

# 使用 golang 开发 Linux 命令行实用程序中的 selpg

曲翔宇 15335124

## 实验要求

[潘老师网站](#)

## 参考资料

[c语言开发selpg](#)

[go中文网教程](#)

## 代码结构

完整代码及注释在selpg.go中

我将 main 主函数分为三个子模块，分别为

1. `params_init` : 用于设定参数默认值
2. `check_params` : 用于检查命令行参数输入是否正确
3. `i_print` : 用于解析参数信息，并按照要求打印

```
//帮助信息-h
func usage() {
    fmt.Printf("Usage : selpg -s Number -e Number [options] [filename]\n\n")
    fmt.Printf("Arguments and Options:\n")
    fmt.Printf("-s Number\t: 从<Number>页尾开始打印, 即<Number+1>页头开始打印\n")
    fmt.Printf("-e Number\t: 到<Number>页尾结束打印\n")
    fmt.Printf("-l Number\t: 每页有<Number>行, 默认为72行\n")
    fmt.Printf("-f\t\t: 每页之间是否用\\f作为分隔, 默认为否\n")
    fmt.Printf("[filename]\t: 要打印的文件名, 例如doc.txt, 如果没有显示文件名则会接收命令行输入, 并以control+d作为")
}
```

这是帮助菜单信息

```
//参数默认值
func params_init(args *selpgargs) {
    flag.Usage = usage
    flag.IntVar(&args.start_page, "s", -1, "开始页")
    flag.IntVar(&args.end_page, "e", -1, "结束页")
    flag.IntVar(&args.page_len, "l", 10, "每页行数")
    flag.BoolVar(&args.form_deli, "f", false, "页是否以\\f分隔")
    flag.StringVar(&args.destination, "d", "", "确定打印机")
    flag.Parse()
}
```

设定各参数的默认值

```
//确定输出目标
func cmd_print(args *selpgargs, line string, stdin io.WriteCloser) {
    //写文件
    if args.destination != "" {
        stdin.Write([]byte(line + "\n"))
    } else {
        //命令行输出
        fmt.Println(line)
    }
}
```

确定并打印到命令行还是指定文件中

## 示例测试

```
apple@Quxy ~/Docs/selpg ls
doc.txt i_selpg.go
apple@Quxy ~/Docs/selpg go build i_selpg.go
apple@Quxy ~/Docs/selpg ls
doc.txt i_selpg i_selpg.go
apple@Quxy ~/Docs/selpg ./i_selpg -h
Usage : selpg -s Number -e Number [options] [filename]

Arguments and Options:
-s Number      : 从<Number>页尾开始打印, 即<Number+1>页头开始打印
-e Number      : 到<Number>页尾结束打印
-l Number      : 每页有<Number>行, 默认为72行
-f            : 每页之间是否用\nf作为分隔, 默认为否
[filename]     : 要打印的文件名, 例如doc.txt,如果未有显示文件名则会接收命令行输入, 并以control+d作为结束标志
```

首先是编译成可执行文件, 并用参数 `-h` 显示帮助菜单(Usage)

参考python的帮助菜单样式

```
apple@Quxy ~/Docs/selpg ./i_selpg -s 0 -e 0 doc.txt
line1
line2
line3
line4
line5
line6
line7
line8
line9
line10
apple@Quxy ~/Docs/selpg
```

打印文件 `doc.txt` 的首页, 打印首页

注: 为了方便命令行显示, 我将每页默认页数设置为10; 而且第一页的参数为0, 第二页为1, 以此类推

```
apple@Quxy > ~/Docs/selpg > ./i_selpg -s 0 -e 1 doc.txt
line1
line2
line3
line4
line5
line6
line7
line8
line9
line10
line11
line12
line13
line14
line15
line16
line17
line18
line19
line20
```

和上步操作相似，不过这次是打印前两页

```
apple@Quxy > ~/Docs/selpg > ./i_selpg -s 1 -e 1 doc.txt
line11
line12
line13
line14
line15
line16
line17
line18
line19
line20
apple@Quxy > ~/Docs/selpg >
```

同上两步操作相似，不过这次是打印第二页

```
apple@Quxy > ~/Docs/selpg > ./i_selpg -s 0 -e 0 -l 15 doc.txt
line1
line2
line3
line4
line5
line6
line7
line8
line9
line10
line11
line12
line13
line14
line15
apple@Quxy > ~/Docs/selpg >
```

手动声明每页行数，这里声明为15行，可见打印出第一页的十五行

```
apple@Quxy ~/Docs/selpg ./i_selpg -s 0 -e 1 -l 3
in1
in2
in3
in4
in5
in6
in7
in8
in9
in10
in1
in2
in3
in4
in5
in6
apple@Quxy ~/Docs/selpg
```

之前的打印源都指明为 `doc.txt`，而这里没有显示指明源文件，所以输入源为命令行输入，可见我在命令行输入了10行，再按 `control-d` 标志结束输入，并按照参数要求打印前两页，每页3行

```
apple@Quxy ~/Docs/selpg ./i_selpg -s=0 -e=1 doc.txt
line1
line2
line3
line4
line5
line6
line7
line8
line9
line10
line11
line12
line13
line14
line15
line16
line17
line18
line19
line20
apple@Quxy ~/Docs/selpg
```

当然，参数不只是 `-<arg> <Number>`，也可以是 `-<arg>=<Number>` 格式

```

apple@Quxy ~/Docs/selpg$ ./i_selpg -s 0 -e 1 -d printer1 doc.txt
1 line1
2 line2
3 line3
4 line4
5 line5
6 line6
7 line7
8 line8
9 line9
10 line10
11 line11
12 line12
13 line13
14 line14
15 line15
16 line16
17 line17
18 line18
19 line19
20 line20

```

这里指定了打印机 `printer1`，打印到命令行

```

apple@Quxy ~/Docs/selpg$ ./i_selpg -s 0 -e 1 doc.txt > des.txt
apple@Quxy ~/Docs/selpg$

```

这里指定了目的文件，而不是命令行，所以打印到 `des.txt` 中：

```

des.txt
line1
line2
line3
line4
line5
line6
line7
line8
line9
line10
line11
line12
line13
line14
line15
line16
line17
line18
line19
line20

```

## 错误测试

```

apple@Quxy ~/Docs/selpg$ ./i_selpg
Error: 必须输入前两个参数

Usage : selpg -s Number -e Number [options] [filename]

Arguments and Options:
-s Number      : 从<Number>页尾开始打印，即<Number+1>页头开始打印
-e Number      : 到<Number>页尾结束打印
-l Number      : 每页有<Number>行，默认为72行
-f            : 每页之间是否用\\f作为分隔，默认为否
[filename]     : 要打印的文件名，例如doc.txt,如果未有显示文件名则会接收命令行输入，并以control+d作为结束标志

```

首先如果不输入参数，则会报错

## 每次显示报错信息后，还会打印帮助菜单

```
✖ apple@Quxy ~/Docs/selpg ./i_selpg -s 0 -l 2 -e 2
Error: 第二个参数必须为 -e <Number>

Usage : selpg -s Number -e Number [options] [filename]

Arguments and Options:
-s Number      : 从<Number>页尾开始打印，即<Number+1>页头开始打印
-e Number      : 到<Number>页尾结束打印
-l Number      : 每页有<Number>行，默认为72行
-f            : 每页之间是否用\\f作为分隔，默认为否
[filename]     : 要打印的文件名，例如doc.txt,如果未有显示文件名则会接收命令行输入，并以control+d作为结束标志
```

```
✖ apple@Quxy ~/Docs/selpg ./i_selpg -e 1 -s 0
Error: 第一个参数必须为 -s <Number>

Usage : selpg -s Number -e Number [options] [filename]

Arguments and Options:
-s Number      : 从<Number>页尾开始打印，即<Number+1>页头开始打印
-e Number      : 到<Number>页尾结束打印
-l Number      : 每页有<Number>行，默认为72行
-f            : 每页之间是否用\\f作为分隔，默认为否
[filename]     : 要打印的文件名，例如doc.txt,如果未有显示文件名则会接收命令行输入，并以control+d作为结束标志
```

以上是参数位置不正确而导致的报错问题

还有一些常识错误情况，如下：

```
✖ apple@Quxy ~/Docs/selpg ./i_selpg -s -3 -e 1
Error: 页数必须为非负数且尾页必须大于等于首页

Usage : selpg -s Number -e Number [options] [filename]

Arguments and Options:
-s Number      : 从<Number>页尾开始打印，即<Number+1>页头开始打印
-e Number      : 到<Number>页尾结束打印
-l Number      : 每页有<Number>行，默认为72行
-f            : 每页之间是否用\\f作为分隔，默认为否
[filename]     : 要打印的文件名，例如doc.txt,如果未有显示文件名则会接收命令行输入，并以control+d作为结束标志
```

```
apple@Quxy ~/Docs/selpg ./i_selpg -s 2 -e 1
Error: 页数必须为非负数且尾页必须大于等于首页

Usage : selpg -s Number -e Number [options] [filename]

Arguments and Options:
-s Number      : 从<Number>页尾开始打印，即<Number+1>页头开始打印
-e Number      : 到<Number>页尾结束打印
-l Number      : 每页有<Number>行，默认为72行
-f            : 每页之间是否用\\f作为分隔，默认为否
[filename]     : 要打印的文件名，例如doc.txt,如果未有显示文件名则会接收命令行输入，并以control+d作为结束标志
```