

ls业务功能的实现

到目前为止，已经解决了参数的定义和传参的问题，下面就要解决业务问题：

1. 列出所有指定路径的文件，默认是不递归的
2. -a 显示所有文件，包括隐藏文件
3. -l 详细列表模式显示

代码实现

```
1 import argparse
2 from pathlib import Path
3 from datetime import datetime
4
5 # 获得一个参数解析器
6 parser = argparse.ArgumentParser(prog='ls', add_help=True, description='list
directory contents')
7 parser.add_argument('path', nargs='?', default='.', help="directory") # 位置
参数，可有可无，缺省值，帮助
8 parser.add_argument('-l', action='store_true', help='use a long listing
format')
9 parser.add_argument('-a', '--all', action='store_true', help='show all
files, do not ignore entries starting with .')
10
11 args = parser.parse_args() # 分析参数，同时传入可迭代的参数
12 print(args) # 打印名词空间中收集的参数
13 parser.print_help() # 打印帮助
14
15
16 def listdir(path, all=False):
17     """列出本目录文件"""
18     p = Path(path)
19     # for f in p.iterdir():
20     #     if not all and f.name.startswith('.'): # 不显示隐藏文件
21     #         continue
22     #     yield f.name
23     # yield from filter(lambda f: not(not all and f.name.startswith('.')),
p.iterdir())
24     # yield from filter(lambda f: all or not f.name.startswith('.'),
p.iterdir())
25     yield from map(lambda x: x.name, filter(lambda f: all or not
f.name.startswith('.'), p.iterdir())) # 路径对象转成路径字符串
26
27 print(*(listdir(args.path)))
28
29
30 # -rw-rw-r-- 1 python python      5 Oct 25 00:07 test4
31 def listdir_detail(path, all=False):
32     """详细列出目录"""
33     p = Path(path)
34     for f in p.iterdir():
35         if not all and f.name.startswith('.'): # 不显示隐藏文件
36             continue
37         # mode 硬链接 属主 属组 字节 时间 name
```

```

38         stat = f.stat()
39         mtime = datetime.fromtimestamp(stat.st_mtime).strftime('%Y %m %d
%H:%M:%S')
40         yield (stat.st_mode, stat.st_nlink, stat.st_uid, stat.st_gid,
st.st_size, mtime, f.name)
41
42
43     print(*(listdirdetail(args.path)))

```

mode是整数，八进制描述的权限，最终显示为rwx的格式。

```

1  import stat
2
3  mode = stat.filemode(st.st_mode) # 利用stat的函数

```

合并列出文件函数

listdirdetail和listdir几乎一样，重复太多，合并

```

1  import argparse
2  from pathlib import Path
3  from datetime import datetime
4  import stat
5
6  # 获得一个参数解析器
7  parser = argparse.ArgumentParser(prog='ls', add_help=True, description='list
directory contents')
8  parser.add_argument('path', nargs='?', default='.', help="directory") # 位置
参数，可有可无，缺省值，帮助
9  parser.add_argument('-l', action='store_true', help='use a long listing
format')
10 parser.add_argument('-a', '--all', action='store_true', help='show all
files, do not ignore entries starting with .')
11
12 args = parser.parse_args() # 分析参数，同时传入可迭代的参数
13 print(args) # 打印名词空间中收集的参数
14 parser.print_help() # 打印帮助
15
16
17 def listdir(path, all=False, detail=False):
18     """详细列出本目录"""
19     p = Path(path)
20     for i in p.iterdir():
21         if not all and i.name.startswith('.'): # 不显示隐藏文件
22             continue
23
24         if not detail:
25             yield (i.name,)
26         else:
27             # -rw-rw-r-- 1 python python 5 Oct 25 00:07 test4
28             # mode 硬链接 属主 属组 字节 时间 name
29             st = i.stat()
30             mode = stat.filemode(st.st_mode)
31             mtime = datetime.fromtimestamp(st.st_atime).strftime('%Y/%m/%d
%H:%M:%S')

```

```
32         yield (mode, st.st_nlink, st.st_uid, st.st_gid, st.st_size,
33               mtime, i.name)
34
35 print(*listdir(args.path, detail=True), sep='\n')
```

排序

请实现-r选项，按文件名倒排。

完成功能，并重构代码

```

1 import argparse
2 from pathlib import Path
3 from datetime import datetime
4 import stat
5
6 # 获得一个参数解析器
7 parser = argparse.ArgumentParser(prog='ls', add_help=True, description='list
directory contents')
8 parser.add_argument('path', nargs='?', default='.', help="directory") # 位置
参数, 可有可无, 缺省值, 帮助
9 parser.add_argument('-l', action='store_true', help='use a long listing
format')
10 parser.add_argument('-a', '--all', action='store_true', help='show all
files, do not ignore entries starting with .')
11 parser.add_argument('-r', '--reverse', action='store_true', help="reverse
order while sorting")
12
13 args = parser.parse_args() # 分析参数, 同时传入可迭代的参数
14 print(args) # 打印名词空间中收集的参数
15 parser.print_help() # 打印帮助
16
17
18 def listdir(path, all=False, detail=False, reverse=False):
19     """详细列出本目录"""
20     def _listdir(path, all, detail, reverse):
21         p = Path(path)
22         for i in p.iterdir():
23             if not all and i.name.startswith('.'): # 不显示隐藏文件
24                 continue
25
26             if not detail:
27                 yield (i.name,)
28             else:
29                 # -rw-rw-r-- 1      python python      5 Oct 25 00:07 test4
30                 # mode      硬链接 属主      属组      字节 时间      name
31                 st = i.stat()
32                 mode = stat.filemode(st.st_mode)
33                 mtime =
datetime.fromtimestamp(st.st_atime).strftime('%Y/%m/%d %H:%M:%S')
34                 yield (mode, st.st_nlink, st.st_uid, st.st_gid, st.st_size,
mtime, i.name)
35
36     return sorted(_listdir(path, all, detail, reverse), key=lambda x:
x[len(x)-1], reverse=reverse)
37

```

```
38
39 print(*listdir(args.path, detail=True, reverse=True), sep='\n')
```

-h的实现

-h, --human-readable, 如果-l存在, -h有效。

1、增加选项参数

```
1 parser = argparse.ArgumentParser(prog='ls', description='list directory
  contents', add_help=False)
2 parser.add_argument('-h', '--human-readable', action='store_true', help='with
  -l, print sizes in human readable format')
```

2、增加一个函数, 能够解决单位转换的

```
1 def _gethuman(size: int):
2     units = " KMGTP"
3     depth = 0
4     while size > 1000 and depth < len(units) - 1:
5         # 当前size大于1000, 且depth不是最后一个进入循环
6         depth += 1
7         size //= 1000
8     return "{}{}".format(size, units[depth] if depth else '')
```

3、在-l逻辑部分增加处理

```
1 size = stat.st_size if not human else _gethuman(stat.st_size)
```

其他的完善

uid、gid的转换

pwd模块, The password database, 提供访问Linux、Unix 的 password文件的方式。windows没有。

```
pwd.getpwuid(Path().stat().st_uid).pw_name
```

grp模块, Linux、Unix获取组信息的模块。windows没有

```
grp.getgrgid(Path().stat().st_gid).gr_name
```

pathlib模块, Path().group() 或者 Path().owner()也可以, 本质上它们就是调用pwd模块和grp模块。

由于windows不支持, 这次可以不加这个uid、gid的转换

最终代码

```
1 import argparse
2 from pathlib import Path
3 from datetime import datetime
4 import stat
5
6 # 获得一个参数解析器
```

```

7 parser = argparse.ArgumentParser(prog='ls', add_help=False,
  description='list directory contents')
8 parser.add_argument('path', nargs='?', default='.', help="directory") # 位置
  参数, 可有可无, 缺省值, 帮助
9 parser.add_argument('-l', action='store_true', dest='long', help='use a long
  listing format')
10 parser.add_argument('-a', '--all', action='store_true', help='show all
  files, do not ignore entries starting with .')
11 parser.add_argument('-r', '--reverse', action='store_true', help="reverse
  order while sorting")
12 parser.add_argument('-h', '--human-readable', action='store_true',
13 dest='human', help='with -l, print sizes in human
  readable format')
14
15
16 def _gethuman(size: int):
17     units = " KMGTP"
18     depth = 0
19     while size > 1000 and depth < len(units) - 1:
20         # 当前size大于1000, 且depth不是最后一个进入循环
21         depth += 1
22         size //= 1000
23     return "{}{}".format(size, units[depth] if depth else '')
24
25 def _listdir(path, all, detail, reverse, human):
26     p = Path(path)
27     for i in p.iterdir():
28         if not all and i.name.startswith('.'): # 不显示隐藏文件
29             continue
30
31         if not detail:
32             yield (i.name,)
33         else:
34             # -rw-rw-r-- 1 python python 5 Oct 25 00:07 test4
35             # mode 硬链接 属主 属组 字节 时间 name
36             st = i.stat()
37             mode = stat.filemode(st.st_mode)
38             mtime = datetime.fromtimestamp(st.st_atime).strftime('%Y/%m/%d
%H:%M:%S')
39             size = st.st_size if not human else _gethuman(st.st_size)
40             yield (mode, st.st_nlink, st.st_uid, st.st_gid, size, mtime,
i.name)
41
42 def listdir(path, all=False, detail=False, reverse=False, human=False):
43     """详细列出本目录"""
44     return sorted(_listdir(path, all, detail, reverse, human), key=lambda x:
x[len(x)-1], reverse=reverse)
45
46 if __name__ == '__main__':
47     #args = parser.parse_args('-lrha'.split()) # 分析参数, 同时传入可迭代的参数
48     args = parser.parse_args()
49     print(args) # 打印名词空间中收集的参数
50     parser.print_help() # 打印帮助
51     files = listdir(args.path, args.all, args.long, args.reverse,
args.human)
52     print(*files, sep='\n')

```

测试

```
1 $ python xxx.py -lah -r  
2 $ python xxx.py /etc/ -lahr
```

