**Linux上的Shebang符号(#!)**

首先，这个符号（#!）的名称，叫做”Shebang”或者”Sha-bang”再或者“Hashbang”。

**词源与历史**

Shebang的名字来自于sharp和bang，或hash和bang的缩写，指代Shebang中#!两个符号的典型Unix名称。Unix术语中，井号通常称为sharp，hash或mesh；而叹号则常常称为bang。也有看法认为，shebang名字中的sh来自于默认shell Bourne shell的名称，sh，因为常常使用shebang调用之。

**用途**

Shebang这个符号通常在Unix系统的脚本中第一行开头中写到，它指明了执行这个脚本文件的解释程序。#!是特殊的表示符，其后面跟的是解释此脚本的shell的路径。

1. 如果脚本文件中没有#!这一行，那么它执行时会默认用当前Shell去解释这个脚本(即：*[Math Processing Error]SHELL环境变量）。MacOS下*SHELL = /bin/sh Linux下 $SHELL ＝ /bin/bash

2. 如果#!之后的解释程序是一个可执行文件，那么执行这个脚本时，它就会把文件名及其参数一起作为参数传给那个解释程序去执行。

3. 如果#!指定的解释程序没有可执行权限，则会报错“bad interpreter: Permission denied”。如果#!指定的解释程序不是一个可执行文件，那么指定的解释程序会被忽略，转而交给当前的SHELL去执行这个脚本。

4. 如果#!指定的解释程序不存在，那么会报错“bad interpreter: No such file or directory”。注意：#!之后的解释程序，需要写其绝对路径（如：#!/bin/bash），它是不会自动到$PATH中寻找解释器的。

5. 当然，如果你使用”bash test.sh”这样的命令来执行脚本，那么#!这一行将会被忽略掉，解释器当然是用命令行中显式指定的bash。

6. 如果脚本中还包含有其他的#!行,那么bash将会把它看成是一个一般的注释行。

例如我们写一个python程序，可以在文件头写上python解释器的位置可以用which python查看具体位置，解释which python结果为:/usr/bin/python,那么我们可以在文件头这么写：#! /usr/bin/python

这样在给文件chmod u+x 之后就可以“./”执行（如果使用“python＋文件名”，实际上是不需要使用#!指定的）(“.”执行与”./”的区别：1.如果使用". " 执行，可以理解为程序运行在一个全新的shell中，不继承当前shell的环境变量的值，同时若在程序中改变了当前shell中的环境变量（不使用export），则当前shell的环境变量值不变。2.如果使用”."执行，则程序继承当前shell中的环境变量，同时，若在程序中改变了当前shell中的环境变量（不使用export），则当前shell中该环境变量的值也会改变另外一个区别点在于，“./"只能用于拥有执行权限的文件， 而”."则可以暂时提升)

另外#! /usr/bin/python这么写，如果我换了一个机器，它的python装在其他地方怎么办？可以这么写#!/usr/bin/env python（其就是linux常用的env指令。/usr/bin/env node 表示通过env运行node。env将会从环境变量中寻找node工具。这样能解决多系统中因为工具目录不一样导致脚本执行失败的问题。）