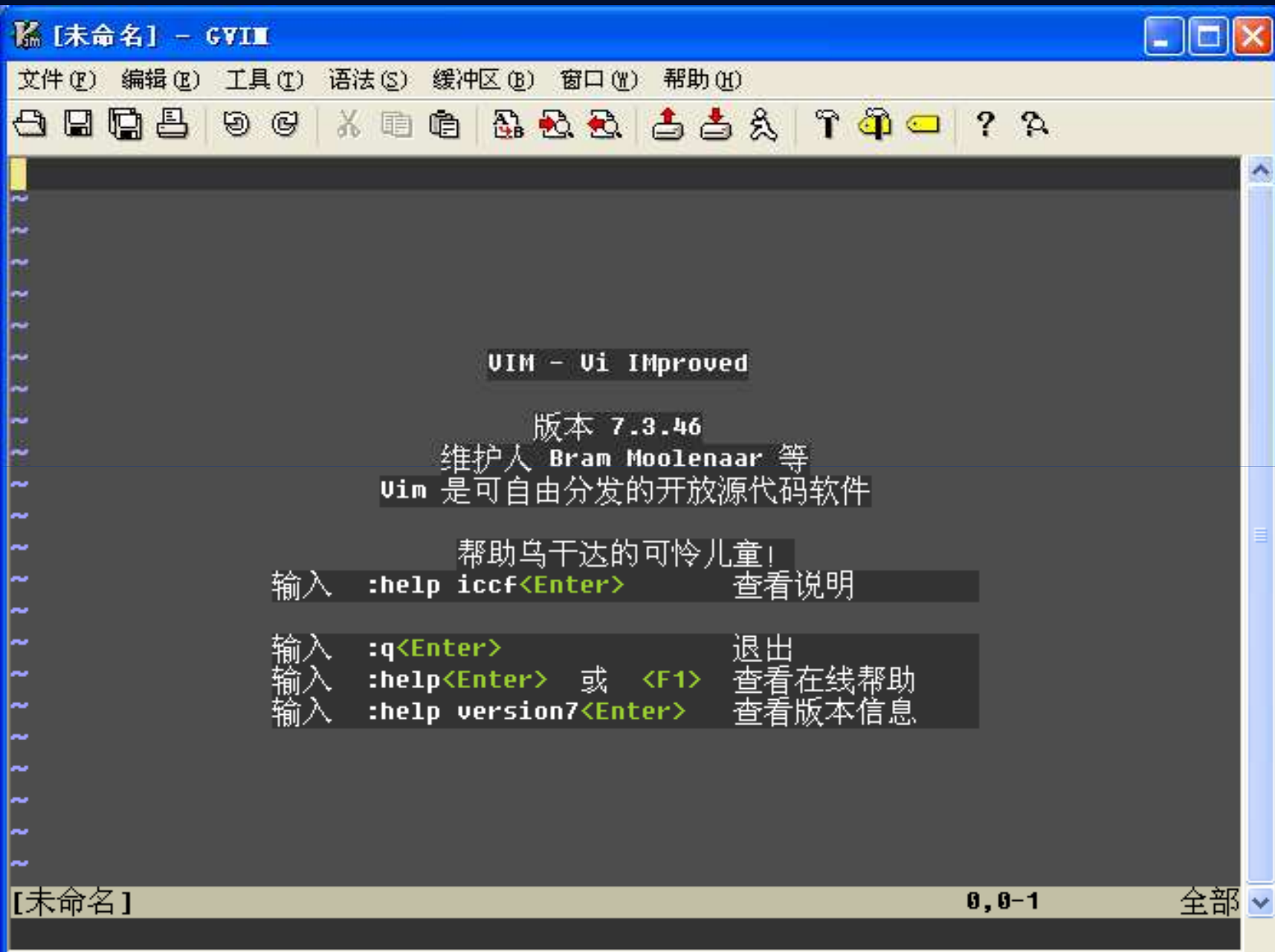


VIM简介

VIM是什么？

- Vim是从vi发展出来的一个文本编辑器。代码补完、编译及错误跳转等方便编程的功能特别丰富，在程序员中被广泛使用。和Emacs并列成为类Unix系统用户最喜欢的编辑器。
- Vim的第一个版本由Bram Moolenaar在1991年发布。最初的简称是Vi IMitation，随着功能的不断增加，正式名称改成了Vi IMproved。现在是在开放源代码方式下发行的自由软件。



BGIDEMO.C (D:\tc2) - GVIM

文件(F) 编辑(E) 工具(T) 语法(S) 缓冲区(B) 窗口(W) 帮助(H)

Press <F1> for Help

11 a BGIDEMO.C
8 a _vimrc
6 h shift.c

<uF List> 2,1 全部

PI
ON
OFF
adjasp
torad
MAXPTS
MaxPts

struct
PTS

variable
Fonts
LineStyle
FillStyle
TextDirect
HorizJust
VertJust
GraphDriver
GraphMode
AspectRatio
MaxX
MaxY
MaxColors
ErrorCode
palette

function
main
Initialize
ReportStatus

```
void CRTModeDemo(void);  
void UserLineStyleDemo(void);  
void FillStyleDemo(void);  
void FillPatternDemo(void);  
void PaletteDemo(void);  
void PolyDemo(void);  
void SayGoodbye(void);  
void Pause(void);  
void MainWindow(char *header);  
void StatusLine(char *msg);  
void DrawBorder(void);  
void changetextstyle(int font, int direction, int c  
int gprintf(int *xloc, int *yloc, char *fmt, ... )  
/*  
/* Begin main function  
/*  
D:\tc2\BGIDEMO.C 99,6 6%  
int x, y;  
getviewsettings( &viewinfo );  
getlinesettings( &lineinfo );  
getfillsettings( &fillinfo );  
gettextsettings( &textinfo );  
getpalette( &palette );  
x = 10;  
y = 4;  
MainWindow( "Status report after InitGraph" );  
settextjustify( LEFT_TEXT, TOP_TEXT );  
driver = getdrivename();  
mode = getmodename(GraphMode); /* get current  
set hidden  
set laststatus=2  
set ignorecase  
set smartcase  
set textwidth=78  
"use ctags -R --c-kinds=+px first  
if has("unix")  
    set tags+="/usr/include/tags  
endif  
"set confirm to popup about unsaved changes  
"set wildmenu?  
"set nomodelines for safety?  
"try set nostartofline?  
if has("unix")  
    set backupdir=~/.tmp  
elseif has("win32")  
    set backupdir=C:\tmp  
endif  
".,~/.tmp,~/  
" Session Options  
set sessionoptions-=curdir  
set sessionoptions+=sesdir,unix,slash  
" GUI Chinese Menu  
if has("unix") && has("gui_running")  
    set langmenu=zh_CN.utf-8  
    source $VIMRUNTIME/delmenu.vim  
    source $VIMRUNTIME/menu.vim  
endif  
" Colorscheme  
if has("gui_running")  
    colorscheme desert  
elseif &t_Co>=256  
    colorscheme tdesert
```

<g_List_ 33,5 20% D:\tc2\BGIDEMO.C 195,3 13% \Program Files\Vim_vimrc 126,6-9 68%

关于Vim的趣价

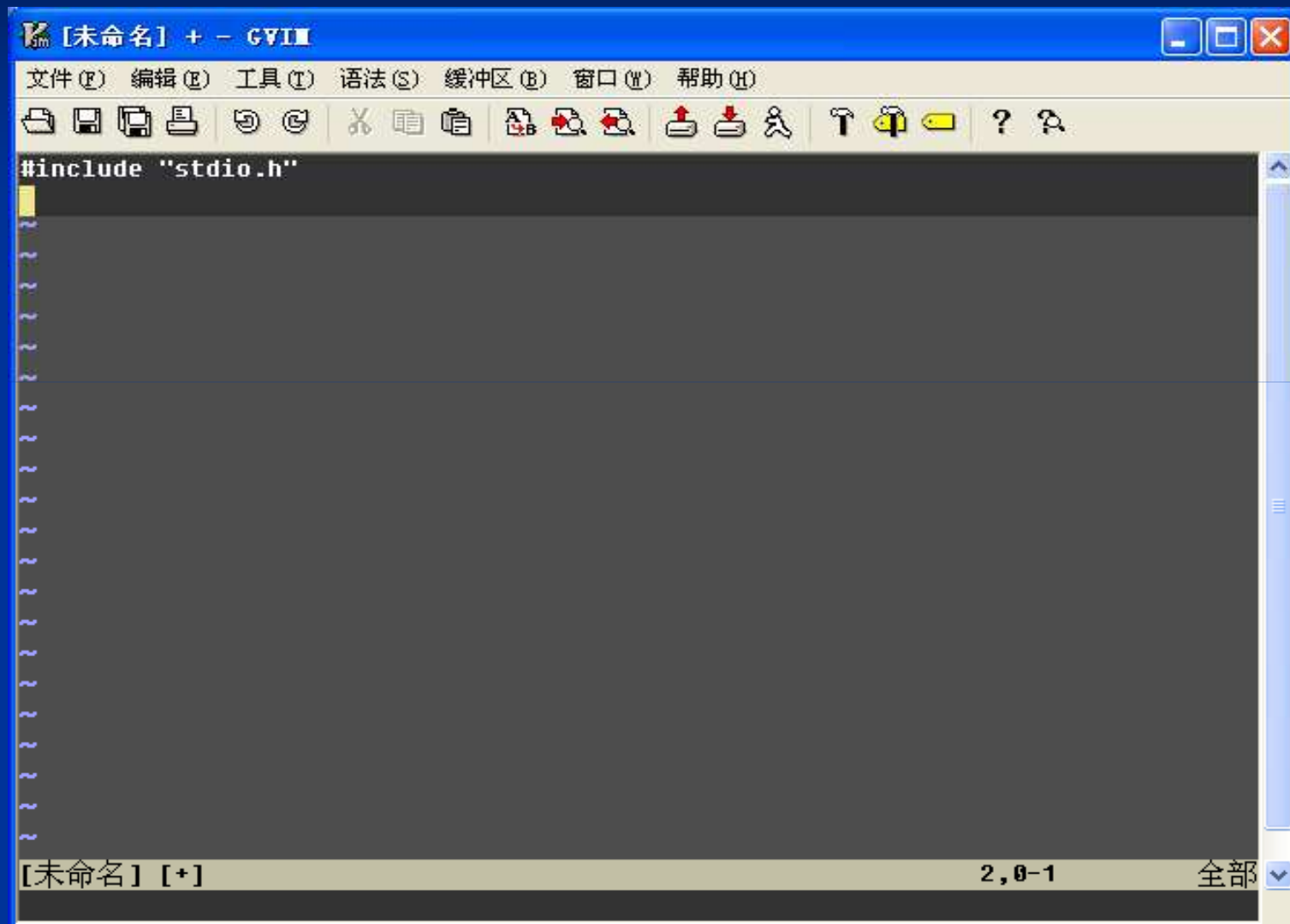
- vim编辑器之神

- emacs神的编辑器

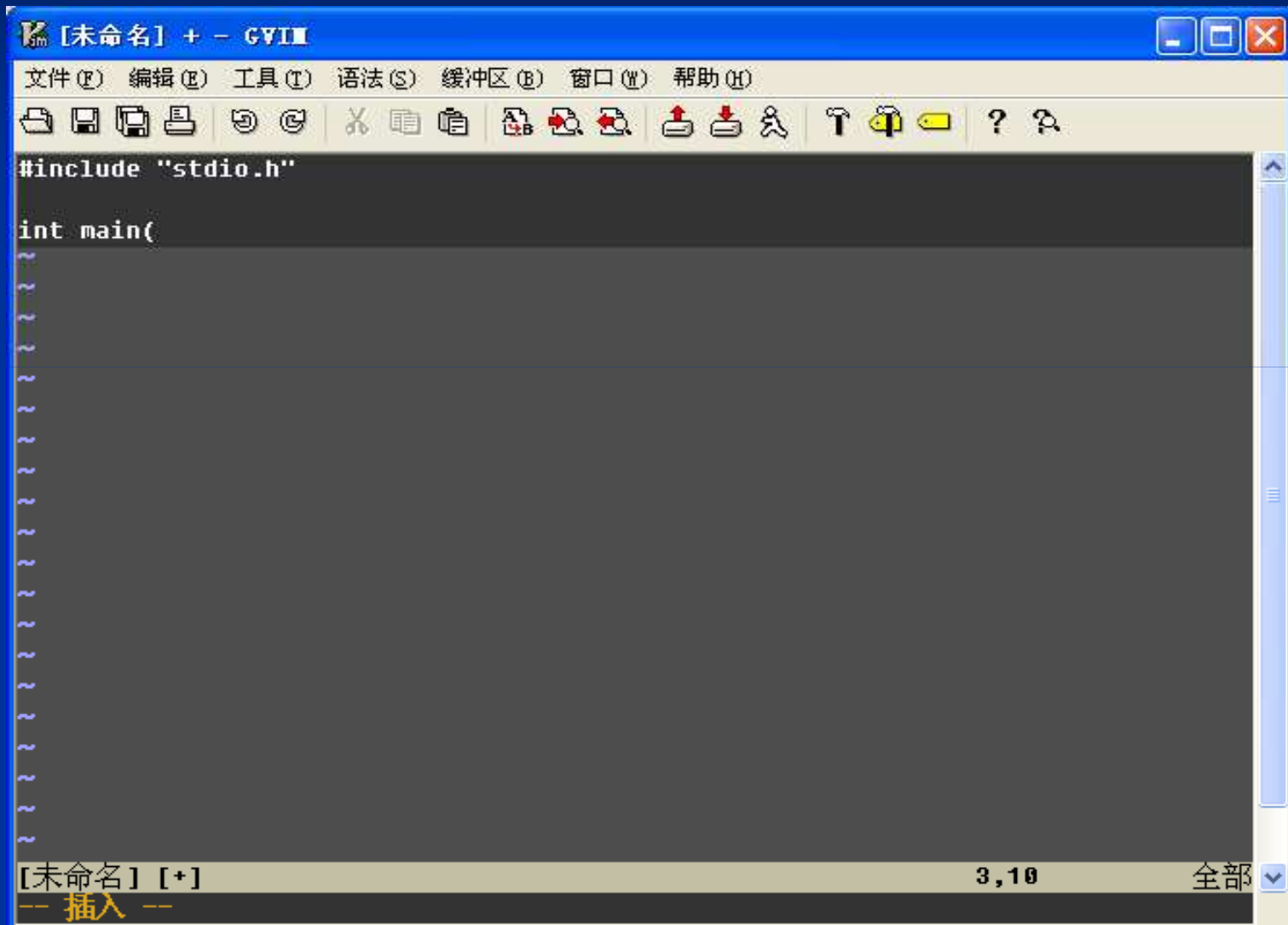
VIM的五个模式

- 正常模式 (Normal Mode)
- 插入模式 (Insert Mode)
- 可视模式 (Visual Mode)
- 命令模式 (Command Mode)
- 替换模式 (Replace Mode)

正常模式 (Normal Mode)



插入模式 (Insert Mode)



可视模式 (Visual Mode)

文件(F) 编辑(E) 工具(T) 语法(S) 缓冲区(B) 窗口(W) 帮助(H)

```

#include "stdio.h"

int main()
{
    printf("Hello World!\n");
    return 0;
}
  
```

[未命名] [+]
 -- 可视 --

命令模式 (Command Mode)

The screenshot shows the GVIM (GNU Vim) editor interface. The title bar at the top reads "[未命名] + - GVIM". Below it is a menu bar with options: 文件(F), 编辑(E), 工具(T), 语法(S), 缓冲区(B), 窗口(W), 帮助(H). A toolbar with various icons follows. The main editing area has a dark background and contains the following code:

```
#include "stdio.h"

int main()
{
    printf("Hello World!\n");
    return 0;
}

~
~
~
~
~
~
~
~
~
~
```

The status bar at the bottom displays "[未命名] [+]", "7,1", and "全部".

模式之间的切换

- “i”，进入插入模式
- “v”，进入可视模式
- “:”，进入命令模式
- 可以使用“ESC”、“Ctrl+C”或者“Ctrl+[”回到正常模式。

Vim的基本操作(1)

- :w 写入文件 :sav 另存为
- :q 退出 :wq 写入并退出
- :q! :wq! 表示强制退出
- :e 编辑文件

Vim的基本操作(2)

- hjkl方式移动光标，代表左下上右。
- “^”到行首，“\$”到行尾。
- Ctrl+F：下翻一页 Ctrl+B：上翻一页
- Ctrl+D：下翻半页 Ctrl+U：上翻半页
- Ctrl+E：下滚一行 Ctrl+Y：上滚一行
- 大写H：光标到最上方 大写M：光标到中间
- 大写L：光标到最下方
- 提示：使用”11k”和”5l”,可以向下移动11行，向右移动5列。

Vim的基本操作(3)

- Vim中的复制、粘贴和删除特别的方便
- yy复制整行，dd删除整行
- yaw复制一个词，daw删除一个词
- 5yy,删除5行
- P粘贴在光标前
- p粘贴在光标后
- 提示：可以先用v选择文本，然后在执行y或d

Vim基本操作（4）

- 丰富的移动功能

- w,W 向后移动到单词开头
- e,E 向后移动到单词末尾
- b,B 向前移动到单词开头
- ge,gE 向前移动到单词末尾
- fx,tx 向后寻找字符x
- Fx,Tx 向前寻找字符x
- gg,G 跳到文件开始、结尾
- xG 跳到文件第x行

Vim基本操作（5）

- 更方便的移动
- % 移动到配对的括号
- } { 向后、向前移动到空行
-]] 移动到下一函数开头、末尾
- [[移动到上一函数开头、末尾
- 注：函数名后的“{”必须写在第一行
- gd 寻找局部变量定义
- gD 寻找全局变量定义

Vim基本操作（6）

- 方便的查找功能

- `/xxx` 向后查找模式xxx
- `?xxx` 向前查找模式xxx
- `*,g*` 向后查找光标所在单词
- `#,g#` 向前查找光标所在单词
- `n ,N` 向后、向前查找下一个

Vim基本操作（6）

- 组合操作
- 上边介绍的操作可以组合起来使用，例如：
- y\$ 从光标处复制到行末
- dt(从光标处删除到(
- ggdG 删除整个文档
- ○ ○ ○ ○

Vim基本操作（7）

- 丰富的替换功能
- r 替换单个字符，退回正常模式
- s 替换单个字符，进入插入模式
- cW 删除一个单词，并进入插入模式
- cib 删除()中的内容，并进入插入模式
- ciB 删除{}中的内容，并进入插入模式
- u 撤销，U 撤销整行，Ctrl+R 逆向撤销

Vim基本操作（8）

- Ctrl +] 标签跳转 Ctrl + O 后退
- Ctrl + T 后退到前一个标签
- Ctrl + I 前进
- gf 跳到文件
- Ctrl + 6 切换到前一个文件

Vim基本操作（9）

- 方便的QuickFix功能
- :make 通过make编译程序
- :cc 显示当前错误信息
- :cn 跳到下一个错误
- :cp 返回到上一个错误
- :cw 打开QuickFix列表
- :ccl 关闭QuickFix列表

Vim基本操作 (10)

- 更多命令
- I gI 在行首插入
- A 在行尾插入
- a 在光标后插入
- o 在下一行插入，并插入空行
- O 在本行插入，并插入空行
- Ctrl + G 显示文本行数和当前行

Vim基本操作(11)

- Vim支持带正则表达式的替换
- :s/pat1/pat2/ 把pat1替换为pat2
- 其后可以跟上参数:
- g:对整行进行替换 c:替换前确认
- n:显示匹配个数
- 若要替换整个文件,则使用:%s/pat1/pat2/gc

VIM的内置功能

- 语法高亮
- 自动缩进
- 自动补全
- 分割窗口
- 代码折叠
- 支持宏、书签、映射和缩写
- 可调颜色配置

我对VIM的印象

- Vim并不是完美的编辑器
- 自由配置方式
- 双手操作比较舒服

改变了编辑方式



改变了编辑方式



改变了编辑方式



VIM的优点

- 开放源代码
- 可编程性强，支持vimscript,Python,Perl等等语言对Vim的操作。
- 插件众多：Pyclewn, Taglist, Zen Coding, ConqueTerm ...
- 对国际化支持:utf-8, GBK(cp936) ...
- 跨平台:Windows Linux AIX Solaris HP-UX System V ...

VIM的缺点

- 学习时间长
- 打开大文件比较慢

入门与参考信息

- 在命令行下输入vimtutor，将会出现vim的基础教程，可以边学边练
- :help xxx,显示xxx的帮助
- www.vim.org vim的官方站点
- wikia.vim.org vim的wiki
- 个人网站：www.vimer.cn
<http://easwy.com/blog/>
- 或者 <http://forum.ubuntu.com.cn>的vim板块

谢谢！