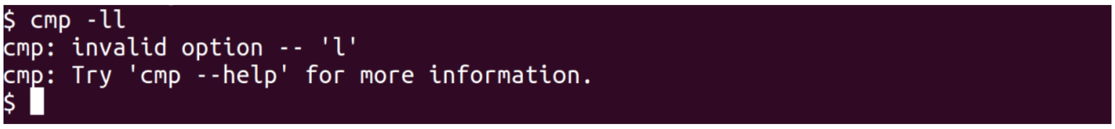
```
$ cmp -
$ 
```

缺少操作数：



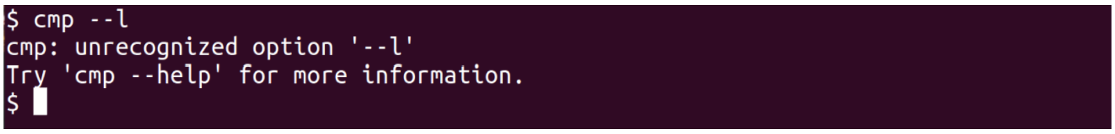
```
$ cmp --
cmp: missing operand after '--'
cmp: Try 'cmp --help' for more information.
$ 
```

无效的命令:



```
$ cmp -ll
cmp: invalid option -- 'l'
cmp: Try 'cmp --help' for more information.
$ 
```

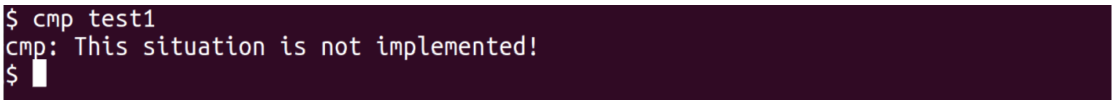
无法识别的命令:



```
$ cmp --l
cmp: unrecognized option '--l'
Try 'cmp --help' for more information.
$ 
```

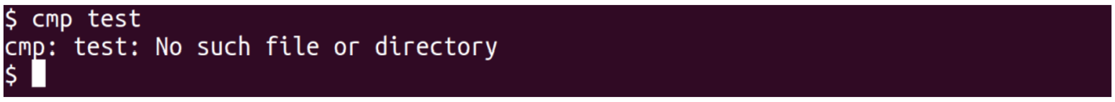
有效参数:

A．file1 存在时：为统计标准输入，未实现。



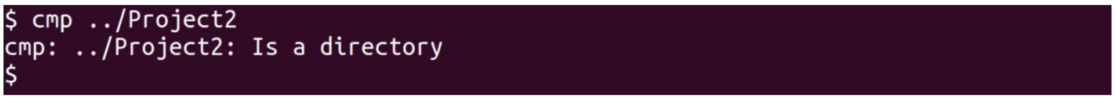
```
$ cmp test1
cmp: This situation is not implemented!
$ 
```

B．file1 不存在时:



```
$ cmp test
cmp: test: No such file or directory
$ 
```

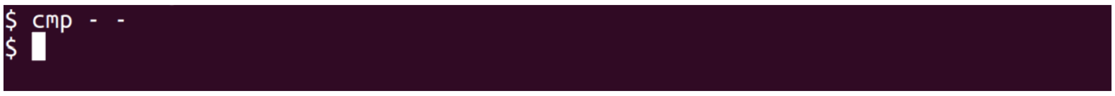
C．file1 为目录时:



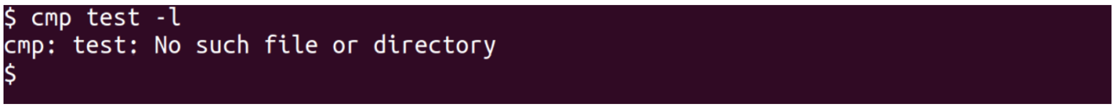
```
$ cmp ../Project2
cmp: ../Project2: Is a directory
$
```

（3）两个参数

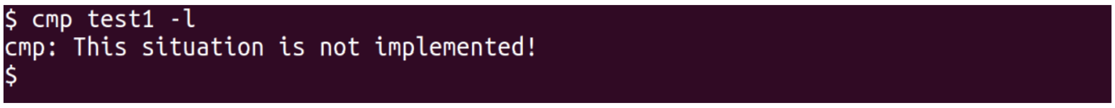
含有无效参数:



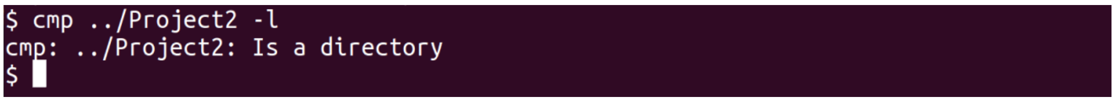
```
$ cmp - -
$ 
```



```
$ cmp test -l
cmp: test: No such file or directory
$
```



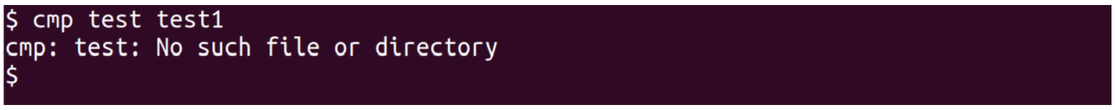
```
$ cmp test1 -l
cmp: This situation is not implemented!
$
```



```
$ cmp ../Project2 -l
cmp: ../Project2: Is a directory
$
```

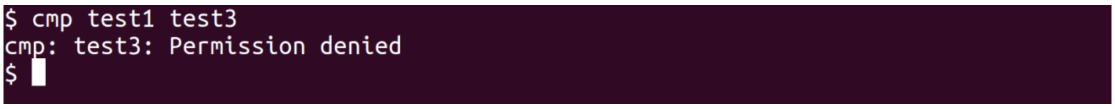
只含有有效参数:

A． file1 或 file2 不存在



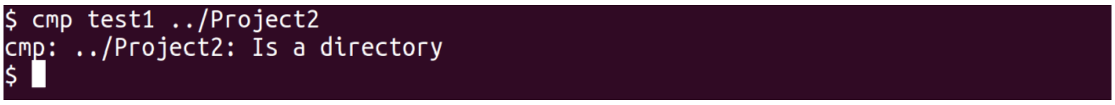
```
$ cmp test test1
cmp: test: No such file or directory
$
```

B．file1 或 file2 没有读权限



```
$ cmp test1 test3
cmp: test3: Permission denied
$
```

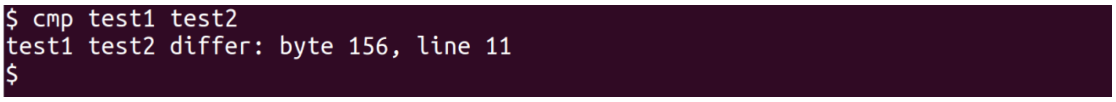
C．file1 或 file2 为目录



```
$ cmp test1 ../Project2
cmp: ../Project2: Is a directory
$
```

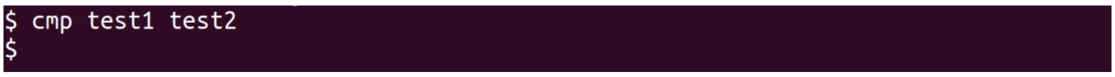
成功示例:

file1 和 file2 中间不同



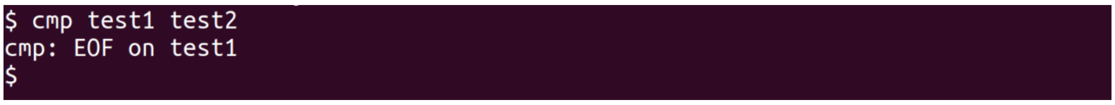
```
$ cmp test1 test2
test1 test2 differ: byte 156, line 11
$
```

file1 和 file2 完全相同



```
$ cmp test1 test2
$
```

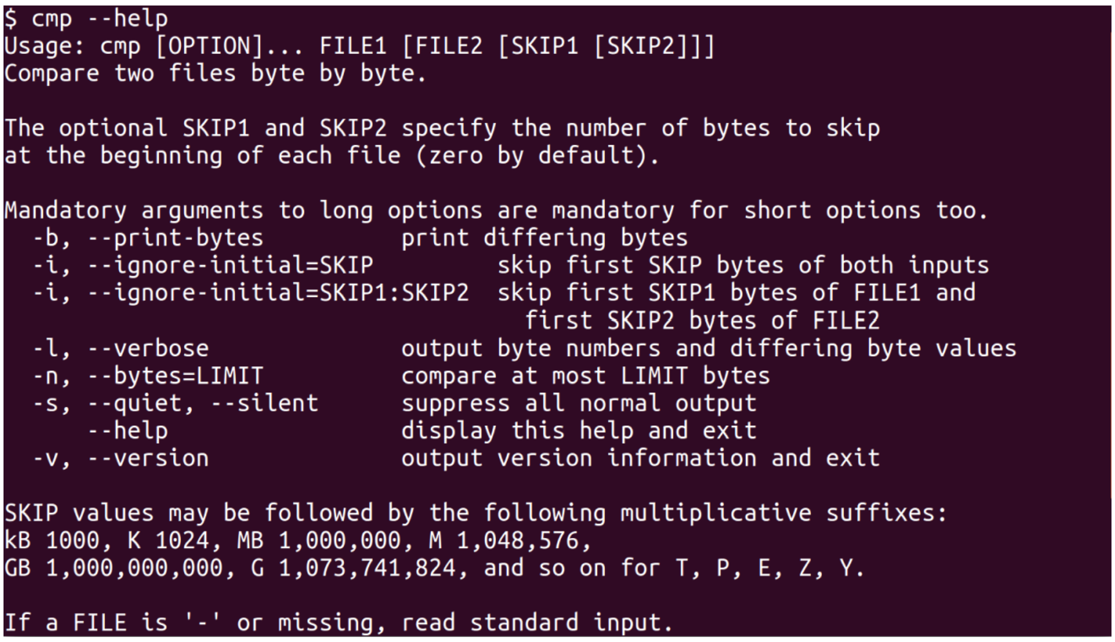
file1 第一部分与 file2 相同，file2 末尾还有数据（以\n 开头）



```
$ cmp test1 test2
cmp: EOF on test1
$
```

2、 cmp --help

与 cp --help 规则完全一致，请参见 cp --help。



```
$ cmp --help
Usage: cmp [OPTION]... FILE1 [FILE2 [SKIP1 [SKIP2]]]
Compare two files byte by byte.

The optional SKIP1 and SKIP2 specify the number of bytes to skip
at the beginning of each file (zero by default).

Mandatory arguments to long options are mandatory for short options too.
  -b, --print-bytes          print differing bytes
  -i, --ignore-initial=SKIP         skip first SKIP bytes of both inputs
  -i, --ignore-initial=SKIP1:SKIP2  skip first SKIP1 bytes of FILE1 and
                                      first SKIP2 bytes of FILE2
  -l, --verbose              output byte numbers and differing byte values
  -n, --bytes=LIMIT          compare at most LIMIT bytes
  -s, --quiet, --silent      suppress all normal output
      --help                 display this help and exit
  -v, --version              output version information and exit

SKIP values may be followed by the following multiplicative suffixes:
kB 1000, K 1024, MB 1,000,000, M 1,048,576,
GB 1,000,000,000, G 1,073,741,824, and so on for T, P, E, Z, Y.

If a FILE is '-' or missing, read standard input.
```

三、wc

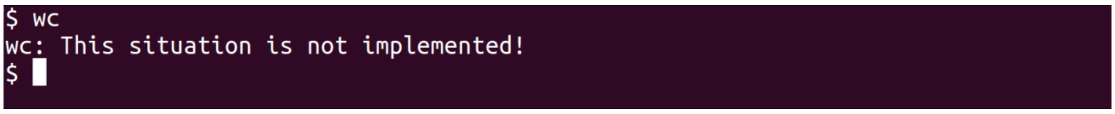
1、 wc

格式：wc file

功能：统计 file 中的行数，字数，字节数，并依次输出。

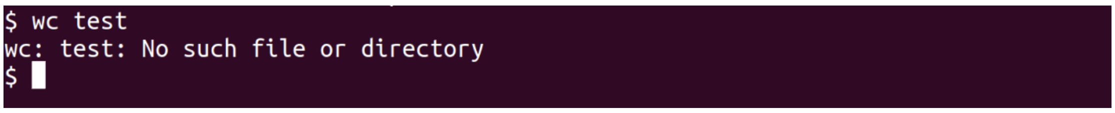
失败示例：

（1） 无参数

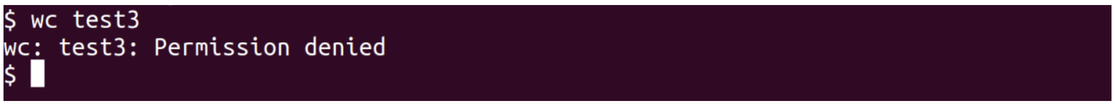

```
$ wc
wc: This situation is not implemented!
$ █
```

（2） 一个参数

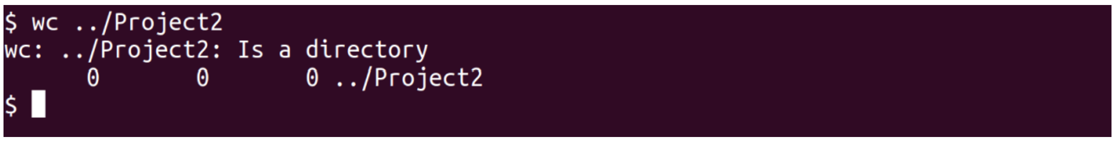
file 不存在：


```
$ wc test
wc: test: No such file or directory
$ █
```

file 没有读权限：

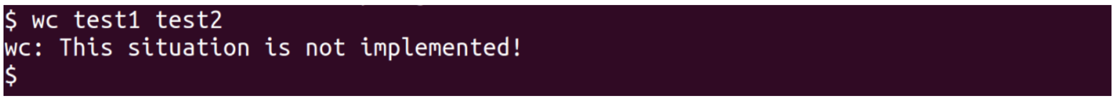

```
$ wc test3
wc: test3: Permission denied
$ █
```

file 为目录：

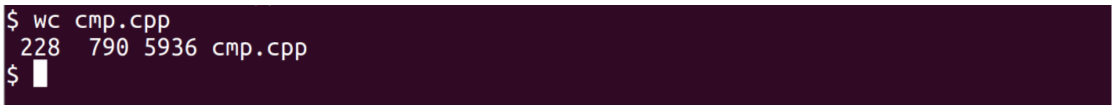

```
$ wc ../Project2
wc: ../Project2: Is a directory
      0       0       0 ../Project2
$ █
```

（3）多个参数

不支持多个参数。


```
$ wc test1 test2
wc: This situation is not implemented!
$
```

成功示例：


```
$ wc cmp.cpp
 228  790 5936 cmp.cpp
$ █
```

2、wc -l/-w/-c

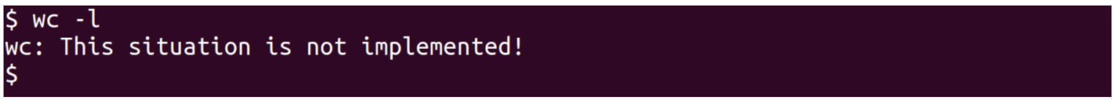
格式：wc -l file / wc -w file / wc -c file

功能：统计 file 中的行数/字数/字节数，并输出。

失败示例：

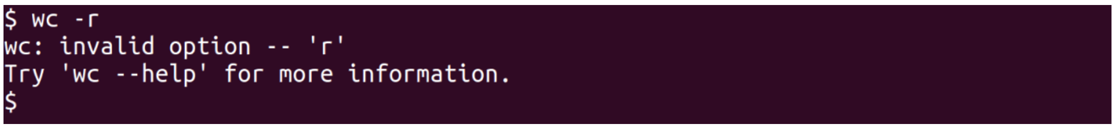
（1） 一个参数（开头为“-”，其余情况请参见 wc 的失败示例）

参数为-l 或-w 或-c：



```
$ wc -l
wc: This situation is not implemented!
$
```

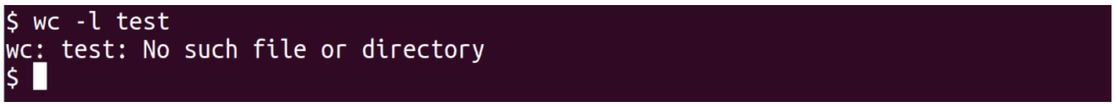
参数为其它：



```
$ wc -r
wc: invalid option -- 'r'
Try 'wc --help' for more information.
$
```

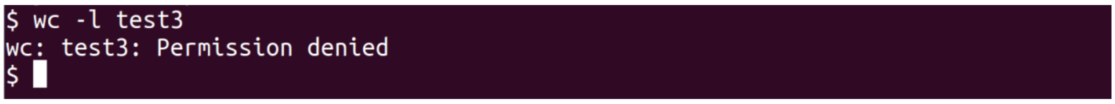
（2） 两个参数

file 不存在



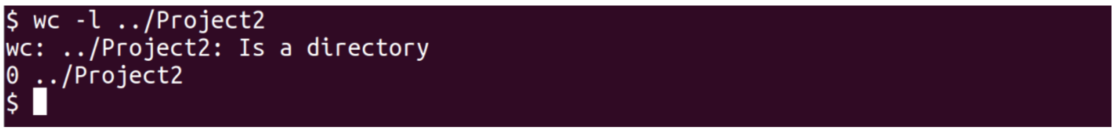
```
$ wc -l test
wc: test: No such file or directory
$ █
```

file 没有读权限



```
$ wc -l test3
wc: test3: Permission denied
$ █
```

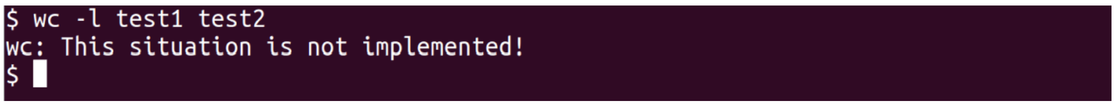
file 为目录



```
$ wc -l ../Project2
wc: ../Project2: Is a directory
0 ../Project2
$ █
```

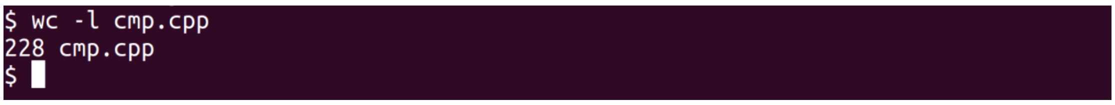
（3） 多个参数

不支持多个参数。



```
$ wc -l test1 test2
wc: This situation is not implemented!
$ █
```

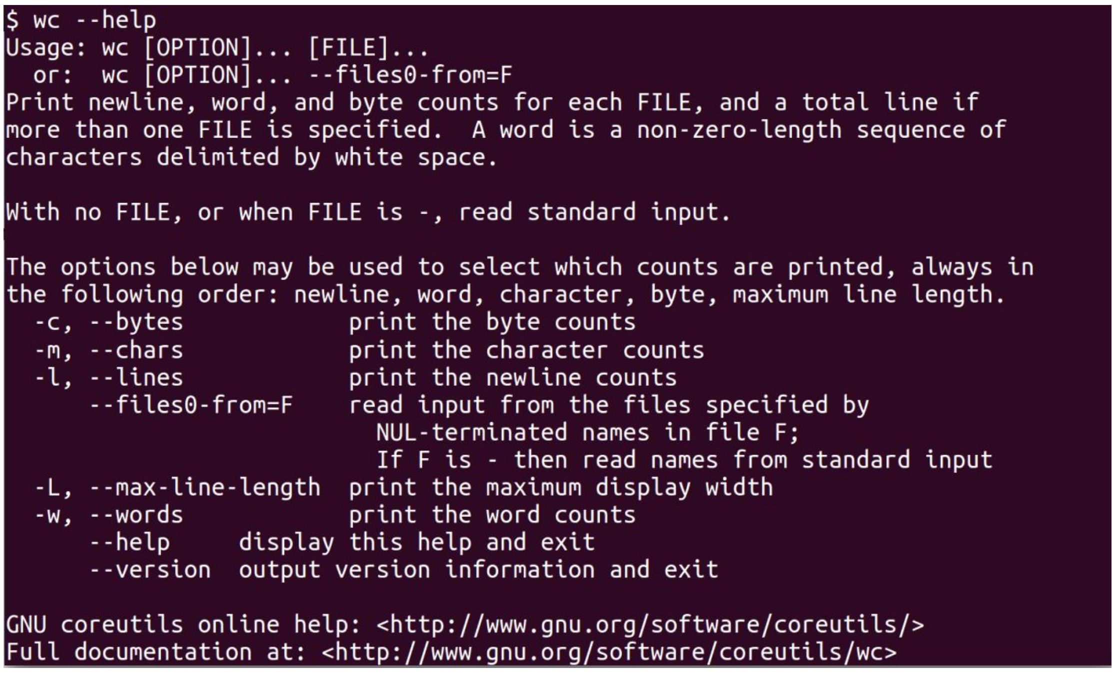
成功示例:



```
$ wc -l cmp.cpp
228 cmp.cpp
$ █
```

3、wc --help

与 cp --help 规则完全一致，请参见 cp --help。



```
$ wc --help
Usage: wc [OPTION]... [FILE]...
  or:  wc [OPTION]... --files0-from=F
Print newline, word, and byte counts for each FILE, and a total line if
more than one FILE is specified.  A word is a non-zero-length sequence of
characters delimited by white space.

With no FILE, or when FILE is -, read standard input.

The options below may be used to select which counts are printed, always in
the following order: newline, word, character, byte, maximum line length.
  -c, --bytes            print the byte counts
  -m, --chars            print the character counts
  -l, --lines            print the newline counts
      --files0-from=F    read input from the files specified by
                           NUL-terminated names in file F;
                           If F is - then read names from standard input
  -L, --max-line-length  print the maximum display width
  -w, --words            print the word counts
      --help     display this help and exit
      --version  output version information and exit

GNU coreutils online help: <http://www.gnu.org/software/coreutils/>
Full documentation at: <http://www.gnu.org/software/coreutils/wc>
```

四、cat

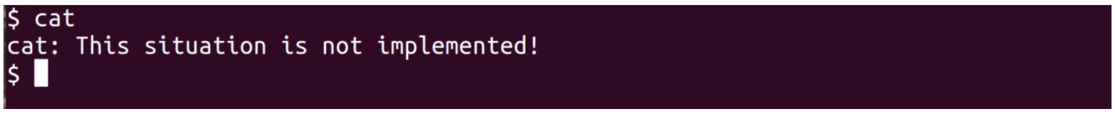
1、cat

格式：cat file

功能：显示 file 中的内容。

失败示例：

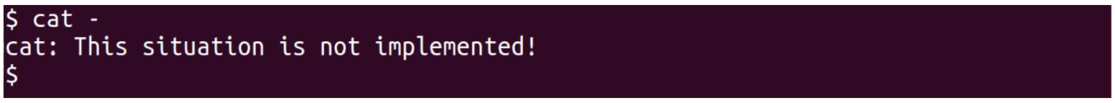
(1) 无参数



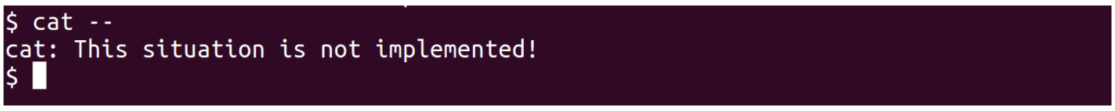
```
$ cat
cat: This situation is not implemented!
$ 
```

（2）一个参数

统计标准输入，未实现：

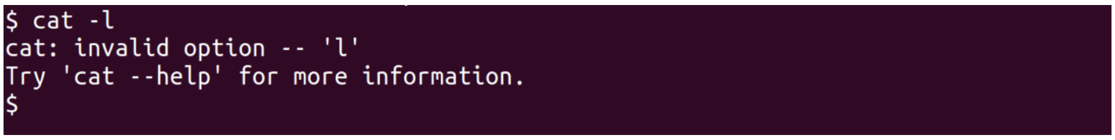


```
$ cat -
cat: This situation is not implemented!
$
```



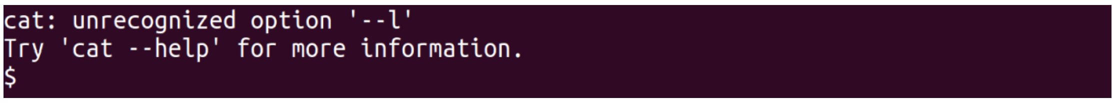
```
$ cat --
cat: This situation is not implemented!
$ 
```

无效的命令：



```
$ cat -l
cat: invalid option -- 'l'
Try 'cat --help' for more information.
$
```

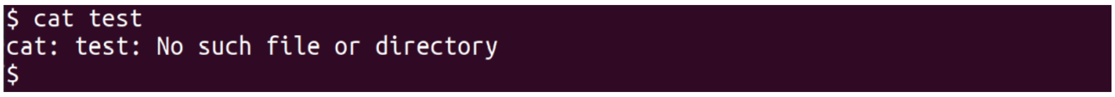
无法识别的命令：



```
cat: unrecognized option '--l'
Try 'cat --help' for more information.
$
```

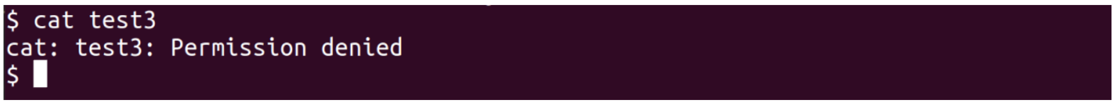
有效命令：

A .　　file 不存在



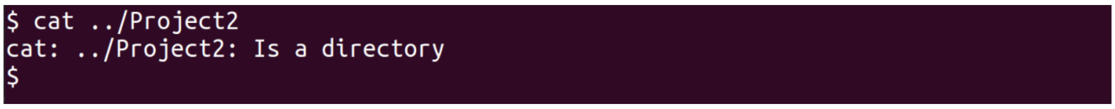
```
$ cat test
cat: test: No such file or directory
$
```

B . file 没有读权限



```
$ cat test3
cat: test3: Permission denied
$ █
```

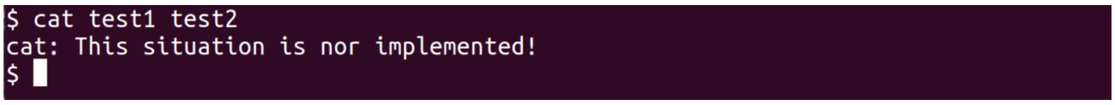
B .　　file 为目录



```
$ cat ../Project2
cat: ../Project2: Is a directory
$
```

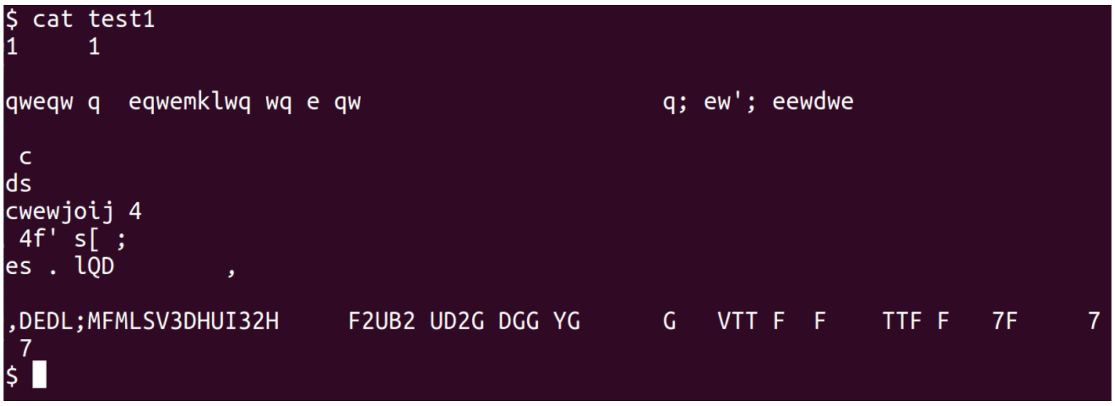
（3）多个参数

不支持多个参数。



```
$ cat test1 test2
cat: This situation is nor implemented!
$ █
```

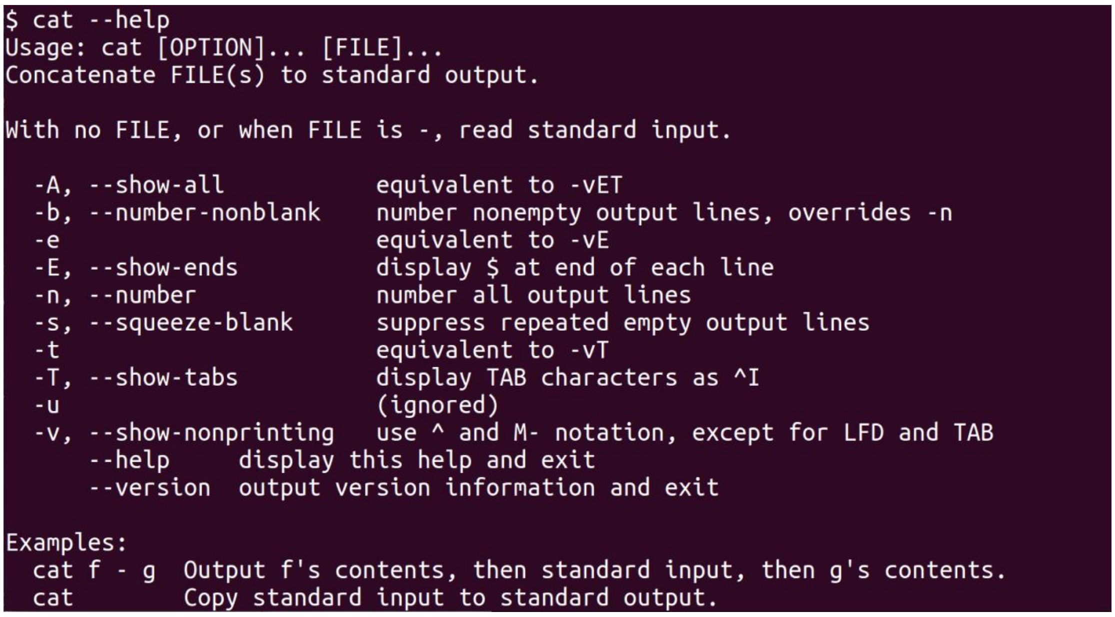
成功示例：



```
$ cat test1
1     1

qweqw q  eqwemklwq wq e qw                      q; ew'; eewdwe

 c
ds
cwewjoij 4
 4f' s[ ;
es . lQD        ,

,DEDL;MFMLSV3DHUI32H     F2UB2 UD2G DGG YG      G   VTT F  F    TTF F   7F      7
 7
$ █
```

3、cat　--help

与 cp --help 规则完全一致，请参见 cp --help。


```
$ cat --help
Usage: cat [OPTION]... [FILE]...
Concatenate FILE(s) to standard output.

With no FILE, or when FILE is -, read standard input.

  -A, --show-all           equivalent to -vET
  -b, --number-nonblank    number nonempty output lines, overrides -n
  -e                       equivalent to -vE
  -E, --show-ends          display $ at end of each line
  -n, --number             number all output lines
  -s, --squeeze-blank      suppress repeated empty output lines
  -t                       equivalent to -vT
  -T, --show-tabs          display TAB characters as ^I
  -u                       (ignored)
  -v, --show-nonprinting   use ^ and M- notation, except for LFD and TAB
      --help     display this help and exit
      --version  output version information and exit

Examples:
  cat f - g  Output f's contents, then standard input, then g's contents.
  cat        Copy standard input to standard output.
```

五、man

六、其它