

准备工作:

1. 建好头文件, 以及所需源文件



```
[felixg@192 progress bar]$ touch ProBar.h
[felixg@192 progress bar]$ touch ProBar.c
[felixg@192 progress bar]$ touch Main.c
[felixg@192 progress bar]$ ls
Main.c  ProBar.c  ProBar.h
[felixg@192 progress bar]$
```

2. 建立Make file文件



```
[felixg@192 progress bar]$ ls > Makefile
[felixg@192 progress bar]$ ls
Main.c  Makefile  ProBar.c  ProBar.h
[felixg@192 progress bar]$
```

3. 写入Makefile: 两种写法都可以



```
1: Makefile+ 斌
1  procbar: ProBar.c Main.c
2      gcc ProBar.c Main.c -o procbar
3  .PHONY: clean
4  clean:
5      rm -f procbar
6
7
```

```
1 procbar: ProBar.c Main.c
2 gcc $^ -o $@
3 #gcc ProBar.c Main.c -o procbar
4 .PHONY: clean
5 clean:
6 rm -f procbar
7
```

4. 测试一下:



5. 加@后, 语句仍会执行, 但不会打印



行缓冲区概念



运行printf, 3秒后, 退出



去掉\n后, 先执行sleep(3),再执行printf

原因: C语言运行的顺序的确是从上往下执行, 只不过运行到printf的结果被放到了缓冲区, sleep结束后与=才显示到屏幕。

1.缓冲区

缓冲区就是一块内存区域

- a. 无缓冲 (立即)
- b. 行缓冲 (\n)
- c. 全缓冲 (缓冲区满的时候再刷新 -- 磁盘文件file)
- e. 程序退出, 缓冲区内容自动刷新

fflush(stdout)

不想写\n，还行让printf的内容先显示出来，怎么办？



fflush(stdout) -- 把缓冲区的内容刷新到标准输出流

2. \r \n

\r -- 回车，将光标回复到当前行的最开始

\n -- 换行，不管有没有在最开始，从当前位置新起一行继续往右走

在C语言中，\n代表换行和回车两个动作

\n



\r

\r\n

运行结果一样，原因就是因为在C语言中\n默认执行\r



\r

行缓冲，遇到\n才刷新，碰到\r不刷新，因此看不到输出



使用fflush(stdout)，就可以将缓冲区的内容频闪到屏幕上 -- 倒计时程序

原理：9\r，遇到\r就会回到最开始，并把9覆盖掉，8\r...



进度条原理：

#

##

###

...

每次多输出一个#, 回车把之前的#覆盖掉, 不断打印, 不断回车

代码实现:

```
1  #include "ProBar.h"

2  void process_bar()
3  {
4      char bar[NUM];
5      memset(bar, '\0', sizeof(bar));
6      const char* lable="/-\\\";
7      int i=0;
8      while(i<=100)
9      {
10         printf("[%10s][%-3d %][%c]\r", bar, i, lable[i%4]);
11         //把[]空间预留出来, 格式化输出%-100左对齐, 默认右对齐 (输出方向)
12         //%%是输出%
13         //因为lable中的元素个数只有4个, 因此每四个为循环, 要i%4

14         fflush(stdout);
15         bar[i++]='#';
16         usleep(50000); //usleep微秒
17     }
18     printf("\n");
19 }
```