

Vi Improved 简介



edyfox@smth.org (滇狐)

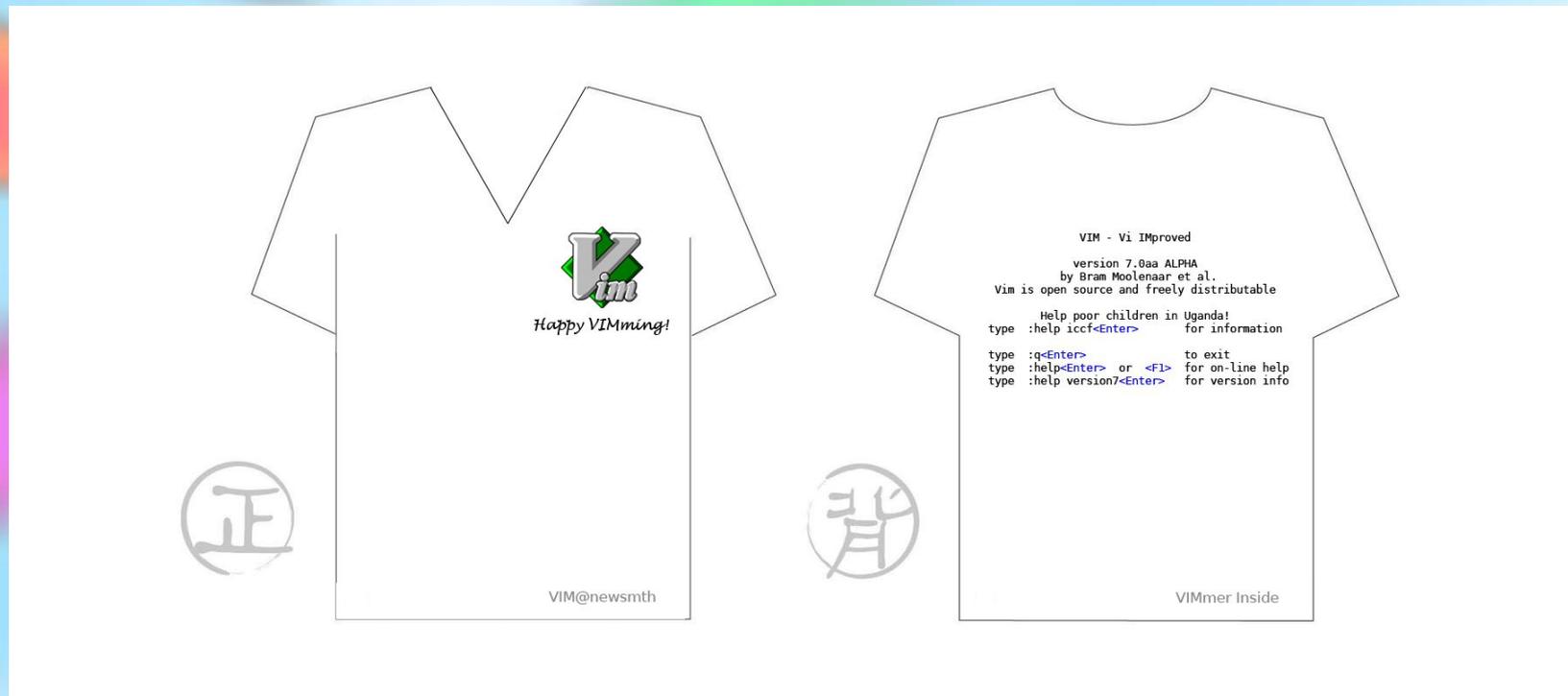
<http://edyfox.codecarver.org>

清华大学软件学院

北京大学 Linux 俱乐部

版衫

- 定价 50 元，多出部分捐给 Bram Moolenaar
- 版衫样图：



主要内容

- VIM 简介
- VIM 入门
- VIM 进阶

主要内容

- **VIM 简介**
 - 什么是 VIM ?
 - 什么人适合使用 VIM ?
 - 我与 VIM
- VIM 入门
- VIM 进阶

vi 与 VIM

- VIM 的发音
 - 根据官方资料，应该读做 Vee Eye Emm
- vi 简介
 - Unix 下的全屏幕可视化编辑器
 - 由于其源代码不可获得，衍生了许多克隆版本
 - nvi 、 elvis 、 vile 、 VIM...
- VIM 即 Vi IMproved ， vi 克隆版本之一

VIM 简介

- 目前最流行的 vi 克隆版本
 - 支持目前绝大部分平台
 - Windows、Unix、Linux、MacOS、DOS...
 - 功能强大，扩展性强
- Bram Moolenaar 编写
 - 1991 年 11 月完成第一个版本
 - 目前最新稳定版本为 6.3
 - 慈善软件

慈善软件

- 不强制注册
 - 即使不通过注册，也能使用 VIM 全部功能
- 注册用户可以选择想要的功能
 - 个人家庭使用注册费 10 欧元
 - 用于专业领域编辑工作注册费 30 欧元
- 缴纳的注册费将用于救助乌干达的可怜儿童
 - 主要用于救助艾滋病肆虐的地区失去父母的孤儿

历年奖项

- Linux Journal Readers' Choice Awards

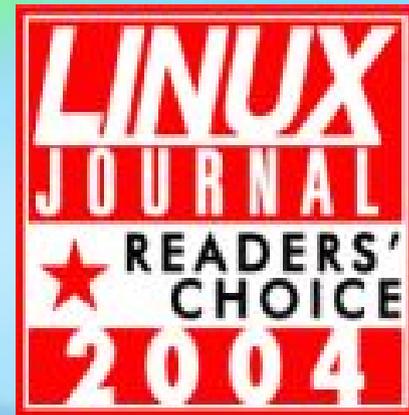
- 2004

- 2003

- 2002

- 2001

- 2000 : 未将 VIM 与 vi 分开授奖



什么人适合使用 **VIM** ？

- 追求编辑效率的程序员
 - VIM 的作者是程序员
 - 他为程序员开发了许多有用的功能
- 其它经常处理纯文本的人员
 - 中小型企业 Web 开发人员
 - 编写 LaTeX 文档的研究人员
 - Unix / Linux 系统管理员

什么人不适合使用 **VIM** ？

- 生活、工作和学习无需和纯文本打交道的人
 - 处理其它类型的文件有专门的工具
- 打字速度慢于智能感知运行速度的人
 - **VIM** 目前暂不支持智能感知
- 被强大的 **IDE** 惯坏了的人
 - 强大顺手的 **IDE** 比方便快捷的编辑器更重要

我与 VIM

- 2000 年首次使用 elvis (??)
 - 传说中的 XTeam Lindows
 - 新奇而古怪
 - 没报刊杂志上说的那么疼
- 2003 年 3 月接触 Solaris ， 使用 VI 约一周
 - 明白了 VI 与 VIM 原来是不一样的
 - 改用 VIM
 - 仅能完成简单编辑

我与 VIM

- “非典”期间，沉迷 Solaris
 - 每天用 VIM 写日记
 - 一度怀疑 VIM 是否真能“方便快捷地编辑文件”
- 之后一年，逐渐能够在 VIM 下编写短小程序
 - 日益熟练，但仍不能胜任复杂编辑
 - 尚不能离开 Visual Studio 环境
 - 在部分 Windows 机器上安装了 VIM，目的是耍酷

我与 VIM

- 2004 年 7 月山东出差期间，研究 VIM 手册
 - 一周时间将 VIM 用户手册通读了一遍
 - 出差回来后发现其它所有编辑器都不好用
 - “被迫”在所有机器上装了 VIM
- 2004 年 8 月，收集各种插件
 - 每天搜索、下载、安装、使用、卸载……
 - 淘到不少好用的插件

我与 VIM

- 2004 年 9 月，企图使用 VIM 作为邮件编辑器
 - FoxMail 无法内嵌 VIM
 - 安装 mutt for Windows，不支持中文
 - 放弃 Windows 平台
- 2004 年 10 月，逐渐习惯了 Linux 下的生活
- 2005 年 2 月至今，逐渐熟悉少数较生僻功能
 - 在 BBS 为冒充高手硬着头皮学了不少功能

主要内容

- VIM 简介
- VIM 入门
 - VIM 基本操作
 - VIM 快速编辑示例
 - 编写自己的 .vimrc
- VIM 进阶

屏蔽系统自带的 **vimrc**

- Linux 用户
 - 创建 ~/.vimrc 文件
- Windows 用户
 - 在个人主目录下创建空文本文件 _vimrc
 - 注意显示扩展名问题

Linux 下中文处理

- 编辑时不能输入中文
 - `:set encoding=cp936`
- 菜单显示是乱码
 - `:so $VIMRUNTIME/delmenu.vim`
 - `:so $VIMRUNTIME/menu.vim`

选择喜欢的字体

- 通过菜单与对话框选择字体
 - 编辑 --> 选择字体
 - 只有等宽字体可以被选中
 - 适合程序员使用的字体基本原则
 - Bitstream Vera Sans Mono
 - 00o 1Il
- 记住选中的字体名称
 - `:set guifont?`
 - 将显示结果抄下来备用

写入 .vimrc

- VIM 不能自动记住你所做的配置修改
- 需要手工写入 .vimrc
 - 双引号开头的行表示注释
- 你现在的 .vimrc 也许是这个样子的：

```
" Windows 下建议也加上这句
set encoding=cp936

" 这两句只在 Linux 下需要
so $VIMRUNTIME/delmenu.vim
so $VIMRUNTIME/menu.vim

" 设置字体。
" 字体名称可以通过前一节介绍的语句获得，
" 如果字体名称中含有空格，
" 需要在空格前面加上一个反斜杠，如下：
set guifont=Courier\ 9
```

VIM 基本操作

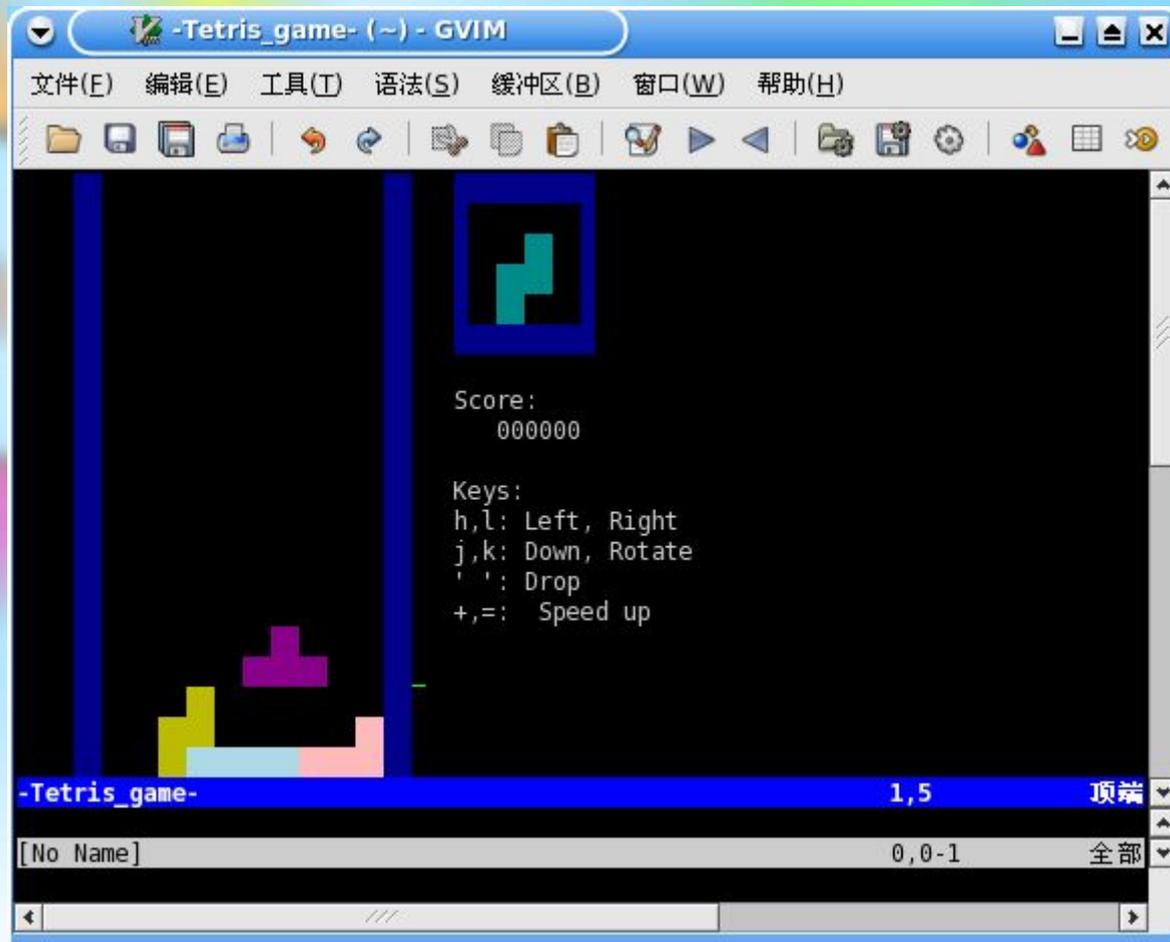
- VIM 是一个有模式的编辑器
 - Normal 模式下按键序列对应命令
 - 按 i 键进入 Insert 模式
 - 按 : 进入 command 模式
 - Insert 模式下可以输入文本
 - 按 ESC 回到 Normal 模式
 - Command 模式下输入命令
- 有模式导致 VIM 快捷键长度较短

使用 **hjkl**

- VIM 支持使用方向键移动光标
 - 在 insert 状态下移动光标会比较方便
- 使用 **hjkl** 移动光标编辑效率更高
 - 距离近，按键方便，眼睛无需离开屏幕

俄罗斯方块

- http://vim.sf.net/scripts/script.php?script_id=172



vimtutor

- vimtutor 是 VIM 入门非常好的学习材料
- Linux 用户
 - 在终端中输入 vimtutor
 - 启动字符界面的 vimtutor
- Windows 用户
 - 在开始菜单中找 vimtutor 快捷方式
 - 启动后出现乱码，重新选择一个中文字体即可

快速编辑示例

- 不再详细 VIM 基本操作
 - 时间所限，对 VIM 的介绍不可能面面俱到
 - 水平所限，所讲内容不可能强过 vimtutor
- 介绍几个 VIM 快速编辑示例
 - 对于老用户，进一步增强记忆
 - 对于新用户，坚定学习 VIM 的决心

查找替换

- 准备工作
 - 打开 VIM，按 i，输入：
 - `this is a test`
- 查找替换
 - 按 ESC 回到 normal 状态，输入：
 - `:%s/ /^M/g`
 - 其中 ^M 使用 Ctrl-v Ctrl-m 的方式输入
 - 将所有空格替换为回车，文本变为 4 行

行的拼接

- 查找 `\n` 替换为空格可以实现拼接
- 这里使用 `j` 命令拼接行
 - 按 `ggVG`，全选
 - 按 `:`，此时命令行显示为：
 - `:'<,'>`
 - 在其后输入 `j`，回车
 - 拼接完毕

复制粘贴与重复动作

- 在 normal 状态下按 yy ， 复制当前行
- 按 12p ， 屏幕上即出现 13 行上述字样
 - 12 表示将其后的命令执行 12 次
 - p 表示粘贴到当前行之后
 - VIM 里大部分命令都可以加上数字表明执行次数

列操作

- 选中全文第一列
 - 按 `gg` 跳到开头
 - 按 `Ctrl-v` 进入列选择状态
 - 按 `G` 跳到末尾
- 将每行第一列变为大写
 - 按 `U` 表示把选中内容变大写
 - 按 `u` 表示把选中内容变小写
 - 按 `~` 表示翻转原有大小写

列操作

- 在每行前面加一个星号和一个空格
 - 选中第一列
 - 按 I 进入插入状态
 - 输入 * ，按 ESC
- 选中前两列后按 x ，可以进行列删除
 - 编程时可以用于批量注释和批量解注释

宏的录制

- 将所有偶数行的 a 改成 another
- 录制宏
 - 按 gg 进入第一行
 - 按 qm 进入宏录制状态，录制到寄存器 m
 - 按 j 进入第二行，按 \$ 跳到行末
 - 按 2b ，光标停在字母 a 上
 - 按 caw ，输入 another ，按 ESC ，按 j 进入下一行
 - 按 q 结束宏的录制

宏的播放

- 单次播放
 - 按下 @m ，执行 m 寄存器中的宏
 - 可以发现第 4 行的 a 也变成了 another
- 多次播放
 - 输入 1000@m
 - 可以发现所有偶数行的 a 都变成了 another
 - 到达文件底部时宏执行出错，会自动停止
 - 最终并没有真的执行 1000 次

点命令

- 在每行末尾添加一个感叹号
- 执行次数不多的简单操作无需使用宏
 - 按 gg 进入第一行，按 A 进入行尾添加状态
 - 输入 !，按 ESC，按 j 进入第二行
 - 按 . 表示重复前一个操作，即行尾添加 !
 - 反复按 jj.jj.
 - 推荐大家左手鼠标
- :%s/\$/!/g

VIM 入门

- 21 天学习计划
 - 多读文档，少走弯路
- 第一周：每天一遍 vimtutor
 - 熟悉基本操作，做到随心所欲
- 第二周、第三周：通读用户手册
 - 读用户手册的同时修订出适合自己的 .vimrc
 - 目前手册有中文版译本可供下载

.vimrc for Beginners

- 基本设置

- `set nocp` - 设置不兼容模式
- `set ru` - 打开状态栏标尺
- `set hls` - 搜索时高亮显示被找到的文本
- `set is` - 递进搜索
- `syntax on` - 打开关键字上色
- `set backspace=indent,eol,start`
- `set whichwrap=b,s,<,>,[,]`
 - `h,l,~`

编码设置

- 一段可以自动猜测编码的脚本，很强的

```
" Encoding settings
if has("multi_byte")
    " Set fileencoding priority
    if getfsize(expand("%")) > 0
        set fileencodings=ucs-bom,utf-8,cp936,big5,euc-jp,euc-kr,latin1
    else
        set fileencodings=cp936,big5,euc-jp,euc-kr,latin1
    endif

" CJK environment detection and corresponding setting
if v:lang =~ "^zh_CN"
    " Use cp936 to support GBK, euc-cn == gb2312
    set encoding=cp936
    set termencoding=cp936
    set fileencoding=cp936
elseif v:lang =~ "^zh_TW"
    " cp950, big5 or euc-tw
    " Are they equal to each other?
    set encoding=big5
    set termencoding=big5
    set fileencoding=big5
```

编码设置

- 续前页

```
elseif v:lang =~ "^ko"  
    " Copied from someone's dotfile, untested  
    set encoding=euc-kr  
    set termencoding=euc-kr  
    set fileencoding=euc-kr  
elseif v:lang =~ "^ja_JP"  
    " Copied from someone's dotfile, unteste  
    set encoding=euc-jp  
    set termencoding=euc-jp  
    set fileencoding=euc-jp  
endif  
" Detect UTF-8 locale, and replace CJK setting if needed  
if v:lang =~ "utf8$" || v:lang =~ "UTF-8$"  
    set encoding=utf-8  
    set termencoding=utf-8  
    set fileencoding=utf-8  
endif  
else  
    echoerr "Sorry, this version of (g)vim was not compiled with multi_byte"  
endif
```

.vimrc for Beginners

- 文本编辑设置

- `set sw=4` - 缩进尺寸 4 个空格
- `set ts=4` - Tab 宽度为 4 个字符
- `set et` - 将所有 Tab 替换为空格

.vimrc for Beginners

- C/C++ 编码设置
 - `set cin` - 自动缩进
 - `set ai` - 还是自动缩进
 - `set cino=:0g0t0(sus`
 - `set sm` - 显示括号匹配
- 在 console 下粘贴的版式问题
 - “`:set paste`”、“`set nopaste`”
 - “`=`” 命令重调格式

.vimrc for Beginners

- 断行设置
 - `set lbr` - 不在单词中间断行
 - `set fo+=mB` - 打开断行模块对亚洲语言支持
- 其它设置
 - `set selectmode=`
 - `set mousemodel=popup`
 - `set keymodel=`
 - `set selection=inclusive`

.vimrc for Beginners

- 图形界面与字符界面分别设置
 - `if (has("gui_running"))` - 图形界面设置
 - `set nowrap` - 不折行
 - `set guioptions+=b` - 水平滚动条
 - `colo torte` - 配色方案
 - `else` - 字符界面设置
 - `set wrap` - 折行
 - `colo ron` - 配色方案
 - `endif`

主要内容

- VIM 简介
- VIM 入门
- VIM 进阶
 - VIM 常用插件简介
 - 在 VIM 中编写程序
 - 在 VIM 中编写 LaTeX 文档

插件安装路径

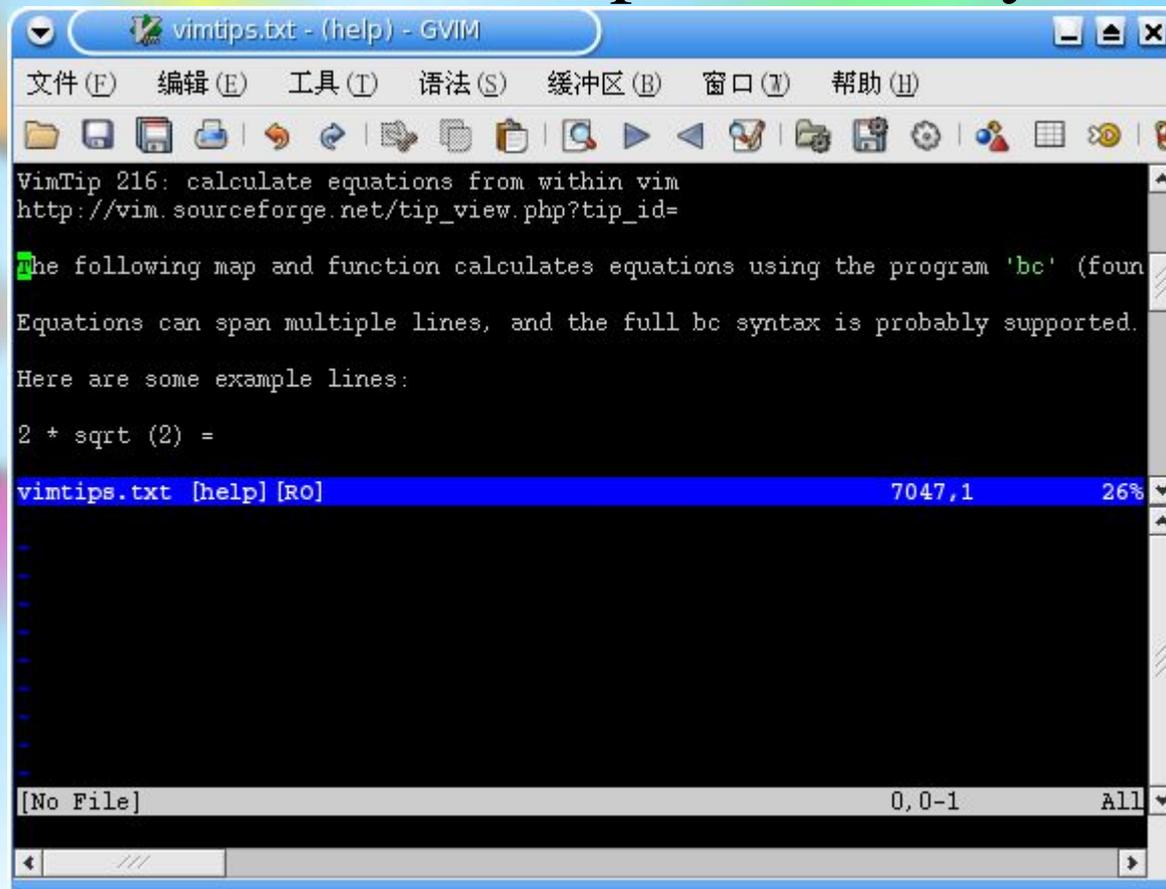
- `$HOME/.vim` 或 `%HOME%/vimfiles`
- `$VIM/vimfiles`
- `$VIMRUNTIME`
- `$HOME/.vim/after` 或 `%HOME%/vimfiles/after`
- `$VIM/vimfiles/after`

插件目录结构

- colors
- compiler
- doc
- etc
- ftdetect
- ftplugin
- indent
- keymap
- plugin
- syntax

常用插件推荐

- 学习 VIM 的帮手—— Tip of the Day



```
vimtips.txt - (help) - GVIM
文件(F) 编辑(E) 工具(T) 语法(S) 缓冲区(B) 窗口(W) 帮助(H)
VimTip 216: calculate equations from within vim
http://vim.sourceforge.net/tip_view.php?tip_id=
The following map and function calculates equations using the program 'bc' (found
in the system path).
Equations can span multiple lines, and the full bc syntax is probably supported.
Here are some example lines:
2 * sqrt (2) =
vimtips.txt [help] [RO] 7047,1 26%
[No File] 0, 0-1 All
```

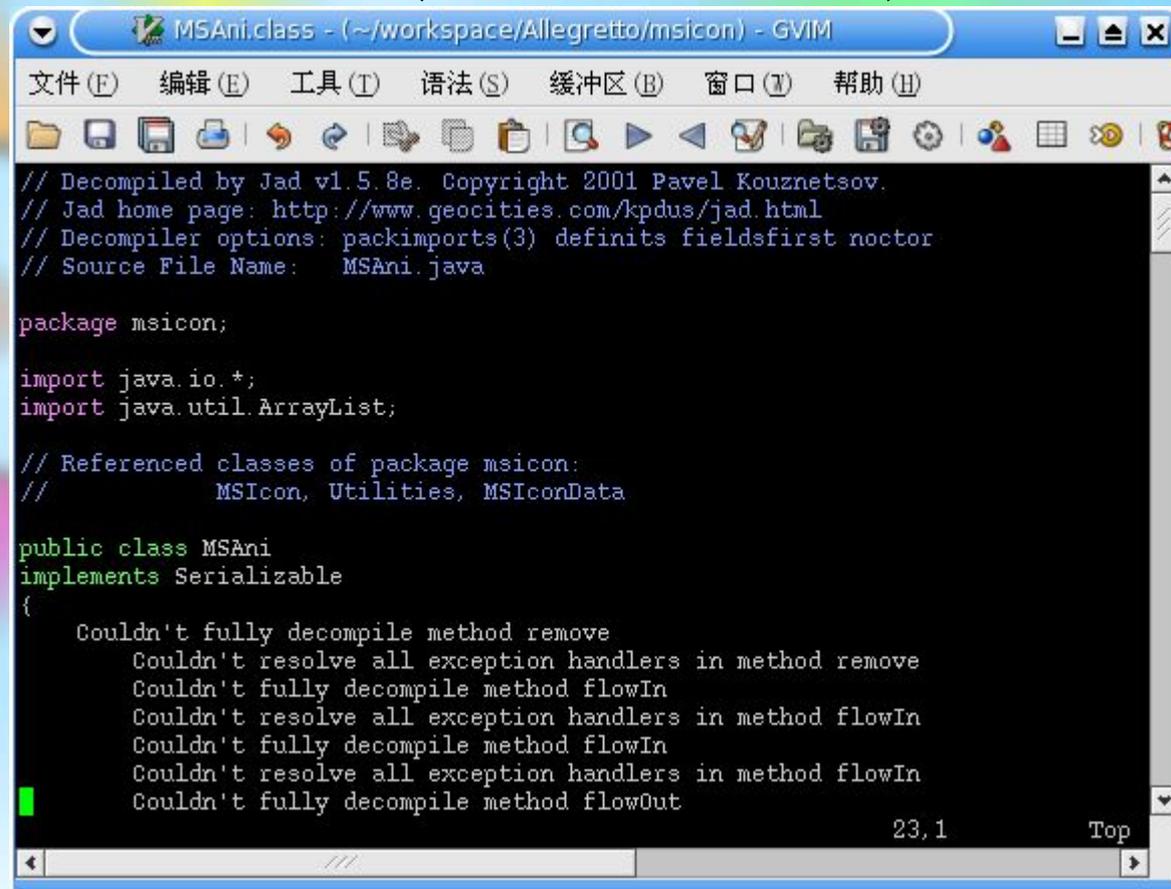
常用插件推荐

- 日程安排工具 Calendar



常用插件推荐

- Java 反编译插件 (需安装 JAD)



```
MSAni.class - (~/.workspace/Allegretto/msicon) - GVIM
文件(E) 编辑(E) 工具(T) 语法(S) 缓冲区(B) 窗口(W) 帮助(H)
// Decompiled by Jad v1.5.8e. Copyright 2001 Pavel Kouznetsov.
// Jad home page: http://www.geocities.com/kpdus/jad.html
// Decompiler options: packimports(3) definits fieldsfirst noctor
// Source File Name:  MSAni.java

package msicon;

import java.io.*;
import java.util.ArrayList;

// Referenced classes of package msicon:
//      MSIcon, Utilities, MSIconData

public class MSAni
implements Serializable
{
    Couldn't fully decompile method remove
    Couldn't resolve all exception handlers in method remove
    Couldn't fully decompile method flowIn
    Couldn't resolve all exception handlers in method flowIn
    Couldn't fully decompile method flowIn
    Couldn't resolve all exception handlers in method flowIn
    Couldn't fully decompile method flowOut
}
23,1 Top
```

常用插件推荐

- 涂鸦插件 Sketch



常用插件推荐

- 一系列插件实现的 VIM IDE
 - 窗口管理器 + 文件浏览器（窗口管理器插件）
 - Tlist + ctags
 - VIM + ctags + cscope
 - ctags、cscope 简介



- doc/
- in/
- out/
- res/
- utils/
- filelist.txt
- in_kmk.kmk
- in_kmk.mak
- in_vcproj.kmk
- in_vcproj.mak
- kittie.cpp
- kittie.kmk
- kittie.mak
- kittie.rc
- out_kmk.kmk
- out_kmk.mak
- out_mak.kmk
- out_mak.mak
- out_txt.kmk
- out_txt.mak
- out_vcproj.kmk
- out_vcproj.mak
- tags
- util.cpp
- kittie.h
- util.h
- version.h

```

// 获取第一条错误信息。
// 如果错误信息不存在，则返回 NULL。
in_plugin::handle_type in_plugin_base::get_first_error(void)
{
    ~/c cpp/kittie/in/in plugin base.cpp [Preview] 555,26 78%
    show_version();
    return 0;
}

if (!initialize())
    return 1;

for (std::list<std::string>::iterator ite = _in_args.begin();
     ite != _in_args.end(); ++ite)
    _in_plg->set_argument(ite->c_str());

// 读入输入文件
bool result = _in_plg->read(_in_file.c_str());

// 输出错误信息
in_plugin::handle_type in_handle = _in_plg->get_first_error();
while (in_handle)
{
    std::cerr << _in_plg->get_error_filename(in_handle) << " ";

    int line_number = _in_plg->get_error_line_number(in_handle);
    if (line_number != -1)
        std::cerr << line_number << " ";
    else
        std::cerr << " ";

    if (_in_plg->get_error_level(in_handle) == 0)
        std::cerr << "warning: ";
    else
        std::cerr << "error: ";

    std::cerr << _in_plg->get_error_message(in_handle) << std::endl;
}

```

```

free_output_plugin [ki
load_input_plugin [ki
load_output_plugin [ki
main

in_plugin_base.cpp (/home/
macro
REGISTER
UNREGISTER
SHARED_PREFIX

function
in_plugin_base [in_plu
~in_plugin_base [in_pl
get_default_filename [
get_first_configuratio
get_next_configuration
get_config_name [in_pl
get_config [in_plugin_
get_root [in_plugin_ba
get_first_sub [in_plug
get_next_sub [in_plugi
get_folder_name [in_pl
get_folder_filter [in_
get_first_file [in_plu
get_next_file [in_plug
get_file_name [in_plug
get_customized_build [
get_output_filename [i
get_description [in_pl
get_first_dependency [
get_next_dependency [i
get_dependency [in_plu
get_first_command_line
get_next_command_line
get_command_line [in_p
get first error [in pl

```

[File List] 19,4 All ~/.c_cpp/kittie/kittie.cpp 142,55 21% Tag List 10,5 44%

```

:|kittiemake 2>&1| tee /tmp/v426725/8
----- 生成配置项：in_kmk.so - Debug -----
链接...
----- 生成配置项：in_vcproj.so - Debug -----
编译 in_vcproj/in_vcproj.cpp
链接...
----- 生成配置项：kittie - Debug -----
编译 kittie.cpp

```

常用插件推荐

- ViKi + Deplate
 - 类似 WiKi 的语法
 - 用于制作基于内容的简单个人主页
 - 滇狐的个人主页就是使用 ViKi 制作的
- 优点
 - 使用方便，功能比较强大
- 缺点
 - 配置复杂，依赖项太多

在 **VIM** 下编写程序

- 正确地编写 Makefile
- 设置合适的 makeprg
 - 例如在有的系统下你也许想用 gmake
- 在 Windows 下最好安装一份 tee
 - 能够看到编译过程
 - `:set shellpipe=2>&1\|\ tee`
- 根据编译器输出正确设置 errorformat

在 **VIM** 下编写程序

- 常见编译器可通过 VIM 自带脚本设置参数
 - `:compiler xxx`
 - 菜单项: 工具 --> Set compiler
- 使用 `:make` 编译程序
 - `:make`
 - `:make clean`
- 使用 `:copen` 查看错误列表

copen 小技巧

- 在多个文件中搜索指定字符串
 - 将搜索结果列在一个 `buffer` 中
 - 可以条目跳转到相应文件的相应位置
 - `copen` 具备这样的能力
- 需求
 - 在文件中检索指定字符串的工具
 - 在检索结果给出文件名和行号信息
 - 合适的 `errorformat` 设置

copen 小技巧

- 设置 grep 参数
 - `:set grepprg=grep\ -nH\ $*`
 - 建议写到 `.vimrc` 中
- 使用 grep 命令查找文件
 - `:grep "test string 2" *`
 - `:grep -R "test string 2"`
 - `:grep "function" *.cpp`

在 **VIM** 下编写 **LaTeX** 文档

- 安装 LaTeX Suite
- 配置正向、反向搜索
 - 正向搜索
 - 在生成的 dvi 文件中查找源代码所对应元素
 - 反向搜索
 - 在生成的 dvi 文件中查找元素所对应源代码的位置。
- 初学者关于 `<++>` 的困惑
 - 是 LaTeX Suite 一项非常好用的功能

主要内容

- VIM 简介
- VIM 入门
- VIM 进阶

学习资源

- VIM 网站
 - <http://www.vim.org>
 - <http://vim.sourceforge.org>
- 中文站点
 - 水木社区 VIM 版及版面附属 FTP

学习资源

- VIM 邮件列表
 - vim-announce@vim.org
 - vim@vim.org
 - vim-dev@vim.org
 - vim-mac@vim.org
 - vim-multibyte@vim.org

VIM 使用小结

- VIM 是非常强大而方便的文本编辑器
 - 需要通过学习才能够用好
 - 学习付出的努力是值得的
- 不要抱怨 VIM 作为 IDE 不够强大
 - 先想想自己为所谓“强大”的 IDE 付了多少钱
 - 开源社区的项目需要所有人为之付出劳动

谢谢大家！



感谢大家支持！

Q & A

Happy VIMming~~