

shellesc 宏包^{*}

L^AT_EX 项目组 【著】

张泓知 【译】

2023 年 7 月 8 日

1 引言

多年来，基于 web2c 的 T_EX 实现一直使用 `\write` 命令的语法来访问系统命令，它使用了特殊的 18 号流（15 号及以上的流不能分配给经典 T_EX 中的文件，所以 18 号流本应只是输出到终端）。

这是一个有用的扩展，没有违反经典 T_EX 中关于扩展的严格规则。该宏包提供了一个简单的宏级接口来隐藏 `write18` 的实现，因此可以使用类似 `\ShellEscape{rm file.txt}` 的命令在类 Unix 系统中指定删除文件（在 Windows 中使用 `del`）。请注意，默认情况下禁止访问系统，并且通常需要使用命令行选项 `--shell-escape` 调用 L^AT_EX。

该包可以与标准的 `latex`、`pdflatex` 或 `xetex` 一起使用，但它主要是为了 `lualatex` 而设计的，因为从 LuaT_EX 0.87 开始，LuaT_EX 不再支持 `\write18` 语法来访问系统命令：它具有 256 个写入流，流 18 可以关联到文件（没有此包时）并且没有特殊意义。这个包在 LuaL^AT_EX 中定义了相同的 `\ShellEscape` 语法，但实现是通过 Lua 和 `os.execute` 函数完成的。

实际上，`\ShellEscape` 对应于 `\immediate\write18`（或 `\directlua`）。极少数情况下，您可能需要延迟系统命令直到输出当前页面（当页码已知时），对于这种情况，您可以经典地使用 `\write18`（或 `\latelua`）。此包提供了 `\DelayedShellEscape` 作为这种用法的常用语法。

可以通过检查整数 `(chardef)` 命令 `\ShellEscapeStatus` 查询 shell escape 状态，0（禁用）、1（启用）、2（受限）。

为了帮助将现有文档移植到 LuaT_EX 0.87，该包重载了 `\write` 命令，以便 `\write18{rm file.txt}` 在 LuaT_EX 中起作用。请注意，`\write` 的重新定义无法检测出是否已使用 `\immediate`，当写入文件流或终端时，`\immediate` 会正常工作，但流 18 的特殊情况，它被定义为始终使用 `os.execute`，始终使用 `\directlua`（因此对应于 `\immediate\write18`）。在需要将非即时

^{*}本文件版本号为 v1.0d，最后修改于 2023 年 7 月 8 日。

`\write18` 转换为当前 Lua_T_EX 的文档中的极少情况下，您需要更改为使用 `\DelayedShellEscape` 命令。

2 代码实现

```
1 (*package)
2 \chardef\shellesc@quotecat\catcode`\ "
3 \chardef\shellesc@underscorecat\catcode`\_
4 \@makeoother\ "
5 \@makeoother\ _
```

2.1 状态检查

2.2 shellesc 宏包接口

`\ShellEscapeStatus` 整数值的含义分别为 0（禁用 shell escape）、1（允许 shell escape）、2（受限的 shell escape）。

```
6 \chardef\ShellEscapeStatus
7 \ifx\pdfshellescape\@undefined
8 \ifx\shellescape\@undefined
9 \ifx\directlua\@undefined
10 \z@
11 \else
12 \directlua{%
13 tex.sprint((status.shell_escape or os.execute()) .. " ")}
14 \fi
15 \else
16 \shellescape
17 \fi
18 \else
19 \pdfshellescape
20 \fi

21 \ifcase\ShellEscapeStatus
22 \PackageWarning{shellesc}{Shell escape disabled}
23 \or
24 \PackageInfo {shellesc}{Unrestricted shell escape enabled}
25 \else
26 \PackageInfo {shellesc}{Restricted shell escape enabled}
27 \fi
```

`\ShellEscape` 执行提供的记号作为一个系统相关命令，假设允许这样的执行。

```
28 \ifx\lastsavedimageresourcepages\@undefined
29   \protected\def\ShellEscape{\immediate\write18 }
30 \else
31   \protected\def\ShellEscape{\directlua\ShellEscape@Lua}
32 \fi
```

`\DelayedShellEscape` 当此节点随着完成的页面输出时，将提供的记号作为一个系统相关命令执行，假设允许这样的执行。

```
33 \ifx\lastsavedimageresourcepages\@undefined
34   \protected\def\DelayedShellEscape{\relax\write18 }
35 \else
36   \protected\def\DelayedShellEscape{\latelua\ShellEscape@Lua}
37 \fi
```

`\ShellEscape@Lua` `\DelayedShellEscape` 和 `\ShellEscape` 的共享 Lua 代码。

```
38 \ifx\directlua\@undefined\else
39 \protected\def\ShellEscape@Lua#1{%
40 local status, msg = os.execute("\luaescapestring{#1}")%
41 if status == nil then
42   texio.write_nl("log",%
43     "runsystem(" .. "\luaescapestring{#1}"%
44     .. ")...( " .. msg .. " )\string\n")
45 elseif status == 0 then
46   texio.write_nl("log",%
47     "runsystem(" .. "\luaescapestring{#1}"%
48     .. ")...executed.\string\n")
49 else
50   texio.write_nl("log",%
51     "runsystem(" .. "\luaescapestring{#1}"%
52     .. ")...failed. " .. (msg or "") .. "\string\n")
53 end
54 }}
55 \fi
```

2.3 write18 宏包接口

在除 Lua_T_EX 外的基于 web2c 的引擎中，可以直接使用 `\write18`。在较旧的 Lua_T_EX 中也是如此，但从版本 0.85 开始，这不再可用。

上述的 `shellesc` 宏包接口推荐用于新代码，然而，为了方便将现有文档和包移植到更新的 Lua_{TeX} 版本，这里通过调用 Lua 的 `os.execute` 提供了一个 `\write18` 接口。

请注意，目前的写法总是对系统进行立即调用。

`\immediate` 是被支持但被忽略的，`\immediate\write18` 和 `\write18` 都会立即执行。要在下一个输出时延迟执行，请使用上面定义的 `\DelayedShellEscape` 命令。

请注意，可以很容易地使此处定义的 `\write18` 使用延迟执行，只需在下面的定义中使用 `\DelayedShellEscape` 而不是 `ShellEscape`。然而，检测 `\immediate` 是有技巧性的，因此这里的选择是始终使用即时形式，这在与 `\write18` 一起使用时是绝大多数情况下使用的形式。

如果不是最近的 Lua_{TeX}，请在此处停止。

```
56 \ifx\lastsavedimageresourcepages\@undefined
57 \catcode`\"\shellesc@quotecat
58 \catcode`\_\shellesc@underscorecat
59 \expandafter\endinput
60 \fi

61 \directlua{%

62 shellesc = shellesc or {}
```

使用记号扫描器获取下一个 _{TeX} 数字的 Lua 函数，并测试是否正在使用流 18，然后在每种情况下插入适当的 _{TeX} 命令来处理随后的大括号组。

```
63 local function write_or_execute()
64   local s = token.scan_int()
65   if (s==18) then
66     tex.sprint(\the\numexpr\catcodetable@atletter\relax,
67               "\string\\ShellEscape ")
68   else
69     tex.sprint(\the\numexpr\catcodetable@atletter\relax,
70               "\string\\shellesc@write " .. s)
71   end
72 end

73 shellesc.write_or_execute=write_or_execute

74 }

75 \let\shellesc@write\write

76 \protected\def\write{\directlua{shellesc.write_or_execute()}}
```

```
77 \catcode`\\"shellesc@quotecat
78 \catcode`\_shellesc@underscorecat

79 \</package>
```