



2 Python学习的预备内容

AI领域中的Python开发 --- by 丁宁

@(SIGAI课程录制)

- 上节课：建立认知；铺垫课程学习的思维方式
- 本节课：Python学习的预备知识及准备工作

本节课主要内容

- Linux系统简介
- Linux命令行简介
- Linux文件系统
- Linux账户读写权限简介
- Python解释器
- IPython
- NoteBook
- 命令行编程模式与脚本编程模式
- SIGAI在线编程模块简介

Linux系统简介

Linux

- 严格来说，Linux单指操作系统内核
- Linux发行版：Linux内核+应用软件（打包）
- 知名的发行版：Ubuntu、RedHat、CentOS、Debian、Fedora、SuSE、OpenSUSE等

跨硬件平台

- Linux最初是作为支持英特尔x86架构的个人电脑的**自由**操作系统，目前已移植到更多硬件平台
- Linux可运行在服务器和大型计算平台之上，如大型主机和超级计算机等
- 世界500个最快的超级计算机，90%以上运行在Linux发行版或其变种之上
- 世界上最快的10个超级计算机运行的均为基于Linux内核的操作系统
- 在嵌入式上广泛应用：手机，平板电脑，路由器，电视，电子游戏机
- Android操作系统就是创建在Linux内核之上

一切皆是文件

- 系统中所有的东西对于Linux内核来说，均是文件：命令，软件，进程，硬件设备等
- 大道至简的设计思路，从1969年的Unix到1991年的Linux基本保持了一致
- **带着一切皆是文件的思路去理解Linux操作系统**

Linux命令行简介

Shell

说到命令行，真正指的其实是 `Shell`。`Shell` 是一个程序，接受从键盘输入的命令，然后把命令传递给操作系统去执行。几乎所有的 Linux 发行版都提供一个名为 `bash` 的来自 GNU 项目的 `shell` 程序。

终端仿真器程序-terminal

当使用图形用户界面时，我们需要另一个和 shell 交互的叫做终端仿真器的程序。

- 功能：让我们能访问Linux系统上的Shell程序

现在，能区分Shell，Bash，Terminal了吗？其实还有一个Console

- Linux 内核
- Linux 内核 + 应用程序组 = Linux 发行版（也就是操作系统）
- Linux 发行版 + 硬件（主板，CPU，内存，硬盘等）+ 电源 = Linux 服务器
- Linux 服务器相连接的外设（显示器，键盘）为 **Console**
- 外设与Linux内核的交互通过 **shell** 程序完成
- 常用的 **shell** 程序是 **bash**
- 通过网络和 **shell** 程序在另一台设备上仿真出了一个 **console** 叫 **terminal**
- 因此，**console** 也叫 **primary terminal**

使用SIGAI在线编程模块的terminal

1. 进入SIGAI在线编程页面
2. 选择terminal
3. 看到如下字符界面：

```
sigai@8a5f47e78164:~$
```

- 什么意思： `用户名@主机名:当前工作目录$`

`$` 叫做 shell 提示符，无论何时当 shell 准备好了去接受输入时，它就会出现。

如果提示符的最后一个字符是“#”，而不是“\$”，那么这个终端会话就有超级用户权限。这意味着，我们或者是以 root 用户的身份登录，或者是我们选择的终端仿真器提供超级用户权限。

- 一般情况下，SIGAI在线编程模块不会提供 **root** 超级用户权限

试试运行一些简单命令

- `date`
- `cal`
- `df`
- `free`

Linux文件系统及读写权限简介

先学几个常用命令

- `ls`
- `cd`
- `..`

→ `date -d 'now'`

→ 阳历

→ 文档使用情况

→ 内存 state

找到SIGAI在线编程平台的根目录

```
sigai@8a5f47e78164:~$ ls
sharedata workspace
sigai@8a5f47e78164:~$ cd ..
sigai@8a5f47e78164:/home$ ls
sigai
sigai@8a5f47e78164:/home$ cd ..
sigai@8a5f47e78164:/$ ls
bin  dev      docker-entrypoint.sh  home  lib64  mnt  proc  run  srv  tmp
var
boot  docker  etc                  lib   media  opt  root  sbin  sys  usr
```

Linux目录结构

- / : 根目录，所有的目录、文件、设备都在/之下，/就是Linux文件系统的组织者，也是最上级的领导者。
- /bin : binary的缩写，可执行二进制文件目录。系统所需要的那些命令位于此目录。
- /boot : Linux的内核及引导系统程序所需要的文件目录
- /dev : device的缩写，包含了Linux系统中使用的所有外部设备
- /etc : 最重要的目录之一，系统管理中的各种配置文件均在此
- /lib : 存放系统级动态链接共享库，所有用户共享
- /mnt : 挂载存储设备目录，如果有外置存储设备，如大型硬盘，挂载之后会在此目录找到
- /media : 有些发行版用此目录挂载usb接口的移动硬盘等
- /opt : 主要存放可选应用程序
- /usr : 占用空间最大的目录。用户的应用程序多存放于此 ✓
- /usr/local : 主要存放手动安装的软件
- /usr/share : 系统共用的文件存放地，如帮助文档，字体文件等
- /home : 如果建立一个用户sigai，那么 /home 下就会有一个 /home/sigai 目录 ✓

~ 就是 /home/sigai 的简写 (如果以sigai登录的话)

用户的数据

账户读写权限

- `ls -al` : 详细打印当前目录文件信息

```
sigai@8a5f47e78164:/$ ls -al
total 4
drwxr-xr-x  1 root root  74 Jul 10 16:42 .
drwxr-xr-x  1 root root  74 Jul 10 16:42 ..
drwxr-xr-x  1 root root  81 Jul  5 22:46 bin
drwxr-xr-x  2 root root   6 Apr 13  2016 boot
drwxr-xr-x  5 root root 420 Jul 10 16:42 dev
drwxr-xr-x  1 root root  30 Jul  6 06:27 docker
-rwxrwxr-x  1 root root  38 Jul  6 07:11 docker-entrypoint.sh
-rwxr-xr-x  1 root root   0 Jul 10 16:42 .dockerenv
drwxr-xr-x  1 root root  25 Jul 10 16:42 etc
drwxr-xr-x  1 root root  19 Jul  6 06:26 home
```

```

drwxr-xr-x  1 root root  45 Jun 28 09:27 lib
drwxr-xr-x  2 root root  34 May 26 01:45 lib64
drwxr-xr-x  2 root root   6 May 26 01:45 media
drwxr-xr-x  2 root root   6 May 26 01:45 mnt
drwxr-xr-x  2 root root   6 May 26 01:45 opt
dr-xr-xr-x 188 root root   0 Jul 10 16:42 proc
drwx----- 1 root root  41 Jul 10 18:51 root
drwxr-xr-x  1 root root  42 Jul  5 22:46 run
drwxr-xr-x  1 root root  21 Jun  6 05:21 sbin
drwxr-xr-x  2 root root   6 May 26 01:45 srv
dr-xr-xr-x 13 root root   0 Jul 10 16:42 sys
drwxrwxrwt  1 root root   6 Jul 10 19:01 tmp
drwxr-xr-x  1 root root  19 May 26 01:45 usr
drwxr-xr-x  1 root root  19 May 26 01:45 var

```

看第一列字符串

- d* ● 文件类型
- rwx* ● root权限
- r-x* ● 用户组权限
- r-x* ● 用户权限

文件类型

- 普通文件: -
- 目录文件: d
- 链接文件: l *→ link*
- 设备文件: b c
- socket文件: s
- 管道: p

权限类型

- r: 可读
- w: 可写
- x: 可执行

Linux系统的简介到此为止，马上进入Python的部分

Python解释器

Python的程序是怎么执行的

- 计算机执行的是机器代码
- 人类写出来的代码需要被翻译成机器代码
- 翻译的形式一般有三种：编译 vs 解释 vs 先编译后解释
- Python是一门解释型语言：先翻译，再执行
- Python的翻译基于虚拟机：先编译，后解释（PyCodeObject与pyc文件）

增强版解释器-IPython

→ 编译的模块

- 代码补全: **tab**
- 查看信息: **?**
- 与命令行交互: **!**
- 魔法命令: **%magic**
- 类MATLAB的科学计算模式: **-pylab**

更强的工具-Jupyter Notebook

- 可以轻松的将一系列的运行结果保存成文档: 分享和保存
- 基于web可以使用远程的后端计算引擎: 笔记本电脑+服务器计算

REPL-DD (交互式命令行驱动开发)

- 命令行模式与脚本模式
 - 初学Python的最佳编程模式: REPL
-

SIGAI在线编程功能

四个模块

- terminal
- editor
- IPython
- notebook

使用SIGAI在线编程模块与本地环境的区别

1. 本地环境可以有root权限, SIGAI在线编程模块没有, 但有时候没有root权限更好
 2. 本地环境性能可以改变, SIGAI上性能一般无法变化, 但用于学习是足够用的
 3. 本地环境容易受其他软件和程序的干扰, SIGAI上仅用于学习, 环境纯粹
-

以上就是本节课的全部内容, 下节课开始动手写Python!