浅谈 PYTHON 与 LINUX 的千丝万缕

09378052 李映真

一 PYTHON/LINUX 简介

1. 关于 PYTHON

PYTHON（英语意为蟒蛇）是一种面向对象、直译式计算机程序设计语言，由 Guido van Rossum 于 1989 年底发明，第一个公开发行版发行于 1991 年。PYTHON 语法简捷而清晰，具有丰富和强大的类库。它常被昵称为胶水语言，它能够很轻松的把用其他语言制作的各种模块（尤其是 C/C++）轻松地联结在一起。常见的一种应用情形是，使用 PYTHON 快速生成程序的原型（有时甚至是程序的最终界面），然后对其中有特别要求的部分，用更合适的语言改写，比如 3D 游戏中的图形渲染模块，速度要求非常高，就可以用 C++重写。

2. 关于 LINUX

LINUX 是一种自由和开放源码的类 Unix 操作系统。目前存在着许多不同的 LINUX,但它们都使用了 LINUX 内核。LINUX 可安装在各种计算机硬件设备中，从手机、平板电脑、路由器和视频游戏控制台，到台式计算机、大型机和超级计算机。严格来讲，LINUX 这个词本身只表示 LINUX 内核，但实际上人们已经习惯了用 LINUX 来形容整个基于 LINUX 内核，并且使用 GNU 工程各种工具和数据库的操作系统。

3. LINUX 和 PYTHON

对于开发人员来说，PYTHON 是简历上的一大亮点，而且 PYTHON 开发者们也明显会从熟悉 LINUX 平台的角度受益，因为任何 LINUX 发行版都会将 PYTHON 作为同绑推出的一项标准功能。PYTHON 能够从许多不同的站点处免费下载甚至是从各类 LINUX 发行版的软件包中获取，这意味着一定会有大量潜在的用户在下载并利用其进行编程。我的几台电脑分别使用的是 UBUNTU11.04 和 11.10 发行版，在安装的时候均默认绑定安装了 PYTHON2.7.2，可见 PYTHON 逐渐被主流的 LINUX 玩家认可，并且愿意学习使用 PYTHON 进行开发的人也越来越多，至少在我身边的不少有志于从事 IT 行业的同学是如此。

在 LINUX JOURNAL 于 2011 年 9 月在网上所进行的调查中，24%(约 8600 名用户)将 PYTHON 列为自己最喜欢的编程语言。PYTHON 以极为明显的优势击败了人气榜中位居第二的竞争对手 C 语言。显然，PYTHON 在 LINUX 社区中的人气相当高。而在一些国外大学和科研机构共享的开源代码中我们发现，大部分代码均是 PYTHON 或者 MATLAB 写成。尤其是 PYTHON 程序可以在 LINUX 下完美运行，这对于计算海量数据（比如海量数据挖掘）和大型科学计算等需要进行并行计算和网格计算的项目来说，PYTHON 也成了 C/C++之外的一个不错的选择。

另外，已经有不少 IT 企业开始使用 PYTHON 进行企业级程序开发，其中不乏 GOOGLE、微软、百度等名企。这些企业有些是使用 PYTHON 开发部分模块，有些是使用其来开发程序框架，甚至一小部分企业完全使用 PYTHON 进行开发。据我了解的几家国外的

使用 PYTHON 进行全过程开发的企业，大部分是利用开源资源开发科学计算程序，并且也开放源码供外界下载，这也提供了一个让我了解企业级程序编码的机会。

二 PYTHON 的优势与劣势：与 C/C++对比

1. PYTHON 的优势

PYTHON 在设计上坚持了清晰划一的风格，这使得 PYTHON 成为一门易读、易维护，并且被大量用户所欢迎的、用途广泛的语言。PYTHON 直接编写的程序段有时运行效率甚至高于用 C 编写的程序。从个人开发 PYTHON 程序和 C++程序的经验来看，编写 PYTHON 程序能让我更加集中精力去考虑算法的整体流程框图和逻辑结构，相比之下写 C++程序则需要事先处理许多细节问题。

PYTHON 是一种代表简单主义思想的语言。阅读一个良好的 PYTHON 程序就感觉像是在读英语一样。PYTHON 极其容易上手，因为 PYTHON 有极其简单的语法，它使你能够专注于解决问题而不是去搞明白语言本身。在我最近进行的某个项目的开发过程中，团队中有一名开发者并不熟悉 PYTHON 语言，但是他依旧能在我们标注的注释的指导下轻松地理解其他人所写的 PYTHON 程序，并且现在已经学会基础的 PYTHON 编程。

PYTHON 的免费与开源。PYTHON 是 FLOSS（自由/开放源码软件）之一。使用者可以自由地发布这个软件的拷贝、阅读它的源代码、对它做改动、把它的一部分用于新的自由软件中。FLOSS 是基于一个团体分享知识的概念。因为我主要是在国外的一些开源代码共享网站，如 GITHUB, SOURCEFORCE 等下载及上传开源代码，发现一些科学计算的开源程序，开发者大多都提供 MATLAB 或 PYTHON 的代码，这与 PYTHON 的可移植性和大量共享的开源科学计算包密不可分。这些高效的共享资源，也是吸引我学习 PYTHON 编程的重要原因之一。

用 PYTHON 语言编写程序的时候无需考虑诸如如何管理你的程序使用的内存一类的底层细节。这一点相比 C/C++来说确实更能让开发者集中精力在算法设计上。我虽然没有 C/C++内存编程的经验，但是在学习汇编语言的时候，仅仅是做一个小型的汉诺塔游戏就已经体会到设计内存布局和存取的复杂性。

PYTHON 的作者有意的设计限制性很强的语法，使得不好的编程习惯（例如 if 语句的下一行不向右缩进）都不能通过解释器。其中很重要的一项就是 PYTHON 的缩进规则。这一点尤其在交互环境下进行编程的时候要特别注意，因为不是所有的 IDLE 都可以很好的自动提示缩进，尤其是直接在 LINUX TERMINAL 下进行 PYTHON 交互编程的时候，需要时刻提醒自己是否有及时键入缩进符。由于对缩进的强制限制，使得 PYTHON 源码更加清晰可读，而且对于一些嵌套用法，阅读者也能快速分辨，并且可以以此作为评价算法优劣的一个补充。C/C++是使用花括号{}区分各模块和函数，尽管在编译的时候不会出现问题，但是对于良好的编程习惯的养成，C/C++并没有特别的体现。

其他的一些优势还有：

可扩展性：如果需要一段关键代码运行得更快或者希望某些算法不公开，可以部分程序用 C 或 C++编写，然后在 PYTHON 程序中使用它们。

可嵌入性：可以把 PYTHON 嵌入 C/C++程序，从而向程序用户提供脚本功能。

丰富的库：PYTHON 标准库确实很庞大。它可以帮助处理各种工作，这被称作

PYTHON 的“功能齐全”理念。除了标准库以外，还有许多其他高质量的库，并且都是开源共享的，开发者可以从开发者社区免费获得它们。

可移植性：由于它的开源本质，PYTHON 已经被移植在许多平台上（经过改动使它能够工作在不同平台上）。

2. PYTHON 的劣势

在相同的算法设计下，PYTHON 程序的运行速度总体上看不如 C/C++程序。这是因为 C 或 C++写的程序可以从源文件（即 C 或 C++语言）转换到计算机使用的语言（二进制代码，即 0 和 1），而 PYTHON 语言写的程序不需要编译成二进制代码，直接从源代码运行程序。两者相比，速度高下立分。所以使用 PYTHON 开发辅助模块或者小程序，由于不需要编译，直接进行解释运行，所以编程效率会比较高。但是对一些大型程序以及一些不希望移植到别的平台的程序，C/C++还是主流的选择。

三 在 LINUX/WINDOW 下进行 PYTHON 编程

1. 编程环境

习惯在 WINDOWS 下工作的入门者可能会不习惯 PYTHON SHELL 下的命令行工作方式，PYTHON 也支持在 IDLE 下新建*.py 文件并编写。个人觉得命令行交互也算是 PYTHON 设计者吸收了 LINUX 终端编程的优秀特质，而且这种编程方式对于习惯 LINUX 操作的开发者来说非常亲切，也增加了吸引他们进行 PYTHON 开发的筹码。对于一些习惯用 MATLAB/MATHEMATICA 进行科学计算程序开发的程序员来说，由于 PYTHON 导入科学计算相关模块后对数值计算、符号计算等强大的支持以及类似的交互环境编程，也让他们在使用 PYTHON 时得心应手。

大部分非计算机专业的学生在学习编程语言（如 C/JAVA）等时习惯使用 VISUAL STUDIO 等编程环境，而对 PYTHON 偏 LINUX 风格的编程方式感到头疼。其实很多 PYTHON 的程序设计者也推出了类似 VISUAL STUDIO 的集成环境，如 WINGIDE, PyScripter, SPYDER 等。我在 windows 下使用的集成开发环境是 SPYDER，它提供了编程、调试、交互等功能，并且有强大的在线帮助、函数查询以及实时纠错等功能。这里又体现出 PYTHON 的优势：因为 PYTHON 是解释器解释运行，所以在写代码过程中解释器可以提示语法错误，而 C/C++需要进行编译后才能返回错误信息，相比之下比较麻烦。

关于科学计算的编程环境，我个人比较推荐 PYTHON(X,Y)。这个发行版附带了科学计算方面的很多常用库，最常用的有 NUMPY/SCIPY 等。另外还有大量常用库和工具如 IDE，制图制表工具。PYTHON(x, y)还附带了手工整理出的所有库的离线文档，每个小版本升级都提供单独的补丁。

2. LINUX 下 PYTHON 编程

LINUX 下并没有 WINDOWS 下的集成编程环境，所以对 LINUX 初学用户或者习惯在集成编程环境下写代码的开发者来说会不习惯。但是 LINUX 下也有一些开发环境非常适合进行 PYTHON 编程。

在 LINUX 下写程序，相信很多人都会推荐 VIM 或者 EMACS。作为现今最优秀的几个

文本编辑器，二者都提供了许多强大的功能，不仅仅像 WINDOWS 下简单的文本编辑器和写字板一样的功能，还在非插入模式下提供了许多命令，可以方便的进行高亮、定位、移动、删除等操作，可以说是省去了编程人员双手游移于鼠标和键盘之间的麻烦，而且当熟悉二者的用法后，程序员的双手甚至不用离开主键盘区域。

我在 LINUX 下使用 EMACS 进行编程。EMACS 拥有类似 WINDOWS 下 IDLE 的编程环境，可以在其下编写各种编程语言的代码（如 C/C++/PYTHON/JAVA 等）以及文本文件。它会根据创建新对象的后缀名自动识别编程语言类型，并且在编程过程中根据特定编程语言的特性自动高亮、提示缩进等。

四 结束语

在过去几年中一直都有对 PYTHON 发展前景的质疑。然而，自 1991 年诞生以来，PYTHON 这种稳定且全面的高级编程语言在普及程度方面一直发展良好，尤其在 LINUX 用户群体中更是得到广泛认同。而部分的质疑，也是来自于评论者对 PYTHON 的不熟悉或者对与 C/C++等不同的编程风格等的排斥心理。

PYTHON 的许多忠实拥趸都会赞同并心醉于 PYTHON 官方网站上的一段关于这款组织严密、标准统一的梦幻语言的描述：“PYTHON 将强大的功能与清晰的语法结合起来。它具备大量指向各类系统调用及函数库的接口，包括 Windows 系统的各版本;它同样能与 C 或 C++相联动。它又是一款具备极强可扩展性的语言，为那些需要可编程接口的应用程序带来福音。”我在进行 PYTHON 程序开发的时候深深体会到，它能使我集中于算法设计本身，贯彻自顶向下逐步求精的思想。进行 PYTHON 开发给我带来了极大的乐趣，而方便快捷的解释器功能也为我节省了不少的调试时间，使得程序开发更加简洁高效。

LINUX 自诞生以来在程序设计界中广受欢迎，究其根本也是因为其提倡的开源性和可扩展性。PYTHON 和 LINUX 在设计思想上的契合，使得它们的合作走向必然，而且相信在日后的发展中，PYTHON 可以借助 LINUX 平台以及背后广布全球的开发者社区，在程序设计语言领域占据越来越重要的位置。

五 参考文献

1. PYTHON-百度百科 http://baike.baidu.com/view/21087.htm
2. PYTHON 在 LINUX 平台上的发展前景 http://hb.qq.com/a/20110823/000017.htm
3. LINUX-百度百科 http://baike.baidu.com/view/1634.htm
4. PYTHON(X,Y) http://www.PYTHONxy.com/
5. VIM-百度百科 http://baike.baidu.com/view/113188.htm#2