Linux下如何指定某一类型程序用特定程序打开（通过binfmt_misc）

原创

镇上村树 于 2018-10-27 08:36:11 发布 2921 收藏 4

分类专栏： linux 文章标签： Linux

本文链接：https://blog.csdn.net/realDonaldTrump/article/details/83443187

文章目录

概述

在Windows平台上，文件系统中的文件可以拥有特定的扩展名，系统根据不同的扩展名选择使用特定的程序打开。

在Linux平台上，也提供了类似的功能，甚至从某种意义上来说更加的强大，只不过没有windows那么直观那么浅显。Linux的内核从很早开始就引入了一个叫做Miscellaneous Binary Format（binfmt_misc）的机制，可以通过要打开文件的特性来选择到底使用哪个程序来打开，该机制根据文件的元属性来选择合适的程序将其打开。比Windows更加强大的地方是，它不光光可以通过文件的扩展名来判断，还可以通过文件开始位置的特殊的字节（Magic Byte），即文件的元数据、文件的格式头来判断。

binfmt_misc

- 根据扩展名选择程序打开文件
- 根据文件元属性来选择程序打开文件

如果要使用这个功能的话，首先要绑定binfmt_misc，可以通过以下命令来绑定：

```
mount binfmt_misc -t binfmt_misc /proc/sys/fs/binfmt_misc
```

这样绑定的话，系统重新启动之后就失效了。如果想让系统每次启动的时候都自动绑定的话，可以往/etc/fstab文件中加入下面这行：

```
none  /proc/sys/fs/binfmt_misc binfmt_misc defaults 0 0
```

绑定完之后，就可以通过向/proc/sys/fs/binfmt_misc/register（这个文件只能写不能读）文件中写入一行匹配规则字符串来告诉内核什么样的程序要用什么样的程序打开（一般使用echo命令）。

这行字符串的格式如下：

```
:name:type:offset:magic:mask:interpreter:flags
```

每个字段都用冒号":"分割。某些字段拥有默认值，或者只在前面字段被设置成了某个特定值后才有效，因此可以跳过某些字段的设置，但是必须保留相应的冒号分割符。各个字段的意义如下：

1）name：这个规则的名字，理论上可以取任何名字，只要不重名就可以了。但是为了方便以后维护一般都取一个有意义的名字，比如表示被打开文件特性的名字，或者要打开这个文件的程序的名字等；

2）type：表示如何匹配被打开的文件，只可以使用"E"或者"M"，只能选其一，两者不可共用。"E"代表只根据待打开文件的扩展名来识别，而"M"表示只根据待打开文件特定位置的几位魔数（Magic Byte）（==即文件的元属性，通过魔数来表示文件的格式头，可以用来识别文件）==来识别；

3）offset：这个字段只对前面type字段设置成"M"之后才有效，它表示从文件的多少偏移开始查找要匹配的魔数。如果跳过这个字断不设置的话，默认就是0；（因为不同平台的文件的魔数可能不是从同一个位置开始的，因此可以设置偏移值来从不同的位置去取得魔数）

4）magic：它表示真正要匹配的魔数，如果type字段设置成"M"的话；或者表示文件的扩展名，如果type字段设置成"E"的话。对于匹配魔数来说，如果要匹配的魔数是ASCII码可见字符，可以直接输入，而如果是不可见的话，可以输入其16进制数值，前面加上"\x"或者"\x"（如果在Shell环境中的话。对于匹配文件扩展名来说，就在这里写上文件的扩展名，但不要包括扩展名前面的点号（"."），且这个扩展名是大小写敏感的，有些特殊的字符，例如目录分隔符正斜杠（"/"）是不允许输入的；

5）mask：同样，这个字段只对前面type字段设置成"M"之后才有效。它表示要匹配哪些位，它的长度要和magic字段魔数的长度一致。如果某一位为1，表示这一位必须要与magic对应的位匹配；如果对应的位为0，表示忽略对这一位的匹配，取什么值都可以。如果是0xff的话，即表示全部位都要匹配，默认情况下，如果不设置这个字段的话，表示要与magic全部匹配（即等效于所有都设置成0xff）。还有同样对于NUL来说，要使用转义（\x00），否则对这行字符串的解释将到NUL停止，后面的不再起作用；

6）interpreter：解释器，表示要用哪个程序来启动这个类型的文件，一定要使用全路径名，不要使用相对路径名；

7）flags：这个字段可选，主要用来控制interpreter打开文件的行为。比较常用的是'P'（请注意，一定要大写），表示保留原始的argv[0]参数。这是什么意思呢？默认情况下，如果不设置这个标志的话，binfmt_ misc会将传给interpreter的第一个参数，即argv[0]，修改成要被打开文件的全路径名。当设置了'P'之后，binfmt_misc会保留原来的argv[0]，在原来的argv[0]和argv[1]之间插入一个参数，用来存放要被打开文件的全路径名。比如，如果想用程序/bin/foo来打开/usr/local/bin/blah这个文件，如果不设置'P'的话，传给程序/bin/foo的参数列表argv[]是["/usr/local/bin/blah", "blah"]，而如果设置了'P'之后，程序/bin/foo得到的参数列表是["/bin/foo", "/usr/local/bin/blah", "blah"]。

除了以上的规则之外，还有一些额外的限制条件：

1）每一行匹配规则字符串的长度不能超过1920个字符；

2）魔数（Magic Byte）必须在文件头128个字节内，也就是说offset+sizeof(magic)不能超过128；

3）interpreter字段的长度不能超过127个字符。

如何判断本机已经生效的文件打开规则

每次成功写入一行规则，都会在/proc/sys/fs/binfmt_misc/目录下，创建一个名字为输入的匹配规则字符串中"name"字段的文件。

通过读取这个文件的内容，可以知道这条匹配规则当前的状态：

```
cat /proc/sys/fs/binfmt_misc/<name>
```

开关已有的规则

而通过向这个文件中写入0或1，可以关闭或打开这条匹配规则，而写入-1表示彻底删除这条规则：

```
echo 0 > /proc/sys/fs/binfmt_misc/<name>    # Disable the match
echo 1 > /proc/sys/fs/binfmt_misc/<name>    # Enable the match
echo -1 > /proc/sys/fs/binfmt_misc/<name>   # Delete the match
```

> 确保具有root权限写入。

一键启停bingmt_misc

在/proc/sys/fs/binfmt_misc/目录下，还缺省存在一个叫做status的文件，通过它可以查看和控制整个binfmt_misc的状态，而不光是单个匹配规则。

查看当前binfmt_misc是否处于打开状态：

```
cat /proc/sys/fs/binfmt_misc/status
```

也可以通过向它写入1或0来打开或关闭binfmt_misc：

```
echo 0 > /proc/sys/fs/binfmt_misc/status    # Disable binfmt_misc
echo 1 > /proc/sys/fs/binfmt_misc/status    # Enable binfmt_misc
```

如果想删除当前binfmt_misc中的所有匹配规则，可以向其传入-1：

```
echo -1 > /proc/sys/fs/binfmt_misc/status    # Disable all matches
```

原文：https://blog.csdn.net/roland_sun/article/details/50062295?utm_source=copy