



Linux TTY framework(1)_基本概念

作者：wowo 发布于：2016-9-18 22:55 分类：TTY子系统

1. 前言

由于串口的缘故，TTY是Linux系统中最普遍的一类设备，稍微了解Linux系统的同学，对它都不陌生。尽管如此，相信很少有人能回到这样的问题：TTY到底是什么东西？我们常常挂在嘴边的终端（terminal）、控制台（console）等概念，到底是什么意思？

本文是Linux TTY framework分析文章的第一篇，将带着上述疑问，介绍TTY有关的基本概念，为后续的TTY软件框架的分析，以及Linux serial subsystem的分析，打好基础。

2. 终端（terminal）

2.1 基本概念

在计算机或者通信系统中，终端是一个电子（或电气）设备，用于向系统输入数据（input），或者将系统接收到的数据显示出来（output），即我们常说的“人机交互设备”。

关于终端最典型的例子，就是电传打字机（Teletype）^{[1][2]}——一种基于电报技术的远距离信息传送器械。电传打字机通常由键盘、收发报器和印字机构等组成。发报时，按下某一字符键，就能将该字符的电码信号自动发送到信道（input）；收报时，能自动接收来自信道的电码信号，并打印出相应的字符（output）。

2.2 Unix终端

在计算机的世界里，键盘和显示器，是最常用的终端设备，一个用于向计算机输入信息，一个用于显示计算机的输出信息。

在大型机(mainframe)和小型机(minicomputer)的时代里，终端设备和计算机主机都同属一个整体。但到PC时代，情况发生了变化。Unix创始人肯·汤普逊和丹尼斯·里奇想让Unix成为一个多用户系统。多用户系统意味着要给每个用户配置一个终端，每个用户都要有一个显示器、一个键盘。但当时所有的计算机设备(包括显示器)价格都非常昂贵，而且键盘和主机是集成在一起的，根本没有独立的键盘。

最后他们找到了一样东西，那就是ASR33电传打字机。虽然电传打字机的用途是在电报线路上收发电报，但是它也可以作为人与计算机的接口，而且价格低廉。ASR33打字机的键盘用来输入信息，打印纸用来输出信息。所以他们把ASR33电传打字机作为终端，很多个ASR33连接到同一个主机，每个用户都可以在终端输入用户名和密码登录主机。这样他们创造了计算机历史上的第一个真正的多用户系统Unix，而ASR33成为第一个Unix终端。

2.3 TTY设备

由上面的介绍可知，第一个Unix终端是一个名字为ASR33的电传打字机，而电传打字机的英文单词为Teletype（或Teletypewriter），缩写为TTY。因此，该终端设备也被称为TTY设备。这就是TTY这个名称的来源，当然，在现在的Unix/Linux系统中，TTY设备已经演变为不同的意义了，后面我们会介绍演变的过程。

注1：读到这里，希望读者再仔细思索一下“设备”的概念。ASR33的电传打字机本身是一个硬件设备，在Unix/Linux系统中，这个硬件设备被抽象为“TTY设备”。

2.4 串口终端（Serials Terminal）

早期的TTY终端（这里暂时特指电传打字机），一般通过串口和Unix设备连接的，如下所示：

站内搜索

请输入关键词搜索

功能

留言板
评论列表

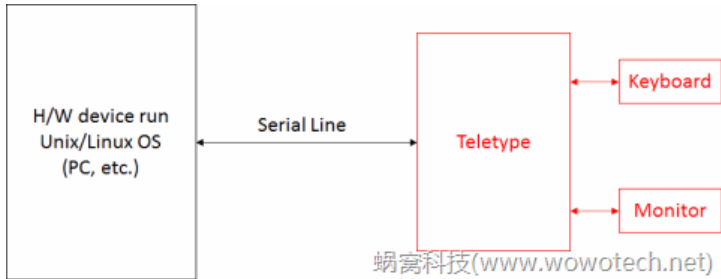
最新评论

luke
因公司网络限制，文章里的图片不能显示，看的好辛苦啊！！
linuxer
@hit201j：已经修改，多谢！
bigpillow
Hi Linuxer, 我们这边有写一套完整的GPIO &...
hit201j
错别字：“还以一个就是bootloader传递过来的bl...
GrayMonkey
膜拜大佬，早找到你的文章就好了,Android开发一枚，一直...
fy
@wowo：是的。我只是就这里的“禁止抢占”非必要，问问wo...

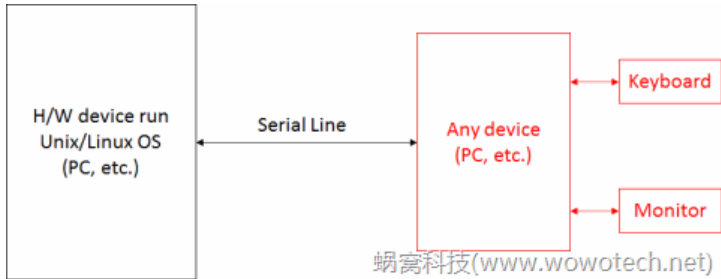
文章分类

- Linux内核分析(9)
- 统一设备模型(13)
- 电源管理系统(42)
- 中断子系统(14)
- 进程管理(13)
- 内核同步机制(17)
- GPIO子系统(5)
- 时间子系统(14)
- 通信类协议(7)
- 内存管理(20)
- 图形子系统(1)
- 文件系统(3)
- TTY子系统(5)
- u-boot分析(3)
- Linux应用技巧(13)
- 软件开发(6)
- 基础技术(13)
- 蓝牙(16)
- ARMv8A Arch(13)
- 显示(3)
- 基础学科(9)
- 技术漫谈(12)
- 项目专区(0)
- X Project(28)

随机文章



然后，正如你所熟知的，我们可以把上面红色部分（电传打字机），替换为任意的具有键盘、显示器、串口的硬件设备（如另一台PC），如下：



因此，对Unix/Linux系统来说，只要是通过串口连接的设备，都可以作为终端设备，因而不需要再关注具体的终端形态。久而久之，终端设备、TTY设备、串口设备等概念，逐渐混在一起，就不再区分了，总结来说，在当今的Linux系统中：

- 1) TTY设备就是终端设备，终端设备就是TTY设备，无需区分。
- 2) 所有的串口设备都是TTY设备。
- 3) 当然，除了串口设备，也发展出来了其它形式的TTY设备，例如虚拟终端（VT）、伪终端（Pseudo Terminal）等等，这些概念本文就不展开描述了，后续会使用专门的文章分析。

3. 控制台（console）

了解了终端和TTY的概念之后，再来看看另一个比较熟悉的概念：console。

回到Unix系统刚刚支持多用户（2.2小节的描述）的时代，此时的PC有一个自带的、昂贵的终端（自身的键盘、显示器等），另外为了支持多用户，可以通过串口线连接多个TTY终端（Teletype）。为了彰显自带终端崇高的江湖地位，人们称它为console。

当然，“江湖地位”之说，纯属玩笑，不过从console的中文翻译——控制台，可以看出，自带终端（console）有别于TTY终端的地方如下：

- 1) 控制台（console）是昂贵的。
- 2) 控制台（console）比TTY终端拥有更多的权限，例如用户建立、密码更改、权限分配等等，这也是“控制”的意义所在。
- 3) 系统的运行日志、出错信息等内容，通常只会输出到控制台（console）终端中，以方便管理员进行“控制”和“管理”。

不过，随着计算机技术的发展、操作系统的改进，控制台（console）终端和普通TTY终端的界限越来越模糊，console能做的事情，普通终端也都能做了。因此，console逐渐退化，以至于在当前的Linux系统中，它仅仅保留了第三点“日志输出”的功能，这就是Linux TTY framework中console的概念（具体可参考后续文章的分析）。

4. 参考文章

[1] <https://en.wikipedia.org/wiki/Teleprinter>
[2] 电传打字机(Teletype), <http://baike.baidu.com/view/1773688.htm>
[3] 你真的知道什么是终端吗？
[4] 串口通信技术浅析

计算机基础知识（四）：动态库和位置无关码
Linux DMA Engine framework(3)_dma controller 驱动
Linux kernel内存管理的基本概念
Why Memory Barriers？中文翻译（上）
Linux时间子系统系列文章之目录

文章存档

- 2017年11月(1)
- 2017年10月(1)
- 2017年9月(5)
- 2017年8月(4)
- 2017年7月(4)
- 2017年6月(3)
- 2017年5月(3)
- 2017年4月(1)
- 2017年3月(8)
- 2017年2月(6)
- 2017年1月(5)
- 2016年12月(6)
- 2016年11月(11)
- 2016年10月(9)
- 2016年9月(6)
- 2016年8月(9)
- 2016年7月(5)
- 2016年6月(8)
- 2016年5月(8)
- 2016年4月(7)
- 2016年3月(5)
- 2016年2月(5)
- 2016年1月(6)
- 2015年12月(6)
- 2015年11月(9)
- 2015年10月(9)
- 2015年9月(4)
- 2015年8月(3)
- 2015年7月(7)
- 2015年6月(3)
- 2015年5月(6)
- 2015年4月(9)
- 2015年3月(9)
- 2015年2月(6)
- 2015年1月(6)
- 2014年12月(17)
- 2014年11月(8)
- 2014年10月(9)
- 2014年9月(7)
- 2014年8月(12)
- 2014年7月(6)
- 2014年6月(6)
- 2014年5月(9)
- 2014年4月(9)
- 2014年3月(7)
- 2014年2月(3)
- 2014年1月(4)



原创文章，转发请注明出处。蜗窝科技，www.wowotech.net。

标签: Linux Kernel 内核 tty terminal 终端 console 控制台



« TLB flush操作 | X-011-UBOOT-使用bootm命令启动kernel(Bubblegum-96平台)»

评论：

feixiahn

2016-12-06 10:59

您好，我想问一个问题，我们如何修改串口工具登录时的root权限的密码呢？
能解释一下如何更改吗？

回复

wowo

2016-12-06 13:21

@feixiahn：是在PC上修改target的密码吗（最好把问题描述清楚;-）？
可以用openssl命令：
openssl passwd -1
输入两次密码后，会生成MD5加密的字符串，然后替换/etc/shadow中对应的字段就可以了。

回复

fexiahn

2016-12-07 14:39

@wowo：@wowo,谢谢，您的文章写的确实不错的

回复

fexiahn

2016-12-07 14:48

@fexiahn：如果将从代码中/etc/passwd路径改为etc_us/passwd，那么修改用户名的密码时是不是
要将/etc/shadow的路径一同修改，我最近在看这部分资料，没有搞明白/etc/passwd
和/etc/shadow之间的关系，麻烦wowo帮忙分析一下，谢谢

回复

wowo

2016-12-08 09:23

@fexiahn：用户空间的东西，我也是一知半解，一时半会儿没办法给你讲啊，抱歉哈~~

回复

笨小孩

2016-10-26 21:22

窝窝你好，想问一下，在linux的U-BOOT启动时，有一个CONSOLE = ttySA0,在启动后，信息会通过串口打印到屏幕
上，那应该这个ttySA0应该代表的是控制台吧，为何在进行其他的操作时，却要求提升用户权限呢？这又是为什么？

回复

wowo

2016-10-26 21:50

@笨小孩：是的，console指定的是控制台。
对于权限的事情，是这样的：
用户空间程序是通过文件系统（字符设备，/dev/ttySA0）访问TTY设备的，只要在用户空间牵涉到文件操作，
都有权限的概念。

回复

笨小孩

2016-10-27 08:50

@wowo：谢谢窝窝！

回复

lucifer

2016-10-04 15:51

还有就是 类似这种链表操作比较多的。
static int sysfs_readdir(struct file * filp, void * dirent, filldir_t filldir)
366{
.....
370 struct list_head *p, *q = &cursor->s_sibling;
371 ino_t ino;
372 int i = filp->f_pos;
373
374*/
389 default:
390 if (filp->f_pos == 2) {

```
391         list_del(q);
392         list_add(q, &parent_sd->s_children);
393     }
394     for (p=q->next; p!= &parent_sd->s_children; p=p->next) {
395         struct sysfs_dirent *next;
396         const char * name;
397         int len;
398
399         next = list_entry(p, struct sysfs_dirent,
400             s_sibling);
401         if (!next->s_element)
402             continue;
403
404         name = sysfs_get_name(next);
405         len = strlen(name);
406         if (next->s_dentry)
407             ino = next->s_dentry->d_inode->i_ino;
408         else
409             ino = iunique(sysfs_sb, 2);
410
411         if (filldir(dirent, name, len, filp->f_pos, ino,
412             dt_type(next)) < 0)
413             return 0;
414
415         list_del(q);
416         list_add(q, p);
417         p = q;
418         filp->f_pos++;
419     }
420 }
421 return 0;
422}
```

=====

我读起来迷迷糊糊的

[回复](#)

lucifer

2016-10-04 15:50

请教大奖一个基础的C语言的问题或者说对指针的理解吧
我在2.6.10中看到
#define subsys_set_kset(obj,subsys) (obj)->subsys.kset.kobj.kset = &(_subsys).kset
还有down_write(&kobj->kset->subsys->rwsem);
怎么理解这么多的->或者. --- 什么时候该用 '.'
什么时候用->

[回复](#)

wowo

2016-10-04 16:28

@lucifer : ->是指针，例如struct device *p;p->name;
.是变量，例如struct device aa; aa.name;

[回复](#)

electrlife

2016-09-22 15:11

这篇文章来的太及时了，前两天正准备部关于tty的相关东西，有一些概念还没有理清楚：
terminal console vt tty pseudo terminal等，再加上control terminal, session等
纠缠在一起！

[回复](#)

维尼

2016-09-20 16:35

内核 /dev下已经有好多TTY了 tty又演变成驱动了

[回复](#)

熊猫盼盼

2016-09-21 15:41

@维尼：这段时间也一直再看tty相关的东西，起因是在弄lxc/docker时，不明白 l x c/docer的 c o n s o l e /tty 是怎么实现的。正如w o w o所说的， t t y是unix/linux 下一个既熟悉又陌生的东西。 非常期待w o w o后续的文章， 如果能把 设备下的文件 /dev/console /dev/tty /dev/ttyN /dev/pty 之间的关系 以及实现讲清楚， 那就更好了！

[回复](#)

wowo

2016-09-21 17:17

@熊猫盼盼：多谢建议，为我写后续的文章指了一条明路:-)	回复
wink 2016-11-20 21:22	
@熊猫盼盼：我也是啊，	回复

发表评论：

昵称

邮件地址 (选填)

个人主页 (选填)

wLhGB

发表评论