inotifywait

异步文件系统监控机制

补充说明

Inotify 一种强大的、细粒度的、异步文件系统监控机制，它满足各种各样的文件监控需要，可以监控文件系统的访问属性、读写属性、权限属性、删除创建、移动等操作，也就是可以监控文件发生的一切变化。。

inotify-tools 是一个C库和一组命令行的工作提供Linux下inotify的简单接口□inotify-tools安装后会得到`inotifywait`和`inotifywatch`这两条命令：

- **inotifywait命令** 可以用来收集有关文件访问信息□Linux发行版一般没有包括这个命令，需要安装inotify-tools□这个命令还需要将inotify支持编译入Linux内核，好在大多数Linux发行版都在内核中启用了inotify□
- **inotifywatch命令** 用于收集关于被监视的文件系统的统计数据，包括每个 inotify 事件发生多少次。

开始之前需要检测系统内核是否支持inotify□

使用`uname -r`命令检查Linux内核，如果低于2.6.13，就需要重新编译内核加入inotify的支持。

使用`ll /proc/sys/fs/inotify`命令，是否有以下三条信息输出，如果没有表示不支持。

```
ll /proc/sys/fs/inotify
total 0
-rw-r--r-- 1 root root 0 Jan  4 15:41 max_queued_events
-rw-r--r-- 1 root root 0 Jan  4 15:41 max_user_instances
-rw-r--r-- 1 root root 0 Jan  4 15:41 max_user_watches
```

安装inotify-tools

- inotify-tools项目地址：https://github.com/rvoicilas/inotify-tools
- inotify-tools下载地址：http://github.com/downloads/rvoicilas/inotify-tools/inotify-tools-3.14.tar.gz

```
#CentOS release 5.8/64位：
tar zxvf inotify-tools-3.14.tar.gz
cd inotify-tools-3.14
./configure
make
make install
```

其他Linux发行版安装方法可以参见：https://github.com/rvoicilas/inotify-tools/wiki#wiki-getting

inotify相关参数

inotify定义了下列的接口参数，可以用来限制inotify消耗kernel memory的大小。由于这些参数都是内存参数，因此，可以根据应用需求，实时的调节其大小：

- `/proc/sys/fs/inotify/max_queued_evnets`表示调用inotify_init时分配给inotify instance中可排队的event的数目的最大值，超出这个值的事件被丢弃，但会触发IN_Q_OVERFLOW事件。

- /proc/sys/fs/inotify/max_user_instances表示每一个real user id可创建的inotify instatnces的数量上限。
- /proc/sys/fs/inotify/max_user_watches表示每个inotify instatnces可监控的最大目录数量。如果监控的文件数目巨大，需要根据情况，适当增加此值的大小。

根据以上在32位或者64位系统都可以执行：

```
echo 104857600 > /proc/sys/fs/inotify/max_user_watches
echo 'echo 104857600 > /proc/sys/fs/inotify/max_user_watches' >>
/etc/rc.local
```

如果遇到以下错误：

```
inotifywait: error while loading shared libraries: libinotifytools.so.0:
cannot open shared object file: No such file or directory
```

```
**解决方法：**
32位系统ln -s /usr/local/lib/libinotifytools.so.0
/usr/lib/libinotifytools.so.0
64位系统ln -s /usr/local/lib/libinotifytools.so.0
/usr/lib64/libinotifytools.so.0
```

inotifywait命令使用

```
#!/bin/bash
#filename watchdir.sh
path=$1
/usr/local/bin/inotifywait -mrq --timefmt '%d/%m/%y/%H:%M' --format '%T %w
%f' -e modify,delete,create,attrib $path

执行输出：
./watchdir.sh /data/wsdata/tools/
04/01/13/16:34 /data/wsdata/tools/ .j.jsp.swp
04/01/13/16:34 /data/wsdata/tools/ .j.jsp.swx
04/01/13/16:34 /data/wsdata/tools/ .j.jsp.swx
04/01/13/16:34 /data/wsdata/tools/ .j.jsp.swp
04/01/13/16:34 /data/wsdata/tools/ .j.jsp.swp
04/01/13/16:34 /data/wsdata/tools/ .j.jsp.swp
04/01/13/16:34 /data/wsdata/tools/ .j.jsp.swp
04/01/13/16:34 /data/wsdata/tools/ .j.jsp.swp
04/01/13/16:35 /data/wsdata/tools/ 4913
04/01/13/16:35 /data/wsdata/tools/ 4913
04/01/13/16:35 /data/wsdata/tools/ 4913
04/01/13/16:35 /data/wsdata/tools/ j.jsp
04/01/13/16:35 /data/wsdata/tools/ j.jsp
04/01/13/16:35 /data/wsdata/tools/ j.jsp
04/01/13/16:35 /data/wsdata/tools/ j.jsp~
04/01/13/16:35 /data/wsdata/tools/ .j.jsp.swp
```

inotifywait命令参数

- -m是要持续监视变化。
- -r使用递归形式监视目录。
- -q减少冗余信息，只打印出需要的信息。
- -e指定要监视的事件列表。
- --timefmt是指定时间的输出格式。
- --format指定文件变化的详细信息。

可监听的事件

access	**访问**	读取文件。
modify	**修改**	文件内容被修改。
attrib	**属性**	文件元数据被修改。
move	**移动**	对文件进行移动操作。
create	**创建**	生成新文件
open	**打开**	对文件进行打开操作。
close	**关闭**	对文件进行关闭操作。
delete	**删除**	文件被删除。