`pathsuris.sty`: Paths and URIs for sTeX*

Jinbo Zhang, Michael Kohlhase
Jacobs University, Bremen

February 18, 2016

Abstract

This package provides macros to deal with paths and base URIs for sTeX. In particular, it offers a path canonicalizer, which is used in package `modules`, in order to support modules specified with relative path.

Contents

*Version v1.1 (last revised 2016/02/18)

1 Usage

By calling `\@cpath{`⟨*path*⟩`}`, the canonicalized path will be stored in `\@CanPath`. To print a canonicalized path, simply use `\cpath{`⟨*path*⟩`}`.

2 Examples

path	canonicalized path
aaa	aaa
../../aaa	../../aaa
aaa/bbb	aaa/bbb
aaa/..	
../../aaa/bbb	../../aaa/bbb
../aaa/../bbb	../bbb
../aaa/bbb	../aaa/bbb
aaa/bbb/../ddd	aaa/ddd
aaa/bbb/../..	

3 The Implementation

```
\RequirePackage{xstring}
\RequirePackage{forloop}
\RequirePackage{calc}
```

We first create some counters. `AddrNum` will count the number of sections in the input path, `iLoop` will be used as the loop iterator, `iName` will be used for generating names such as `Addri`, `Addrii`, `RealAddrNum` will count the number of sections in the canonicalized path, `Cutable` will count the number of sections besides `...`

```
\newcounter{AddrNum}
\newcounter{iLoop}
\newcounter{iName}
\newcounter{RealAddrNum}
\newcounter{Cutable}
```

We define two macros for later comparison.

```
\def\@ToTop{..}
\def\@Slash{/}
```

Then we split the input path.

```
\def\@MultiAddrs#1/#2\@nil{%
  \def\CurArg{#1}%
  \def\NextArg{#2}%
  \ifx\@empty\CurArg% for the first one
  \else%
    \stepcounter{AddrNum}%
    \expandafter\edef\csname Addr\roman{AddrNum}\endcsname{#1}% storing
  \fi%
  \ifx\@empty\NextArg% for the last one
```

```
    \let\next\@gobble%
  \fi%
  \next#2\@nil% recursion
}%
```

Implement `\@cpath`.

```
\def\@cpath#1{%
  \let\next\@MultiAddrs%
  \setcounter{AddrNum}{0}%
  \setcounter{iLoop}{0}%
  \setcounter{iName}{0}%
  \setcounter{RealAddrNum}{0}%
  \setcounter{Cutable}{0}%
  \def\@CurrPath{}%
  \def\@CanPath{}%
  \def\@TempPath{}%
  \def\@Rubbish{}%
  \expandafter\next#1/\@nil% recursion starts
  \forloop{iLoop}{0}{\value{iLoop} < \value{AddrNum}}{%
    \stepcounter{iName}%
    \edef\@CurrPath{\csname Addr\roman{iName}\endcsname}%
    \ifx\@CurrPath\@ToTop%
  \ifnum\value{Cutable} = 0%
        \edef\@CanPath{\@CanPath\csname Addr\roman{iName}\endcsname/}%
        \stepcounter{RealAddrNum}%
      \else%
        % cut the last part, and add a slash at the end
        \StrCut[\value{RealAddrNum}]{/\@CanPath}{/}\@TempPath\@Rubbish%
        \StrCut[1]{\@TempPath/}{/}\@Rubbish\@CanPath%
        \addtocounter{RealAddrNum}{-1}%
        \addtocounter{Cutable}{-1}%
      \fi%
    \else%
      \edef\@CanPath{\@CanPath\csname Addr\roman{iName}\endcsname/}%
      \stepcounter{RealAddrNum}%
      \stepcounter{Cutable}%
    \fi%
  }%
  \StrCut[\value{RealAddrNum}]{\@CanPath}{/}\@CanPath\@Rubbish% cut last /
}%
```

Implement `\cpath` to print the canonicalized path.

```
\newcommand\cpath[1]{% print canonical path
\@cpath{#1}%
\@CanPath%
}%
```