The import package

Donald Arseneau (`asnd@triumf.ca`)

Version 6.2,   01–Apr–2020

This software is in the public domain; free of any restrictions.

Two new LaTeX commands:

> `\import` {⟨*full-path*⟩} {⟨*file*⟩}   and
> `\subimport` {⟨*path-extension*⟩} {⟨*file*⟩}

are defined to input a file from another directory, allowing that file to find its own inputs (using `\input`, `\includegraphics` etc.) in that directory.

Alias command names are "`\inputfrom`" and "`\subinputfrom`". (If `\import` was defined previously, it will not be redefined.) Also provided are the similar commands "`\includefrom`" and "`\subincludefrom`", which are based on the `\include` command, rather than `\input`.

For example, if a remote file `/user/friend/work/report.tex` has contents:

```
My graph: \includegraphics{picture}
\input{explanation}
```

then you can input that file to a document with `\import{/user/friend/work/}{report}` so that both the explanation and picture will be taken from the `/user/friend/work/` directory.

The ⟨*full-path*⟩ argument for `\import` can be a full absolute path or a relative path starting from the main working directory for the document. The `\subimport` command facilitates nesting of file imports. It takes a relative ⟨*path-extension*⟩ based on the location of the current imported file. For example, if a file is imported (using either command) from directory `abc/` and that file contains the command `\subimport{lmn/}{xyz}` then file `abc/lmn/xyz.tex` is

input, and any `\input` commands in that file will read files from directory `abc/lmn/`.[1]

Depending on on how your TeX system is configured, if a file does not exist in the specified import directory, it will be looked for in previous import directories (when nesting `\subimport` files), then in any directory listed in the pre-existing `\input@path`, then in the current working directory, and finally in the `TEXINPUTS` path. Therefore, for `\import` and for other `\input` used within an imported file, a file found on the path of import(s) will be read in preference to others with the same name located elsewhere. So here is the real behavior of the previous example: Given the nested sequence:

    \import{abc/}{one}   (in main document);
    \subimport{lmn/uvw/}{two}   (in file abc/one.tex);
    \input{three}   (in file abc/lmn/uvw/two.tex);

LaTeX first looks for `abc/lmn/uvw/three` (or `abc/lmn/uvw/three.tex`);
if not found, it tries `abc/three` (or `abc/three.tex`);
if still not found, it looks in the `\input@path`, if there was one defined;
if `\input@path` was not defined, or if the file was not found on it, LaTeX then tries to open `three` in the main document directory;
finally, if still not found, it searches the `TEXINPUTS` search path.

Historically, 'star' versions of the commands were defined (to avoid searching the `TEXINPUTS` path) but now the "∗" is ignored.

A command "`\import@path@fix`" is provided to reformat the import path to fit the syntax of a particular operating system. It *could* be defined to convert unix-style path names to the local format. The default definition ensures paths end with a single "/" on most systems (unless the path is empty), but on VMS systems it removes "][" from within combined paths.

Note that the import package works by manipulating the internal 'commands' `\input@path` and `\Ginput@path`, so may behave badly if you redefine them manually, or via another package, within the document.

Presently, the paths are defined 'locally' (not globally) so input files must have balanced grouping.

---

[1] Note that the sub-import path is merely appended to the current import path. Syntactical mistakes from this method may be corrected by `\import@path@fix`.