



时间
关注
发信

Q

- 

东航梧州飞机坠毁132人，历史上还有哪些空难
- 

小公司团队建设思考，协调前端妹纸和IT男氛围
- 自用 Pycharm 主题配色分享（主题才是...
SimpleDateFormat的线程安全问题
Guava包学习---Lists
C#全角和半角转换
[Bhatia.Matrix Analysis.Solutions to Exer...
python 简单验证码 random模块
oracle Hash Join及三种连接方式
SpringMVC：拦截器
Python 的 Pandas 对矩阵的行进行求和
使用 JQuery 实现将 table 按照列排序
论文阅读笔记二十三：Learning to Segm...
【HTML】HTML基础知识 表单
NavigationBar隐藏后，返回时表现不完美
Python:笔记(7)——yield关键字
python基础 Day4

Linux时间子系统之（一）：时间的基本概念

扫码查看

专题文档汇总目录

Notes：Linux时间基准点；Linux时间和broken-down time(struct tm)；不同精度的时间表示time_t、timeval、timespec。

原文地址：[Linux时间子系统之（一）：时间的基本概念](#)

本文使用Q & A的方式来和大家以前探讨一下时间的基本概念

一、什么是时间？

这个问题实在是太复杂了，我都不知道这是一个物理学、宇宙学、还是热力学异或是哲学问题，我只是想从几个侧面来了解一下时间这个概念。本节内容都是我坐在公交车上瞎想的，对物理学有兴趣的人可以指出我的错误（一个搞linux kernel的人不会有太深刻的物理学知识的），对Linux时间子系统有兴趣的人还是忽略这个小节吧。

1、时间和空间以及相对性

有没有绝对时间的概念呢？时间是否是独立于一切存在的呢？相信有绝对时间的存在比较符合人类的思维，也就是说，有一根绝对时间轴存在，任何事件都可以投射在这个时间轴的某个点上，不论观察者处于什么状态，大家共享一个绝对的时间轴。在这样的时间框架下，同时的概念是绝对的，只要发生在时间轴的同一点上，那么两个事件就是同时发生。如果两个事件不同时发生，那么他们之间的间隔也是绝对的，所有感知到的时间间隔都是相同的。与之相对的是绝对空间的概念。任何事件发生的地点都可以对应到绝对空间中的一个位置点，两个位置点的长度是绝对的，任何观察者测量的结果都是一样的，无论观察者处于什么样的状态。

当然，了解狭义相对论的同学应该是可以建立起来自己的时空观，其实没有绝对的时间和空间（时间和空间是相对的而且是有关联的，组成四维时空），同时是相对的（相对于自己的参考系），事件A和事件B先后发生，位于不同的惯性参考系的观察者可以有不同的观测结果，可能一个认为A和B相关1秒，而另外一个惯性参考系的人认为A和B之间相隔1.2秒。当然，我们研究linux kernel中的时间子系统，这里的参考系不可能是以光速级别在运动，因此传统的牛顿的时空观也是OK的，它是爱因斯坦时空观的一个特例。

2、时间的方向

时间有没有起点和终点？时间的箭头指向何方？什么是现在？什么是未来？时间是不是可逆的？有没有可能穿越？

根据热力学第二定律，对于一个孤立系统，其内部自发进行的与热相关的过程必然向熵增的方向进行。一个孤立系统如果不受外界任何影响，且系统最终处于平衡态，那么在平衡态时，系统的熵取最大值。从这个角度看，时间的方向就是熵增的方向，并且是不可逆的。

此外，当时间沿着它的方向流动的时候，是连续的，还是有一个基本的单位，一份一份的流动？我想如果能量的交换（热交换）是一份一份的，那么时间也是离散的。

3、感知时间

时间是和变化相关的，如果一切处于静止态，没有位移、没有能量的变化，那么时间还有存在的意义吗？物体从一个位置A移动到了另外一个位置B，观察者通过感知位置的变化来感知时间，判断事件的先后顺序。当一切处于静止状态，没有任何的运动，各种恒星和行星都处于静止态，电子不再围绕原子核进行高速运动，宇宙中的所有的系统没有能量的交换，那么时间也就停止了，或者说不存在了。本质上，宇宙的状态和时间轴应该是一一对应的，如果宇宙的状态只有一种，那么时间轴就停止在一个点上，不再流动了。

二、度量时间

1、如何定义秒？

在定义秒这个术语的时候是有两个标准的：

(1) 以铯133的振荡频率来定义秒。

(2) 依据地球自转和公转来定义秒。

毫无疑问，第二种方法是符合人类习惯的，大家已经习惯了一天24个小时，每个小时60分钟，一分钟包括60秒数。我们可以用平均太阳日的1/86400来定义，也可以用绕太阳轨道公转一年的时间来定义秒，但是这种基于天体运动而定义的秒其实是不精准的，因为地球的自转和公转的时

间不是一个恒定的值。由此，以铯133的振荡频率这样的物理属性来定义秒实际上可以赋予秒一个恒定的时间长度。

对于linux kernel而言，它采用了第一种定义的秒，并且参考点（linux epoch）也是UTC时间，UTC这个时间标准就是采用第一种方式来定义时间的。历史上，POSIX标准曾经将时间参考点设定为1970年1月1日0点0分0秒（GMT），GMT就是采用第二种方法定义时间的，后面修正为UTC时间。

定义了秒之后，我们可以进一步细分为毫秒、微秒、纳秒等，也可以组织成分钟、小时、日、月、年等概念。

Notes：UTC Universal Time Coordinated，GMT Greenwich Mean Time。

2、如何处理闰秒（leap seconds）？

正是因为UTC时间和我们日常使用的日历时间有差异，因此才有闰秒（leap second）这个概念。虽然你UTC时间使用严格的定义，使用原子钟作为参考，秒的概念被精准的定义，但是，它还是要向现实低头，因为广大人民群众喜闻乐见的是和地球自转和公转相关的日历时间。为了让UTC时间和人民群众的日历时间保持一致，在适当的时间点（误差累积到1秒的时候）对UTC时间进行增加或者减少一秒的操作，这就是闰秒的概念了。

由于目前地球的自转一直在变慢，因此我们的日历时间（GMT）是在变长的，因此目前的闰秒调整都是+1秒的操作。未来是否地球自转会变快，谁知道呢？

对于linux kernel而言，虽然epoch是UTC时间，不过内核不考虑闰秒问题。对于一个天文学意义上的较短的时间段（例如15个月）UTC和GMT是基本相同的，大部分的应用其实也基本是可以正常工作的。对这个问题的解决来自具体的应用场景和系统实现，内核只是支持了NTP协议，可以和外部的时间服务器进行同步，而外部的那些时间同步服务器都是支持UTC的，让它们去考虑闰秒吧，linux kernel只需要保持和它们同步就OK了。

3、如何处理闰年

科学必须用来指导人类的生产和生活，否则就没有意义了。在精确定义了时间的度量后，我们可以用科学的方法来组织时间。例如：地球绕太阳公转的周期是365天6小时9分钟10秒（或者说365.2564天）。不是整数天则需要特别处理，例如：考虑平年365天，闰年366天，每四年有一个闰年。如果这样的话， $0.2564 \times 4 = 1.0256$ ，也就是说，这样的处理就多算了0.0256天，怎么办？累积长了也是一个不小的误差。根据计算，每100次闰年计算（周期400年），就多了2.56天，因此，定义在能被100整除的年份中（如果计算周期是400年，那么有4个这样的年份），选取3个不进行闰年，只有能被400整除的年份才闰年。这样计算的结果就是每400年就少算了0.44天，这样的计算可以无穷无尽计算下去。

4、如何定义基准？

时间有绝对和相对的概念。例如1小时候我们就去吃饭。这里就是一个相对时间的概念，1小时是相对当前时间点而言。1949年10月1日，中华人民共和国成立，这里的时间就是绝对时间。当然，实际上，这里的绝对时间也是相对时间，只不过所有地球上的人都使用同一个基准点的时候，这个时间表示就是一个普遍适用的绝对时间了。

对于一个系统而言，需要定义一个epoch，所有的时间表示是基于这个基准点的，对于linux而言，采用了和unix epoch一样的时间点：1970年1月1日0点0分0秒（UTC）。NTP协议使用的基准点是：1900年1月1日0点0分0秒（UTC）。GPS系统使用的基准点是：1980年1月6日0点0分0秒（UTC）。每个系统都可以根据自己的逻辑定义自己epoch，例如unix epoch的基准点是因为unix操作系统是在1970年左右成型的。

Notes：不同系统的epoch：Linux/Unix-1970年1月1日0点0分0秒；NTP-1900年1月1日0点0分0秒；GPS-1980年1月1日0点0分0秒。

三、Linux kernel中和时间相关的基本概念

1、系统时钟（system clock）

我们在之前的文章中说过，时间就像是一条没有起点、没有终点的线段，除了要给出度量单位（例如：秒），还有给出参考点。对于linux kernel而言，这个参考点就是Linux Epoch。所谓Linux Epoch其实就是1970年1月1日0点0分0秒（UTC）的时间点。虽然人类喜欢年、月、日、时、分、秒这些概念，不过对于计算机，更喜欢使用当前的时间点到linux epoch的秒数。一个符合POSIX标准的系统必须提供系统时钟，以不小于秒的精度来记录到linux epoch的时间值。

内核支持的system clock定义如下：

CLOCK_REALTIME是描述真实世界的时钟（这里的英文realtime有两个意思，一个表示真实的时间的意思，另外一个实时性的意思，其实不止中文有歧义，英文也是一样的，需要根据上下文判断，我们这个场景当然不是讲实时性），就类似挂在墙上的钟表一样，告知人类当前的时间。当然，家里的钟表你当然可以按照你的意愿向前或者向后调整。CLOCK_MONOTONIC是一个禁止人为设定的真实世界的时钟，你没有办法设定它，但是可以通过NTP协议进行调整。更多的system clock ID会在后续的文章中给出详细定义

2、 broken-down POSIX time

在time line上以linux epoch为参考点，用到参考点的秒数来表示timeline上的某个时间点的方法对于计算机而言当然是方便的，不过对于使用计算机的人类而言，我们更习惯broken-down time，也就是将那个冰冷的到参考点的秒数值分解成年月日时分秒。在内核中用下面的数据结构表示（注释非常详细，这里不再赘述）：

3、不同精度的时间表示

Notes：不同精度时间：time_t-秒；timeval-微妙；timespec-纳秒。

传统的unix使用了基于秒的时间定义，相关的数据结构是time_t：

time_t是POSIX标准定义的一个以秒计的时间值。例如大家都比较熟悉的time函数，可以获取一个从linux Epoch到当前时间点的秒值，该函数的返回值类型就是time_t的。如果time_t被实现成一个signed 32-bit integer，那么实际上在2038年的时候就会有溢出问题。

随着应用的发展，秒精度已经无法满足要求，因此出现了微秒精度的时间表示：

struct timeval 的概念和time_t是一样的，只不过多了一个微秒的成员，将时间的精度推进到微秒级别。在计算当前时刻到epoch时间点的微秒数值的时候可以使用公式：tv_sec x 10^6 + tv_usec。

由于实时应用程序的需求，POSIX标准最终将时间精度推进到纳秒，纳秒精度的时间表示如下：

根据POSIX标准，timeline上一个时间点的值用struct timespec来表示，它应该最少包括：

成员数据类型
成员的名字
描述

time_t
tv_sec
该时间点上的秒值，仅在大于或者等于0的时候有效。

long
tv_nsec
该时间点上的ns值，仅在大于或者等于0，并且小于10^9纳秒的时候有效。

struct timespec和struct timeval概念类似，这里不再赘述。

原创文章，转发请注明出处。蜗窝科技
http://www.wowotech.net/timer_subsystem/time_concept.html

请赞赏

朋友，创作不易；为犒赏小编的辛勤劳动，请她喝杯咖啡吧！

给她赞赏，您将财运亨通



支付宝



微信

¥5 ¥10 **¥15** ¥20 ¥30

感谢朋友，赞赏列表

我已完成赞赏，领取礼品

05-07 15:44

SQL 标准中的四种隔离级别

快递鸟电子面单打印功能基于java

Xcode工程添加第三方文件的详细分析 Create folder references for any added folders

ue4框架C++语法汇总文章

什么是Numpy的ndarray

[WinSCP 中普通用户以 root 身份登录 Linux](#)

[cesium随笔 — 隐藏三维场景下方版权信息](#)

[【resultType】Mybatis种insert或update的resultType问题](#)

[树形DP求树的重心 --SGU 134](#)

[20155328 实验四 Android程序设计 实验报告](#)

[RPC介绍](#)

[asp.net 把图片压缩成zip之后再进行下载](#)

[【重学计算机】操作系统D1章：计算机操作系统概述](#)

[dubbo properties](#)

[c#中的as, is和强转](#)

[查看更多](#)