

背景：很多同学在Linux开发比较久，对可视化依赖比较强，针对以上情况分享一个vim插件组合，安装比较简单，有些开源的插件我做了改进：

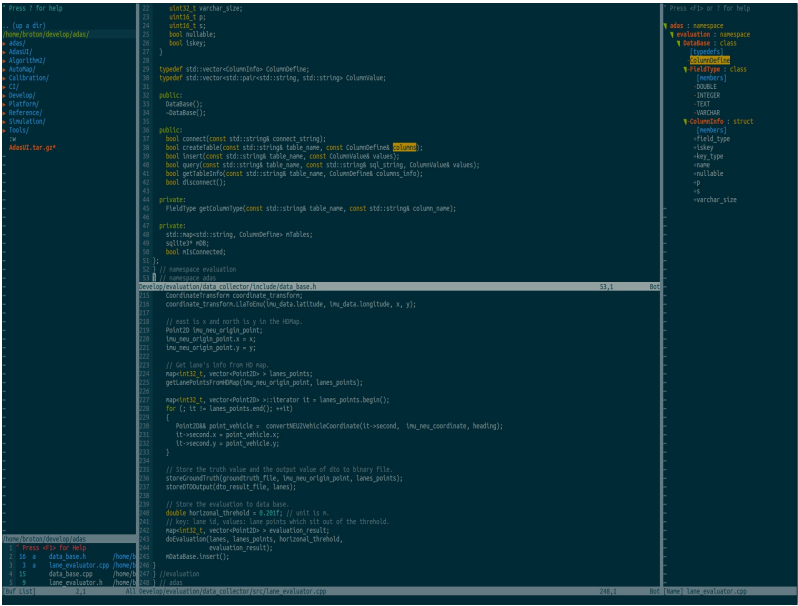
- 1 source tree view 工具nerd tree 增加了跳转当前文件所在的目录的功能
- 2 改进了visual bookmark(可视化书签，vim自带的书签不可视化，并且不能在关闭文件后保存)，增加了书签列表显示和选择功能
- 3 增加session save 功能
- 4 自己写了个工程配置管理功能插件
 - 1) 可以自动为目录生成intelligence，支持java js c++/c
 - 2) 可以在工程上查找并打开某个文件功能
 - 3) 手动给某个目录生成intelligence功能

• 安装步骤

- 1. sudo apt-get install ctags //安装ctags
- 2. git clone https://github.com/itfootman/vim-plugins ~/.vim (如果已经有.vim目录请备份)
- 3. cp ~/.vim/vimrc ~/.vimrc (如果原来有.vimrc请备份)
- 4. cd ~/.vim/bundle/command-t/ruby/command-t
- 5. ruby extconf.rb (如果没有ruby请安装ruby, sudo apt-get install ruby)
- 6. make
- 7. cp ~/.vim/projectcfg \$(your project root)/.projectcfg (.projectcfg所在的目录会被插件管理成一个工程)
- 8. vi \$(your project root)/.projectcfg 修改tagIncludeFolders的值为tagIncludeFolders=.:java|cplusplus
- 9. 如果要保证shift + F4快速可用，最好安装一个konsole, gnome terminal有bug shift + F4会转成奇怪的字符，安装konsole, sudo apt-get install konsole

• 常用方法，以下快捷键，均可在~/.vimrc下自行配置

- 1) F3 显示nerdtree (相当于vs的工程文件浏览)，F4显示taglist 相当于vs的 (class view), 左下角的是打开的文件的list (buffer list and mru) 到nerdtree下连续按 cw 就会切换到当前代码文件的目录下，并列出来该目录的其它文件，这个是自己实现的功能，非开源插件功能



- 2) 打开工程中的某个文件，按F3 F4关闭nerdtree和taglist, (如果不关闭打开的文件会在这两个窗口显示，这是一个bug 暂时还没修复) 输入的命令为: CommandT, 等待片刻，第一次导入文件列表，会要1到2秒钟，出来文件列表后在最下面的GoToFile可以输入自己想查找的源文件

```
53 } // namespace adas
Develop/evaluation/data_collector/include/data base.h
248 } // adas
Develop/evaluation/data_collector/src/lane_evaluator.cpp
Perception/LocalMap/Clustering/Clustering/ClusterFlex.h
Perception/LocalMap/Clustering/Clustering/ClusterFlex.cpp
Perception/LocalMap/Clustering/ClusteringCuda/ClusterFlex.h
Perception/LocalMap/Clustering/ClusteringCuda/ClusterFlex.cpp
Perception/LocalMap/Clustering/ClusteringCuda/PnClusterFlex.h
Perception/LocalMap/Clustering/ClusteringCuda/PnClusterFlex.cpp
Perception/LocalMap/PnCTracking/PnCTracking/PnClusterFlex.h
Perception/LocalMap/PnCTracking/PnCTracking/PnClusterFlex.cpp
Perception/LocalMap/Clustering/Clustering/PlaneFilter.h
Perception/LocalMap/PnCTracking/PnCTracking/PnClusterFlex.cpp
Perception/LocalMap/Clustering/Clustering/PlaneFilter.cpp
Perception/LocalMap/Components/VelodyneClustering/nanoflann.hpp
Perception/LocalMap/Common/Interfaces/LidarClusterInterface/LidarClusterClient.h
Perception/LocalMap/Common/Interfaces/LidarClusterInterface/LidarClusterCommon.h
Perception/LocalMap/Common/Interfaces/LidarClusterInterface/LidarClusterServer.h
Perception/LocalMap/Common/Interfaces/LidarClusterInterface/LidarClusterClient.cpp
Perception/LocalMap/Common/Interfaces/LidarClusterInterface/LidarClusterServer.cpp
Perception/LocalMap/Common/Interfaces/clusteredSickInterface/clusteredSickInterface.h
Perception/LocalMap/Common/Interfaces/clusteredSickInterface/clusteredSickInterface.cpp
Perception/LocalMap/Common/Interfaces/VelodyneClusterInterface/VelodyneClusterInterface.h
Perception/LocalMap/Common/Interfaces/VelodyneClusterInterface/VelodyneClusterInterface.cpp
Perception/LocalMap/Common/Interfaces/VelodyneClusterInterface/VelodyneClusterInterface.cpp
Framework/nucleus/include/nucleus-comm/protocol/HeaderFlyweight.h
Perception/LocalMap/Common/Interfaces/fusion/WriterLogFile.h
Framework/nucleus/include/nucleus-comm/protocol/DataHeaderFlyweight.h
build_debug/Framework/nucleus/rate_subscriber/CMakeFiles/rate_subscriber.dir/flags.make
Framework/nucleus/build/rate_subscriber/CMakeFiles/rate_subscriber.dir/flags.make
Framework/nucleus/build/examples/CMakeFiles/multi_subscriber.dir/flags.make
build_debug/Framework/nucleus/examples/CMakeFiles/multi_subscriber.dir/flags.make
Framework/nucleus/include/nucleus-comm/command/PublicationBuffersReadyFlyweight.h
build_debug/Perception/CMakeFiles/laneproducerio.dir/Digest/Recognize/LDW/tracker/LDW_LaneFilter.cpp.o
build_debug/Perception/CMakeFiles/laneproducerio.dir/Digest/Recognize/LDW/tracker/LDW_WidthFilter.cpp.o
build_debug/Perception/CMakeFiles/laneproducerio.dir/Digest/Recognize/LDW/tracker/LDW_3rdOrderLaneFilter.cpp.o
Framework/nucleus/build/CMakeFiles/gen_properties.dir/flags.make
build_debug/Perception/CMakeFiles/laneproducerio.dir/Digest/Recognize/LDW/tracker_multilane/LDW_MultilaneFilter.cpp.o
build_debug/Perception/CMakeFiles/laneproducerio.dir/Digest/Recognize/LDW/tracker_multilane/LDW_3rdOrderMultilaneFilter.cpp.o
build_debug/Framework/nucleus/rate_subscriber/CMakeFiles/progress.marks
build_debug/Framework/nucleus/CMakeFiles/gen_properties.dir/flags.make
Framework/nucleus/build/rate_subscriber/CMakeFiles/progress.marks
build_debug/Framework/nucleus/rate_subscriber/CMakeFiles/rate_subscriber.dir/link.txt
build_debug/Framework/nucleus/rate_subscriber/CMakeFiles/rate_subscriber.dir/build.make
build_debug/Framework/nucleus/rate_subscriber/CMakeFiles/CMakeDirectoryInformation.cmake
build_debug/Framework/nucleus/rate_subscriber/CMakeFiles/rate_subscriber.dir/depend.make
build_debug/Framework/nucleus/rate_subscriber/CMakeFiles/rate_subscriber.dir/progress.marks
```



```
204 evaluateLaneLine(left_line, LEFT_LANE, dto_lanes_points, result);
205
206 return true;
207 }
208 }
209
210 void LaneEvaluator::evaluate(const ImuData& imu_data, const Lanes& lanes)
211 {
212     double x = 0.0;
213     double y = 0.0;
214
215     CoordinateTransform coordinate_transform;
216     coordinate_transform.LlaToEnu(imu_data.latitude, imu_data.longitude, x, y);
217
218     // east is x and north is y in the HDMap.
219     Point2D imu_neu_origin_point;
220     imu_neu_origin_point.x = x;
221     imu_neu_origin_point.y = y;
222
223     // Get lane's info from HD map.
224     map<int32_t, vector<Point2D>> lanes_points;
225     getLanePointsFromHDMap(imu_neu_origin_point, lanes_points);
226
227     map<int32_t, vector<Point2D>>::iterator it = lanes_points.begin();
228     for (; it != lanes_points.end(); ++it)
229     {
230         Point2D&& point_vehicle = convertNEU2VehicleCoordinate(it->second, imu_neu_coordinate, heading);
231         it->second.x = point_vehicle.x;
232         it->second.y = point_vehicle.y;
233     }
234
235     // Store the truth value and the output value of dto to binary file.
236     storeGroundTruth(groundtruth_file, imu_neu_origin_point, lanes_points);
237
238     storeDTOOutput(dto_result_file, lanes);
239
240     // Store the evaluation to data base.
241     double horizontal_threshold = 0.201f; // unit is m.
242     // key: lane id, values: lane points which sit out of the threshold.
243     map<int32_t, vector<Point2D>> evaluation_result;
244     doEvaluation(lanes, lanes_points, horizontal_threshold,
245                 evaluation_result);
246     mDataBase.insert();
247
248     sfdgd-----
249     -----
250 }
251
252 //evaluation
253 // adas
[Buf List] 2,1 All Develop/evaluation/data_collector/src/lane_evaluator.cpp
```

按下shift+F8, "sfdgd" 附件的空格都被去掉了

```
" Press ? for help
.. (up a dir)
/home/broton/develop/adas/
  adas/
  AdasUI/
  Algorithm2/
  AutoMap/
  Calibration/
  CI/
  Develop/
  Platform/
  Reference/
  Simulation/
  Tools/
  :w
  AdasUI.tar.gz*

187 LanLine& dto_right_line = dto_lanes.right_line;
188 LanLine& dto_left_line = lanes.left_line;
189 size_t hdmap_lane_number = hdmap_lanes_points.size();
190
191 if (hdmap_lane_number == 2)
192 {
193     evaluateLaneLine(dto_right_line, RIGHT_LANE, hdmap_lanes_points[], result);
194 }
195 else if (hdmap_lane_number == 3)
196 {
197 } else if (hdmap_lane_number == 4)
198 {
199 } else
200 {
201     // print_log;
202 }
203
204
205 evaluateLaneLine(left_line, LEFT_LANE, dto_lanes_points, result);
206
207 return true;
208 }
209
210 void LaneEvaluator::evaluate(const ImuData& imu_data, const Lanes& lanes)
211 {
212     double x = 0.0;
213     double y = 0.0;
214
215     CoordinateTransform coordinate_transform;
216     coordinate_transform.LlaToEnu(imu_data.latitude, imu_data.longitude, x, y);
217
218     // east is x and north is y in the HDMap.
219     Point2D imu_neu_origin_point;
220     imu_neu_origin_point.x = x;
221     imu_neu_origin_point.y = y;
222
223     // Get lane's info from HD map.
224     map<int32_t, vector<Point2D>> lanes_points;
225     getLanePointsFromHDMap(imu_neu_origin_point, lanes_points);
226
227     map<int32_t, vector<Point2D>>::iterator it = lanes_points.begin();
228     for (; it != lanes_points.end(); ++it)
229     {
230         Point2D&& point_vehicle = convertNEU2VehicleCoordinate(it->second, imu_neu_coordinate, heading);
231         it->second.x = point_vehicle.x;
232         it->second.y = point_vehicle.y;
233     }
234
235     // Store the truth value and the output value of dto to binary file.
236     storeGroundTruth(groundtruth_file, imu_neu_origin_point, lanes_points);
237
238     storeDTOOutput(dto_result_file, lanes);
239
240     // Store the evaluation to data base.
241     double horizontal_threshold = 0.201f; // unit is m.
242     // key: lane id, values: lane points which sit out of the threshold.
243     map<int32_t, vector<Point2D>> evaluation_result;
244     doEvaluation(lanes, lanes_points, horizontal_threshold,
245                 evaluation_result);
246     mDataBase.insert();
247
248     sfdgd
249
250 }
251
252 //evaluation
253 // adas
[Buf List] 2,1 All Develop/evaluation/data_collector/src/lane_evaluator.cpp [+]
```

253 substitutions on 253 lines

5) tag(intelligence)跳转功能, 将光标移动到要跳转的符号, 按CTRL+], 可能会有多处有该符号, 输入编号即可跳转到相应的文件 (到底是哪个文件里的符号要人工判断一下), 如果要从该文件跳回, 按ctrl+t

```
139 {
140 >----rprC.PushRelPose(msg);
141 >----relativePoseQueueC.PushPacket(msg);
142 >----//printf_s("Got Pose...@ %l>\t", msg.car_ts_ticks);
143 }
144 }
145
146 // ----- necessary variables and functions for VELODYNE sensors ----- //
147 VelodyneQueue veloQueueC(50000, -10000, 100000);
148 FixedQueue<VelodyneScan> veloDelayQueueC(10, VelodyneScan());
149 void VelodyneCallbackC(VelodyneScan &msg, void*)
150 {
151 >----if (veloDelayQueueC.full()) {
152 >---->----veloQueueC.Push(veloDelayQueueC.oldest());
153 >----}
154 >----veloDelayQueueC.push(msg);
155 >----//printf("Got Velodyne...@ %f\n", msg.vts);
156 }
157
158 // ----- Misc. functions ----- //
159 float GetHighClusterMinWidth(float frontDist) {
160 >----float minWidth = 1.0f + .5f*MIN(1.f, MAX(0, (50.f - frontDist) / 50.f));
161 >----return minWidth;
162 }
163
164
165
166
167
168
169
170 void* Cluster::Process(void* vargp)
171 {
172     Cluster* ptr = (Cluster*)vargp;
173     --
174     lcServer = new LidarClusterServer(); //syf ctt1208
175     >-SetErrorMode(7);
176
177 >----// ----- declare sensor interfaces ----- //
178 >----IbeoClient *ibeoClient = new IbeoClient( ptr->cfmq[17] ); // this calls the standard IP and port number (pre-assigned)
179 >----pose_client *poseClient = new pose_client( ptr->cfmq[10] );
180 >----car_timestamp_sync vts_sync; // something that Velodyne receiver requires....
181 >----VelodyneReceiver *velodyneClient = new VelodyneReceiver(&vts_sync, ptr->cfmq[28]);
182
183 >----// ----- sensor calibration and definition ----- //
184 >----// Ibeo position calibration
185 >----Sensor DriverIbeoSensor = IBEQ_CALIBRATION::CURRENT::FRONT_LEFT::SensorCalibration();
186 >----Sensor CenterIbeoSensor = IBEQ_CALIBRATION::CURRENT::FRONT_CENTER::SensorCalibration();
187 >----Sensor PassengerIbeoSensor = IBEQ_CALIBRATION::CURRENT::FRONT_RIGHT::SensorCalibration();
188 >----// Velodyne sensor position
189 >----Sensor VelodyneSensor = VELODYNE_CALIBRATION::CURRENT::SensorCalibration();
190
191 >----// ----- sensor processors ----- //
192 >----// Ibeo processor
193 >----iip[0] = new SingleIbeoProcessor(DriverIbeoSensor, IBEQ_LEFT, driver_ibeo_idx);
194 >----iip[1] = new SingleIbeoProcessor(CenterIbeoSensor, IBEQ_CENTER, center_ibeo_idx);
195 >----iip[2] = new SingleIbeoProcessor(PassengerIbeoSensor, IBEQ_RIGHT, passenger_ibeo_idx);
196 >----// Velodyne processor
197 >----VelodyneProcessor *velodyneProcessor = new VelodyneProcessor(VelodyneSensor);
198
199
```

Develop/Perception/LocalMap/Clustering/Clustering/ClusterFlex.cpp

```
# prt kind tag file
1 F f SingleIbeoProcessor /home/broton/develop/adas/Develop/Perception/LocalMap/Components/GroundGrid/SingleIbeoProcessor.h
class:SingleIbeoProcessor access:public signature:(Sensor& sensor, BYTE sensorID, int sensorIndex)
SingleIbeoProcessor(Sensor& sensor, BYTE sensorID, int sensorIndex)
2 F c SingleIbeoProcessor /home/broton/develop/adas/Develop/Perception/LocalMap/Components/GroundGrid/SingleIbeoProcessor.h
class SingleIbeoProcessor{
Type number and <Enter> (empty cancels): █
```

输入 1, 即可跳转到SingleIbeoProcessor.h这个文件

6) 列出生成 tag(intelligence) 的目录的情况

在vim命令行(esc切入)连续输入lst (list tag)就会在底部显示tag情况, 这里在.projectcfg配置了代码根目录tag, 所以被tag的目录只有root(表示代码根目录), tag的类型是java 和c++

工程配置支持按子目录tag, 在projectcfg里配置即可, 也可以在vim命令行用命令给指定目录生成tag, 该功能属于进阶功能, 可在projectcfg的说明里看到.

```
Develop/Perception/LocalMap/Clustering/Clustering/ClusterFlex.cpp
0 : root:java|cplusplus
```

7) vim 共享 "+" 寄存器, 实现观察程序复制粘贴

- 安装gvim 以安装 "+" 寄存器, sudo apt-get install vim-gnome,
- 复制光标下行到 "+" 寄存器, 在命令行状态, 连续键入 "+y (注意双引号是访问+寄存器的意思), 也可以先V 选中一部分内容再连续按 "+y, 复制完成后到其它窗口按 ctrl +v 就可以粘贴

9) :Man function 查看系统函数

10) :A cpp h 文件切换

11) 附vim常用快捷键

编辑方面:

ESC 从编辑状态切换到命令行状态

i 切换到插入模式

a 切换到光标后面, 进入追加模式

A 切换到行尾追加

I 切换到行首追加

ctrl + v 列选模式

V 行选模式

yy nny 复制1行和n行

dw ndw 删除1个单词和n个单词

cw ncw 修改1个单词和n个单词

dd ndd 删除1行和n行

xp 交换两个字符

r 输入新字符替换当前字符

R 替换光标后所有字符

C nC 修改1行和n行

x nx 删除光标后n个字符

跳转方面:

[[函数块向前跳转

]] 函数块向后跳转

大括号块向前跳转 {{
大括号块向后跳转 }}
% 跳到匹配的括号
ctrl + i 前进到最近编辑位置
ctrl + o 后退到最近编辑位置
连续按下两下 ~ (ESC 下面那个键) 返回上一次编辑的位置
ctrl + 6 切换最近两个文件

还有很多自定义功能请查看 `~/vimrc`

vim非常强大，极细编辑粒度，还可以自己编写插件进行功能扩展，作者是现在已经50多岁了，大部分时间在不以盈利为目的的情况下维护vim这个开源项目，向作者致敬一下，十分推荐使用还有很多，不一一列出

12) 附vim作者的访谈录

贫穷的VIM作者Bram Moolenaar

Moolenaar先生,能介绍一下你自己吗？

我居住在荷兰东边，现在全职搞开源软件。我一个人生活，这样我就不必供养一个家庭。那会很难，因为我做的大部分工作没有报酬。我在家里工作，这样就避免了因为交通阻塞而浪费时间。我和外部世界的大多数沟通是通过email完成的。为了补偿自己，在假期我会到离家很远的国家旅游。我喜欢了解不同的国家及其文化。

你创造了一个VI编辑器的克隆产品。为什么你会选择VI？

最初那只是个偶然的选择。因为在大学的课堂上我们被迫使用这个复杂得可怕的编辑器，却只有一页纸的文档。我开始欣赏VI是很长时间以后的事情了，那时我发现VI让我可以快速编辑文档。一旦我的手指习惯于VI，任何其它的编辑器就显得如此迟钝。因此一旦我拥有了一台amiga计算机，我必须要做一个类似VI的编辑器。那就是我决定开发VIM的原因。

VIM是你从头开发的还是派生于其它的VI克隆？

我开始于Stevie。这是Atari ST电脑上的VI克隆，后来移植到了Amiga。它仍有大量的问题，尚不能完成V能做的任何事情，但是自从放出源代码，我可以自己修正它们。那也是我怎么开始欣赏软件开源的原因。一旦修改代码的数量超过Stevie最初代码，我就把它命名为VIM。

你是用开源协议发布VIM的，向我们描述一下该协议。为什么你不选择GPL？

VIM和其它许多开源项目的区别在于我完成大部分工作。基本上说它是我的项目，即使许多其它人曾在工作上帮助过我。我决定什么特性能加进去什么不能。需要考虑一种符合运行该项目的协议。它允许随意部署未修改的VIM副本。修改过的代码我必须标注到。该协议的一项重要条款是，我可以决定如何处理这些修改。这样如果经过我的允许，一些人就可以把VIM代码用到非自由、非开源的项目中。GPL协议则不允许那样做，因此从那个方面看GPL自由度小一些。理论上你可以在其它协议下使用以GPL协议发布的代码，但是你必须从所有版权人那里获得授权才行。那将非常复杂。实际上我只在很久之前申请了一次特殊授权。VIM的协议从6.1版起升级到兼容GPL。那要求可以部署用GPL协议的代码库编译的VIM副本。这也凸显GPL协议的主要缺点：它事实上是通过限制自由来实行自由的。

籍助于VIM捐助，你援助了乌干达的孩子们。你能看到自己慈善行动的明显成绩吗？

我定期拜访乌干达的孩子中心。每次去那里我都能看到进展。中心的学校在不断发展，我认识的许多孩子持续接受更多的教育。幸运的是，乌干达政局非常稳定，经济也在发展。这让该项目能够专心于改善学校和诊所。这意味着我们寄去的钱真的有助于改善条件并保持长期影响。我们现在看到，孩子完成他们的学业并返回中心工作来帮助教育更年幼孩子。

最初的ex/vi的设计怎么可能仍如此实用？

基本想法一直是这样的，最重要的命令都是通过键盘的一些标准键来传达的，大多数是字母键。这使我们可以很快输入命令。VIM保留了VI的这一设计，但增加了许多在其它编辑器里发现的特性。这样你在两个世界都得到了最优秀的编辑器：常用命令可以快速键入，还有一些高级特性可用。

人们通常对它既爱又恨，为什么？

在你能上手之前你要花费很长时间来学校vi或vim。如果你只试用十分钟就放弃它，那么你将只会记住受它折磨的痛苦时刻。当你不曾花费时间学习超越基本命令的部分的时候，你将不会获益于可能的有效编辑。但是当你的确学会使用vim时，你能非常有效地工作，并会喜欢使用它。之后你会开始觉得其它编辑器很麻烦。比如，大多数其它编辑器不能重复一项变动。当我偶然使用word，结果我在编辑器里输入了".w"和"j"。

是的，比如当记笔记时这非常有帮助。当你在VIM上工作时你在用VIM吗（开玩笑）？

当然了。我用Vim来处理所有问题。大多时间使用最近版本，这样如果我犯了某个错误，我会是第一个注意到的人。

关于这个问题你透露一下：你使用什么操作系统？

我的主战场运行于freebsd4.9。它非常可靠，虽然有一些保守。我的桌面是KDE，虽然在实际工作中我大多时间运行xterms。我也有一台运行windows的机器，它用来进行Vim开发和运行一些驱动程序只兼容windows的特殊硬件。偶然我也切换到我的Amiga2000机器上，Vim开发是在那里开始的。

我们（整个编辑部门）常常使用Vim，我们认为它的开发已经结束。你认同我们的观点吗？

我也奇怪我是否应该停止加入特性并只是修改bug。所以我让赞助人投票决定是否加入某个特性。其中一项就是“停止加入特性，已经足够了”。

它位于投票列表的最底端，这样就很清楚Vim用户的确希望加入新的特性。

所以你不想和Knuth先生对待他的Tex那样的激进。告诉我们有关编辑器整体设计的一些事情,你使用什么语言和技术？

我使用优秀的古老的C代码。为了和足够多的C编译器兼容，我在使用现代C编译器的某些特性方面非常保守。这大多时间只出现在GUI代码里，无论如何它还是需要一个现代C编译器。主代码甚至不使用ANSI函数原型，因为它们不兼容那些更古老的系统。使用纯C的主要优势在于它相对来说简单一些，有很多可以使用的工具。代码体积可以庞大一些，需要在释放分配内存时小心一些。但是它比C++容易得多。

像Java或Python这样的语言可能更容易一些，但是他们运行起来要慢很多。

你认为Vim太艳丽了么？

在Vim里有很多途径来改变颜色。我个人更喜欢只对少数情况配色。它使快速辨认不同的文本块变得非常容易。你可以在C语法高亮里看到配色结果，这样很容易就找到注释。但并不是每块文本都高亮。运算符或函数名并不高亮。一些人写了对所有高亮每块文本的语法脚本，如果你不喜欢那样你可以修订语法脚本规则，为更多的规则选择绝佳文本颜色。

告诉我们一些关于最近稳定版本的新特性。

Vim6.3加入了一些新特性。我专心于修正所有报告的bug。加入了一些命令，主要为了使写脚本更为快捷。比如:keepjumps使我们可以在不影响跳转表（jumplist）的情况下在文件间跳转。一个真正的新特性是支持翻译的帮助文件。那是因为现在有意大利和法国的文档翻译。这样用户需要通过“helplang”选项来选择他们喜欢的语言。以后会陆续增加更多的文档翻译。

你的近期和长期计划是什么？

Vim7将加入许多新特性。我已经开始加入人们发送给我的补丁。现在已经支持KDE了，使用的是Qt库。已经支持在PostScript打印机打印多字节文本。我计划往Vim脚本语言中加入列表和字典，这样使编写高级脚本变得更容易些。Vim脚本语言起初非常简单，但是Vim用户用它编写了越来越多的复杂脚本。在www.vim.org上可以发现超过一千个脚本。加入两个数据类型会使脚本编写更为简单。不知何故Vim用户不喜欢使用python和Perl接口，这样扩展Vim脚本语言是势在必行的了。

我也计划加入一些大特性。我会用投票列表来决定先加入哪个。目前位于列表顶端的是智能补全，也可以称之智能感知（intellisense）。那个实现将是个挑战，尤其假设它应该支持许多语言和所有的平台。

我也需要清理一下代码。大量东西被加入，长时间的修改，导致代码混乱和函数冗长。这不仅是为了使代码看起来好看，它也应该避免犯错误。一些部分已经变得非常复杂，要想修改而不引入新的bug非常困难。但是我很谨慎，代码清理本身也可能引入问题。为了防止这种情况，需要增加更多的测试，这也是Vim7的另一个目标。

你认为Linux发展的方向正确吗？

我不认为对linux而言只存在一条路径。它可以在同一时间无错误地往许多方向发展。我的确看到了一个危机：目前的linux开发大多数是由技术人员驱动的。他们是为linux工作机制做选举的最佳人选。商业和市场人员将尝试把linux朝不同方向推动，目的只有一个：赚更多的钱。我确实希望，这将不会导致错误的选择。

你害怕软件专利权吗？你在源代码里有专利审计吗？

我意识到软件专利权的危机。幸运的是，在欧洲软件专利权仍未被承认。这意味着要想控告我个人在代码里使用有专利的概念是困难的。仍存在一个危机，公司可能停止部署Vim，因为他们需要通过我的授权。这仅是一个很小的风险，因为我没看到哪家公司因为停止部署Vim而获利。很明显，没人会向他们支付一笔协议费。我将只是删除那些侵权代码。

专利审计只能由那些管理大量钞票的专门人员来完成。为Vim不值当做那些。我不确信为了几篇代码的小软件值得那样做，因为为了发布软件那将需要好几年才能申请下来一个专利。因此甚至当审计不出任何问题，次年才出来的专利仍会导致问题。专利系统执行情况实在太恶劣了，尤其在美国。真正获利的人只有那些律师和专利局的雇员。那就是为什么他们是承认软件专利权的主要支持者。

如果软件专利权被承认了并且一些公司为“使用H,J,K,L来移动光标移动”申请了专利，你将采取什么措施？我们说的一些拥有商业化vi克隆产品的公司，他们想消灭对手。你会从vim里移除这项特性吗？

不可能为已经众所周知的方法申请专利。这被称为现有技术（prior art）。很明显像使用hjkl来移动光标这样的方法在专利申请之前就已经存在了。然而，专利局不可能知道这个，无论如何也会授权该专利。那时我将不得不为了自卫而展开一场合法的战争。我不知道我从哪里获得这笔活动资金。专利律师太昂贵了。

感谢你接受这次采访，我们希望你在个人生活和职业生涯都能取得成功。
