9.4.4 其他有关robots.txt的知识

解析 robots.txt 文件时还需遵循其他一些规则。

- 随着规范的发展，robots.txt 文件中可能会包含除了 `User-Agent`、`Disallow` 和 `Allow` 之外的其他字段。机器人应该将所有它不理解的字段都忽略掉。
- 为了实现后向兼容，不能在中间断行。
- 注释可以出现在文件的任何地方；注释包括可选的空格，以及后面的注释符（#）、注释符后面的注释，直到行结束符为止。
- 0.0 版的拒绝机器人访问标准并不支持 `Allow` 行。有些机器人只实现了 0.0 版的规范，因此会忽略 `Allow` 行。在这种情况下，机器人的行为会比较保守，有些允许访问的 URL 它也不去获取。

9.4.5 缓存和robots.txt的过期

如果一个机器人在每次访问文件之前都要重新获取 robots.txt 文件，Web 服务器上的负载就会加倍，机器人的效率也会降低。机器人使用的替代方法是，它会周期性地获取 robots.txt 文件，并将得到的文件缓存起来。机器人会使用这个 robots.txt 文
234 件的缓存副本，直到其过期为止。原始服务器和机器人都会使用标准的 HTTP 缓存控制机制来控制 robots.txt 文件的缓存。机器人应该留意 HTTP 响应中的 `Cache-Control` 和 `Expires` 首部。[22]

现在很多产品级爬虫都不是 HTTP/1.1 的客户端，网管应该意识到这些爬虫不一定能够理解那些为 robots.txt 资源提供的缓存指令。

如果没有提供 `Cache-Control` 指令，规范草案允许将其缓存 7 天。但实际上，这个时间通常太长了。不了解 robots.txt 文件的 Web 服务器管理员通常会在响应机器人的访问时创建一个新的文件，但如果将缺乏信息的 robots.txt 文件缓存一周，新创建的 robots.txt 文件就没什么效果了，站点管理员会责怪机器人管理员没有遵守拒绝机器人访问标准。[23]

9.4.6 拒绝机器人访问的Perl代码

有几个公共的 Perl 库可以用来与 robots.txt 文件进行交互。CPAN 公共 Perl 文档中的 `WWW::RobotsRules` 模块就是一个这样的例子。

注 22：更多有关缓存指令处理方面的内容请参见 7.8 节。

注 23：有几种大型的 Web 爬虫，如果它们在 Web 上勤奋爬行的话，每天都会重新获取 robots.txt。