

更多嵌入式 Linux 学习资料, 请关注: 一口 Linux 回复关键字:1024



windows 提供了计划任务这一功能, 在控制面板 -> 性能与维护 -> 任务计划, 它的功能就是安排自动运行的任务。通过'添加任务计划'的一步步引导, 则可建立一个定时执行的任务。

在 linux 同样具备相同的功能:

第一部分: at 命令

at 命令: 在一个指定的时间执行一个指定任务, 只能执行一次, 且需要开启 atd 进程 (ps -ef | grep atd 查看,

开启用/etc/init.d/atd start or restart; 开机即启动则需要运行 chkconfig --level 2345 atd on)。

[root @ localhost test]at now +1 min

at> echo "hello" > /dev/tty2 后回车 # 一分钟发送"hello"给 tty2 终端

--注意: 使用 ctrl+d 来终结字符输入 (即<EOT>)

at 00:55 2009-12-13

at> echo "test at command" > /home/test.txt

at 1:00 2009-12-13

at>/sbin/shutdown -h now #此处使用绝对路径

at 命令创建的任务会放在/var/spool/at 下面, 是一个脚本文件, 可以用 vi 进行编辑。

at now +1 min

at> echo "hello" <EOF>#此时会以发送邮件的方式给当前执行的人, 用 mail 命令可以查看, 即先使用 mail 命令, 出现&后输入'1'则可查看。

atq 指令: 查询当前机器上有哪些等待执行的任务, 其第一列为工作号。

atrm 指令: 取消任务, eg: atrm 10 #其中 10 为工作号。

batch 指令: 以一个较低优先级延时执行任务。

batch

at> echo "hello" > /home/abc.txt <EOF>

2.at 命令的安全控制

/etc/at.allow #列出可以使用 at 命令的用户, 先找 at.allow 文件,再找 at.deny 文件

/etc/at.deny #列出不可以使用 at 命令的用户, 一行一个用户。

若既无 at.allow 文件, 也无 at.deny 文件, 则只有 root 用户才能执行 at 命令

第二部分: cron 服务机制

1、cron 服务介绍

cron 把命令行保存在/etc/crontab 文件里, 每个系统用户如果设置了自己的 cron, 那都会在/var/spool/cron 下面有对应用户名的 crontab。无论编写/var/spool/cron 目录内的文件还是/etc/crontab 文件, 都能让 cron 准确无误地执行安排的任务, 区别是/var/spool/cron 下各系统用户的 crontab 文件是对应用户级别的任务配置, 而/etc/crontab 文件则是对应系统级别的任务配置。cron 服务器每分钟读取一次/var/crontab/cron 目录内的所有文件和/etc/crontab 文件。

2、使用 cron 服务执行计划任务

```
#service crond start/restart/stop
```

启动/重启/关闭 cron 服务

```
#crontab -l
```

浏览当前用户的 crontab, 即浏览已存在的计划任务列表

```
#crontab -e
```

编辑当前用户 crontab, 如之前从未编辑过 crontab 文件, 那么 crontab 是一个空白文件, 我们可以完全手工编写我们所需要的计划任务列表, 下面详细介绍计划任务列表的写法。

以下是几个例子:

```
9 9 * * * /usr/local/bin/cvsb
```

```
3 3 * * 0 /usr/local/bin/qbbak
```

```
* */6 * * * /usr/local/bin/esbbak
```

```
20,30 * * * /usr/local/bin/esbak
```

六个字段对应的含义如下:

```
9      9      *      *      *
```

分钟 小时 日期 月份 星期

表示每天早上 9 点 9 分执行目录/usr/local/bin/中的 cvsb 文件

表示每周日凌晨 3 时 3 分执行目录/usr/local/bin 中的 qbbak 文件

表示每 6 小时执行一次, 执行时间从第一次执行起计算, 当然也可以自定义执行时间, 比如

0 */6 * * * 那程序就会在整点执行
表示在每小时 20 分和 30 分时执行

上面简单介绍了 cron 的一些基本用法, 这个东东的时间组合可不是一般的强 ~ 可以复杂到你无法想象, 我们只要掌握少许基本就可以满足日常需求了。

3、cron 进阶使用

(1) .避免 cron 产生垃圾文件

细心的朋友可能会发现系统/var/spool/clientmqueue/目录下往往存有大量文件, 原因是系统中用户开启了 cron, 而 cron 中执行的程序有输出内容, 输出内容会以邮件形式发给 cron 的用户, 而 sendmail (系统自带的邮件服务器) 没有启动所以就产生了这些文件。

这时可以使用输出重定向, 即在计划任务后加上 > /dev/null 2>&1

例:

```
9 9 * * * /usr/local/bin/cvsb > /dev/null 2>&1
```

(2>&1: 把错误重定向到输出要送到的地方。)

把命令的执行结果重定向到/dev/null, 即把产生的错误抛弃。

(2) .一种经常碰到的情况, 需要执行的文件没问题, 但通过 cron 就是死活不能正确执行, 这时我们可以通过输出重定向将 cron 的执行过程输出到一个文件, 通过分析执行过程来寻找错误的原因

例:

```
9 9 * * * /usr/local/bin/cvsb > /var/log/crontab_log 2>&1
```

上例将 cron 的执行过程输出到一个文本文件 crontab_log, 这个文件可以随便手动建一个, 放在哪都行, 重定向目录写对就 Ok