(2) GET http://www.foo.com/subdir/index.html
获取 subdir/index.html，找到指向 subdir/logo.gif 的链接。

(3) GET http://www.foo.com/subdir/logo.gif
220 获取 subdir/logo.gif，再没有链接了，结束。

但在图 9-3b 的文件系统中，可能会发生下列情况：

(1) GET http://www.foo.com/index.html
获取 /index.html，找到指向 subdir/index.html 的链接。

(2) GET http://www.foo.com/subdir/index.html
获取 subdir/index.html，但得到的还是同一个 index.html。

(3) GET http://www.foo.com/subdir/subdir/index.html
获取 subdir/subdir/index.html。

(4) GET http://www.foo.com/subdir/subdir/subdir/index.html
获取 subdir/subdir/subdir/index.html。

图 9-3b 的问题是 subdir/ 是个指向“/”的环路，但由于 URL 看起来有所不同，所以机器人无法单从 URL 本身判断出文档是相同的。毫无戒备的机器人就有了陷入循环的危险。如果没有某种循环检测方式，这个环路就会继续下去，通常会持续到 URL 的长度超过机器人或服务器的限制为止。

9.1.9 动态虚拟Web空间

恶意网管可能会有意创建一些复杂的爬虫循环来陷害那些无辜的、毫无戒备的机器人。尤其是，发布一个看起来像普通文件，实际上却是网关应用程序的 URL 是很容易的。这个应用程序可以在传输中构造出包含了到同一服务器上虚构 URL 链接的 HTML。请求这些虚构的 URL 时，这个邪恶的服务器就会捏造出一个带有新的虚构 URL 的新 HTML 页面来。

即使这个恶意 Web 服务器实际上并不包含任何文件，它也可以通过无限虚拟的 Web 空间将可怜的机器人带入爱丽丝漫游仙境之旅。更糟的是，每次的 URL 和 HTML 看起来都有很大的不同，机器人很难检测到环路。图 9-4 显示了一个恶意 Web 服务
221 器生成假内容的例子。

更常见的情况是，那些没有恶意的网管们可能会在无意中通过符号连接或动态内容构造出爬虫陷阱。比如，我们来看一个基于 CGI 的日历程序，它会生成一个月历和