



公告

[XML](#)

本博客文章如非注明均属原创，允许转载，请务必添加原文链接

网名:程默

地址:广东·深圳

行业:IT(linux,服务器，安全，架构)

Q Q:8292669

昵称：[程默](#)

园龄：[7 年 11 个月](#)

粉丝：[337](#)

关注：[0](#)

[+加关注](#)

搜索

常用链接

- [我的随笔](#)
- [我的评论](#)
- [我的参与](#)
- [最新评论](#)
- [我的标签](#)

我的标签

- [linux](#)(37)
- [shell](#)(27)
- [php](#)(15)
- [awk](#)(14)
- [web 安全](#)(5)
- [字符集](#)(3)
- [nginx](#)(2)
- [wordpress](#)(2)
- [字符编码](#)(1)
- [base64](#)(1)
- [更多](#)

随笔分类

- [linux](#)(36)
- [nginx](#)(3)
- [php](#)(16)
- [python](#)(1)
- [web 前端](#)(1)
- [编码解码](#)(1)
- [心得分享](#)(6)
- [字符集](#)(3)

随笔档案

- [2014 年 5 月](#) (1)
- [2013 年 6 月](#) (4)
- [2013 年 5 月](#) (4)
- [2013 年 1 月](#) (1)
- [2011 年 2 月](#) (1)
- [2010 年 12 月](#) (1)
- [2010 年 10 月](#) (42)
- [2010 年 9 月](#) (3)
- [2010 年 7 月](#) (2)
- [2010 年 6 月](#) (4)

友情连接

- [查错网](#)
- [生活查询网](#)

积分与排名

- 积分 - 91496

- 排名 - 3749

阅读排行榜

- 1. linux shell 字符串操作 (长度, 查找, 替换) 详解(298861)
- 2. linux shell 数据重定向 (输入重定向与输出重定向) 详细分析(173282)
- 3. linux shell 流程控制 (条件 if,循环【for,while】,选择【case】语句实例(161696)
- 4. linux shell 自定义函数(定义、返回值、变量作用域) 介绍(138806)
- 5. linux shell 数组建立及使用技巧(137483)

linux shell 实现随机数多种方法 (date,random,uuid)

在日常生活中,随机数实际上经常遇到,想丢骰子,抓阄,还有抽签。呵呵,非常简单就可以实现。那么在做程序设计,真的要通过自己程序设计出随机数那还真的不简单了。现在很多都是操作系统内核会提供相应的 api,这些原始参数是获取一些计算机运行原始信息,如内存,电压,物理信号等等,它的值在一个时间段可以保证是唯一的了。好了,废话我就不说了。呵呵。

shell 脚本程序我们有那些获得随机数方法呢?

一、通过时间获得随机数 (date)

这个也是我们经常用到的,可以说时间是唯一的,也不会重复的,从这个里面获得同一时间的唯一值。适应所有程序里面了。

例子:

?

```
[chengmo@centos5 shell]$ date +%s
1287764773
1 #获得时间戳,当前到:1970-01-01 00:00:00 相隔的秒数
2
3 #如果用它做随机数,相同一秒的数据是一样的。在做循环处理,多线程里面基本不能满足要求
4 了。
5
6 [chengmo@centos5 shell]$ date +%N
7 738710457
8
9 #获得当前时间的纳秒数据,精确到亿分之一秒。
10 #这个相当精确了,就算在多 cpu,大量循环里面,同一秒里面,也很难出现相同结果,不过
11 不同时间里面还会有大量重复碰撞
12
13 [chengmo@centos5 shell]$ date +%s%N
1287764807051101270
```

```
#这个可以说比较完美了，加入了时间戳，又加上了纳秒
```

通过上面说明，用它来做随机数的基数了，接下来我们看怎么样获得一段数据内怎么样获得随机数。

?

```
#!/bin/sh

1 #写个随机函数，调用方法 random min max
2 #在min 与 max 直接获得随机整数
3 #copyright chengmo QQ:8292669
4
5
6
7 #获得随机数返回值，shell 函数里算出随机数后，更新该值
8 function random()
9 {
10     min=$1;
11     max=$2-$1;
12     num=$((date +%s+%N));
13     ((retnum=num%max+min));
14     #进行求余数运算即可
15     echo $retnum;
16     #这里通过 echo 打印出来值，然后获得函数的，stdout 就可以获得值
17     #还有一种返回，定义全价变量，然后函数改下内容，外面读取
18 }
19
20 #得到 1-10 的 seq 数据项
21 for i in {1..10};
22 do
23     out=$((random 2 10000));
24     echo $i,"2-10000",$out;
25 done;
```

看看运行结果：

```
[chengmo@centos5 shell]$ sh testrandom.sh
1,2-10000,5600
2,2-10000,5295
3,2-10000,3432
4,2-10000,3148
5,2-10000,9041
6,2-10000,4290
7,2-10000,2380
8,2-10000,9009
9,2-10000,5474
10,2-10000,3664
```

一个循环里面，得到值各不相同。

这个是我们常用方法，适应各种语言，是一个通用算法，就算服务器不提供，某时刻相同唯一数据标记，我们也可以通过这种方法，做自己的伪随机数。下面还有更简单方法呢，不要我们自己做了。

2、通过内部系统变量(\$RANDOM)

其实，linux 已经提供有个系统环境变量了，直接就是随机数，哈哈，觉得刚学习方法，是不是白费了！！

?

```
1 [chengmo@centos5 shell]$ echo $RANDOM
2 10918
3 [chengmo@centos5 shell]$ echo $RANDOM
4 10001
5
6 #连续 2 次访问，结果不一样，这个数据是一个小于或等于 5 位的整数
```

可能有疑问了，如果超过 5 位的随机数怎么得到呢？

呵呵，加个固定 10 位整数，然后进行求余，跟例 1 一样了。接下来的例子又是我们自立更生做了。

3、通过系统内部唯一数据生成随机数 (/dev/random,urandom)

我们知道 dev 目录下面，是 linux 一些默认设备，它给我们感觉就是放的是键盘，硬盘，光驱等设备的对应文件了。其实 linux 有些设备很特殊，有特殊用途。前面我们说到的：

[/dev/\[udp|tcp\]/host/port](#) 比较特殊吧。呵呵，有扯远了。

/dev/random 设备，存储着系统当前运行的环境的实时数据。它可以看作是系统某个时候，唯一值数据，因此可以用作随机数元数据。我们可以通过文件读取方式，读得里面数据。

/dev/urandom 这个设备数据与 random 里面一样。只是，它是非阻塞的随机数发生器，读取操作不会产生阻塞。

实例：

?

```
1 [chengmo@centos5 shell]$ head -1 /dev/urandom
2 äÅ†ù...•KTpçanVÖä¹Û&;ö¾"ô2iùU" žF|_ ý"†mEöûUráİ=JˉTÿA•îAÚRtÓ
3
4
5 #读一行，怎么是乱码呢？其实它是通过二进制数据保存实时数据的，那么我们怎么样把它变成
6 整型数据呢？
7
8
9
```

```
10 [chengmo@centos5 ~/shell]$ head -200 /dev/urandom | cksum
11 1615228479 50333
12 #由于 urandom 的数据是非常多，不能直接通过 cat 读取，这里取前 200 行，其实整个数据
13 都是变化的，取多少也一样是唯一的。
14 #cksum 将读取文件内容，生成唯一的表示整型数据，只有文件内容不变，生成结果就不会变化，与 php crc 函数

[chengmo@centos5 shell]$ head -200 /dev/urandom | cksum | cut -f1 -d"
"
484750180
#cut 以" "分割，然后得到分割的第一个字段数据
```

得到整型数据，然后，类似一的方法就可以获得随机数了。题外话：在程序里面，我们经常 md5 得到唯一值，然后是字符串的，如果想表示成整型方式，可以通过 crc 函数。crc 是循环冗余校验，相同数据通过运算，都会得到一串整型数据。现在这种验证应用很广。详细要了解，可以参考：[crc](#)。

下面还有个方法，直接从设备读取生成好的 uuid 码。

4、读取 linux 的 uuid 码

在提到这个之前，有个概念，就是什么是 uuid 呢？

UUID 码全称是**通用唯一识别码** (Universally Unique Identifier, UUID), 它是一个软件建构的标准，亦为**自由软件基金会** (Open Software Foundation, OSF) 的组织在**分布式计算环境** (Distributed Computing Environment, DCE) 领域的一部份。

UUID 的目的，是让分布式系统中的所有元素，都能有唯一的辨识信息，而不需要通过中央控制端来做辨识信息的指定。如此一来，每个人都可以创建不与其它人冲突的 UUID。在这样的情况下，就不需考虑数据库创建时的名称重复问题。它会让网络任何一台计算机所生成的 uuid 码，都是互联网整个服务器网络中唯一的。它的原信息会加入硬件，时间，机器当前运行信息等等。

UUID 格式是：包含 32 个 16 进位数字，以“-”连接号分为五段，形式为 8-4-4-4-12 的 32 个字符。范例；550e8400-e29b-41d4-a716-446655440000，所以：UUID 理论上的总数为 $2^{16 \times 8} = 2^{128}$ ，约等于 3.4×10^{38} 。也就是说若每奈秒产生 1 兆个 UUID，要花 100 亿年才会将所有 UUID 用完。

其实，大家做数据库设计时候，肯定听说过，guid(**全局唯一标识符**)码，它其实是与 uuid 类似，由微软支持。这里编码，基本有操作系统内核产生。大家记得把，在 windows 里面，无论数据库，还是其它软件，很容易得到这个 uuid 编码。

linux 的 uuid 码

linux 的 uuid 码也是有内核提供的，在 /proc/sys/kernel/random/uuid 这个文件内。其实，random 目录，里面还有很多其它文件，都与生成 uuid 有关系的。

?

```
[chengmo@centos5 ~/shell]$ cat /proc/sys/kernel/random/uuid
1 dff68213-b700-4947-87b1-d9e640334196
2 [chengmo@centos5 ~/shell]$ cat /proc/sys/kernel/random/uuid
3 7b57209a-d285-4fd0-88b4-9d3162d2e1bc
4 #连续 2 次读取，得到的 uuid 是不同的
5
6 [chengmo@centos5 ~/shell]$ cat /proc/sys/kernel/random/uuid | cksum |
7 cut -f1 -d" "
8 2141807556
9 #同上方法得到随机整数
```

这是 linux 下面，几种常见活动随机数整数方法，除了第一个是不同外，其实后 3 个，产生随机码的伪数据来源，都与 /dev/random 设备有关系。只是它们各自呈现不同而已。如果你还有更多其它方法，请给我消息，与大家分享了。

« 上一篇：[linux shell 脚本实现 tcp/udp 协议通讯（重定向应用）](#)

» 下一篇：[linux shell 命令快捷获得系统帮助\(一\)\[man-pages 定义规范\]](#)

posted on 2010-10-23 10:15 [程默](#) 阅读(81542) 评论(14) [编辑](#) [收藏](#)

评论

#1 楼 2010-10-23 10:56 通用 C#系统架构

我来按反对，这里全是做.NET 的，哈哈。

[支持\(1\)](#)[反对\(0\)](#)

#2 楼[楼主] 2010-10-23 11:11 程默

[引用](#)吉日嘎拉 不仅权通用权限：我来按反对，这里全是做.NET 的，哈哈。

呵呵，我 blog 会主要是 linux 方向的，不知道是不是 cnblogs 不建议开 linux 方面 blog 呢？

[支持\(0\)](#)[反对\(0\)](#)

#3 楼 2010-10-23 11:16 iTech

love linux! ding!

[支持\(0\)](#)[反对\(0\)](#)

#4 楼[楼主] 2010-10-23 11:42 程默

[引用](#) iTech : love linux! ding!

握手...

[支持\(0\)](#)[反对\(0\)](#)

#5 楼 2010-10-23 13:19 通用 C#系统架构

博客园是适合 C#.NET 的方向，可能 Linux 的全国也有地方有知名的博客圈吧？

[引用](#)程默：

[引用](#)吉日嘎拉 不仅权通用权限：我来按反对，这里全是做.NET 的，哈哈。

呵呵，我 blog 会主要是 linux 方向的，不知道是不是 cnblogs 不建议开 linux 方面 blog 呢？

[支持\(0\)](#)[反对\(0\)](#)

#6 楼 2010-10-23 13:26 通用 C#系统架构

从排版上看，能看出博主是高手，但是在首页看到 Linux 的文章，总是感觉乖乖的，可能跟我的水文有一比吧。

[支持\(0\)](#)[反对\(0\)](#)

#7 楼 2010-10-23 13:28 吴秦

推荐+1

[支持\(0\)](#)[反对\(0\)](#)

#8 楼[楼主] 2010-10-23 14:27 程默

[引用](#)吉日嘎拉 不仅权限通用：从排版上看，能看出博主是高手，但是在首页看到 Linux 的文章，总是感觉乖乖的，可能跟我的水文有一比吧。

过奖了，只是在河边走时间比较长，算不上捕鱼的。

[支持\(0\)](#)[反对\(0\)](#)

#9 楼[楼主] 2010-10-23 14:30 程默

握手...[引用](#)吴秦：推荐+1

[支持\(0\)](#)[反对\(0\)](#)

#10 楼 2010-10-23 14:45 killkill

写的不错，值得推荐。

[支持\(0\)](#)[反对\(0\)](#)

#11 楼[楼主] 2010-10-23 15:20 程默

[引用](#) killkill：写的不错，值得推荐。

呵呵，欢迎。

[支持\(0\)](#)[反对\(0\)](#)

#12 楼 2011-06-01 14:46 lindiyer

用加密算法生成随机数，也可以

#13 楼 2011-11-03 21:51 donysk

逛到这里，博主总结得很好，而且排版也不错，学习了！

#14 楼 2016-02-19 16:52 少林功夫好

已学习。

[支持\(0\)](#)[反对\(0\)](#)

[刷新评论](#)[刷新页面](#)[返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

【推荐】超 50 万 VC++源码: 大型组态工控、电力仿真 CAD 与 GIS 源码库！

【推荐】华为云 7 大明星产品 0 元免费使用

【推荐】腾讯云如何降低移动开发成本

【大赛】2018 首届“顶天立地”AI 开发者大赛

最新 IT 新闻:

- 美团打车治理“套牌车”：新增线下验车和临检等机制
- 对话陈一舟：人人网最后还是败给了大势
- 精灵宝可梦第二款手游 Pokémon Quest 今日上线，率先登陆 Switch
- 贝佐斯：和互联网一样，太空产业需要创业大爆炸
- 卫哲：做事要有超配意识

» 更多新闻...

最新知识库文章:

- 你可以把编程当做一项托付终身的职业
- 评审的艺术——谈谈现实中的代码评审
- 如何高效学习
- 如何成为优秀的程序员？
- 菜鸟工程师的超神之路 -- 从校园到职场

» 更多知识库文章...

Powered by:

博客园

Copyright © 程默