The ifplatform package

Original code by Johannes Große
Package by Will Robertson
wspr 81 at gmail dot com

v0.3a[*] 2009/09/10

1 Main features and usage

This package provides the three following conditionals to test which operating system is being used to run TEX:

```
    \ifwindows
    \iflinux
    \ifmacosx
    \ifcygwin
```

If you only wish to detect `\ifwindows`, then it does not matter how you load this package. Note then that use of (Linux *or* Mac OS X *or* Cygwin) can then be detected with `\ifwindows\else`.

If you also wish to determine the difference between which Unix-variant you are using (i.e., also detect `\iflinux`, `\ifmacosx`, and `\ifcygwin`) then shell escape must be enabled.

[*] Thanks to Ken Brown, Joseph Wright, Zebb Prime, and others for testing this version.

This is achieved by using the `-shell-escape` command line option when executing LaTeX.

If shell escape is not enabled, `\iflinux`, `\ifmacosx`, and `\ifcygwin` will all return *false*. A warning will be printed in the console output to remind you in this case.

2 Auxiliary features

`\ifshellescape` is provided as a conditional to test whether shell escape is active or not. (Note: new versions of pdfTeX allow you to query shell escape with `\ifnum\pdfshellescape>0`.)

Also, the `\platformname` command is defined to expand to a macro that represents the operating system. Default definitions are (respectively):

`\windowsname`	→	'Windows'
`\notwindowsname`	→	'*NIX' (when shell escape is disabled)
`\linuxname`	→	'Linux'
`\macosxname`	→	'Mac OS X'
`\cygwinname`	→	'Cygwin'
`\unknownplatform`	→	*whatever is returned by* `uname`

E.g., if `\ifwindows` is *true* then `\platformname` expands to `\windowsname`, which expands to 'Windows'. Redefine the macros above to customise the output of `\platformname`.

This documentation was compiled on Mac OS X.

3 Other platforms

If greater granularity is required to differentiate between various UNIX-like operating systems, then `\unknownplatform` can be interrogated for the platform based on the output of uname. Table 1 lists possible outputs for a range of operating systems.

For example, to test whether the AIX operating system is being used, you could use the following code:

```
 \def\myplatform{aix6}
\ifx\myplatform\unknownplatform
  ... AIX is being used ...
\else
  ... or not ...
\fi
```

The ifthen and xifthen packages might be of interest to those who prefer more LaTeX-like methods of conditional testing.

Platform	uname string
FreeBSD	FreeBSD
OpenBSD	OpenBSD
Solaris	SunOS
HPUX	HP-UX
IRIX	IRIX64
AIX	aix6
Cray UNICOS	sn5176

Table 1: List of operating systems and their uname strings. Adapted from http://en.wikipedia.org/wiki/Uname.

4 Limitations

Some technical information in case things go wrong.

- ifplatform checks for Windows by the presence or absence of the file ‘`nul:`’. If you have a file in your search path in *nix called ‘`nul:.tex`’ (or without the `.tex`) then things may become confused.

- ifplatform checks for *nix by the presence or absence of the file ‘`/dev/null`’. If you have the file in Windows called `/dev/null.tex` (or without the extension) then things might similarly get mixed up.

- When both null files are detected (i.e., things aren’t right with one of the two tests above), ifplatform uses another test to try and sort itself out. For interest, the test is: ‘`echo # > \jobname.w18`’. Under Windows you should end up with a text file containing an octothorpe. On *nix, the `#` will be seen as a comment char and the test will be ignored and the file will not be written.

  This ‘last resort’ test will fail if shell escape is not enabled, or if the file `\jobname.w18` somehow already exists, or if the behaviour of `#` isn’t as reliable as I think.

- Note that if you’re running TeX binaries from Cygwin on Windows, then your platform will *not* be Windows. It will appear to be a *nix system, with platform name ‘Cygwin’.

- If you ever see the error

  ```
  I can’t tell if this is Windows or *nix;
  you appear to be neither.
  ```

  then I’d dearly like to know how it happened. It should never occur, as far as I know.

Keep these points in mind and you’ll never run into trouble. I hope you won’t run into trouble in any case.

5 Implementation

```
\ProvidesPackage{ifplatform}
  [2009/09/10 v0.3a  Testing for the operating system]
```

Packages required: (thanks Heiko)

```
\RequirePackage{pdftexcmds,catchfile}
```

Conditionals we provide:

```
\newif\ifshellescape
\newif\ifwindows
\newif\ifmacosx
\newif\iflinux
\newif\ifcygwin
```

\windowsname
\notwindowsname
\linuxname
\macosxname
\cygwinname
\unknownplatform

Names of operating systems:

```
\newcommand\windowsname{Windows}
\newcommand\notwindowsname{*NIX}
\newcommand\linuxname{Linux}
\newcommand\macosxname{Mac\,OS\,X}
\newcommand\cygwinname{Cygwin}
\newcommand\unknownplatform{[Unknown]}
```

For internal stuff later:

```
\edef\ip@file{\jobname.w18}
\newif\if@ip@nix@
```

\ifshellescape

Determine if shell escape is enabled:

```
\ifnum\pdf@shellescape=1\relax
  \shellescapetrue
\else
  \PackageWarningNoLine{ifplatform}{^^J \space\space\space
    shell escape is disabled,
    so I can only detect \@backslashchar ifwindows%
  }
\fi
```

File	Exists	Windows?	*nix?
`nul:`	true	Probably	Maybe
	false	Definitely not	Definitely
`/dev/null`	true	Maybe	Probably
	false	Definitely	Definitely not

Table 2: Possibilities for testing null files and their prospects for determining the platform.

An error message for when things go wrong:

```
\def\ip@cantdecide{%
  \PackageWarningNoLine{ifplatform}{^^J \space\space\space
    I can't tell if this is Windows or *nix;
    you appear to be both%
  }%
}
```

Test for the null files of Windows and *nix. In a normal situation, this is all we need to do:

```
\IfFileExists{nul:}{\@ip@nix@false}{\@ip@nix@true}
\IfFileExists{/dev/null}{\windowsfalse}{\windowstrue}
```

However, sometimes that's not good enough. If things go wrong above, we still don't know which platform. Can only proceed if shell escape is on; fallback heuristic:

- If the tmp file exists
  - Tell them to delete it and abort.
  - Otherwise:
- Write to it with `echo` that only works on Windows
- Then see again if it exists
  - If the tmp file exists: Windows (and delete the file)
  - Otherwise: *nix

Here's the code for the above 'last resort' test:

```
\edef\ip@windows@echo@test{echo \string# > "\ip@file"}
\def\ip@backupplan{%
  \IfFileExists{\ip@file}{%
    \PackageWarningNoLine{ifplatform}{^^J \space\space\space
      Please delete the file "\ip@file" and try again%
    }%
    \ip@cantdecide
  }{%
    \immediate\write18{\ip@windows@echo@test}%
    \IfFileExists{\ip@file}{%
      \immediate\write18{del "\ip@file"}%
      \windowstrue
    }{%
      \@ip@nix@true
    }%
  }%
}
```

Now we use some odd logic to deduce what's happening in the edge cases when things go wrong: (see table 2)

```
\ifwindows
  \if@ip@nix@
    \PackageWarningNoLine{ifplatform}{^^J \space\space\space
      I can't tell if this is Windows or *nix;
      you appear to be neither%
    }
  \fi
\else
  \if@ip@nix@\else
    \ifshellescape
      \ip@backupplan
    \else
      \ip@cantdecide
    \fi
  \fi
\fi
```

Needed below:

```
\def\ip@only@six#1#2#3#4#5#6#7\@nil{#1#2#3#4#5#6}
```

\iflinux
\ifmacosx
\ifcygwin

Now test for the others; directly test for Linux and Mac OS X; but what about Solaris or FreeBSD or . . . ? Define `\unknownplatform` as the output of `uname` rather than enumerate the possibilities.

```
\if@ip@nix@
\ifshellescape
  \ifwindows\else
    \immediate\write18{uname -s > "\ip@file"}
    \CatchFileDef\@tempa{\ip@file}{}
    \immediate\write18{rm -- "\ip@file"}
```

Kill a trailing space:

```
    \edef\@tempa{\expandafter\zap@space\@tempa\@empty}
    \def\@tempb{Linux}
    \ifx\@tempa\@tempb
      \linuxtrue
    \else
      \def\@tempb{Darwin}
      \ifx\@tempa\@tempb
        \macosxtrue
      \else
        \def\@tempb{CYGWIN}
        \edef\@tempc{\expandafter\ip@only@six\@tempa------\@nil}
        \ifx\@tempb\@tempc
          \cygwintrue
        \else
          \edef\unknownplatform{\@tempa}
        \fi
      \fi
    \fi
```

```
  \fi
\fi\fi
```

\platformname Defined in terms of macros so the output is user-customisable.

```
\edef\platformname{%
  \ifwindows
    \noexpand\windowsname
  \else
    \ifshellescape
      \iflinux
        \noexpand\linuxname
      \else
        \ifmacosx
          \noexpand\macosxname
        \else
          \ifcygwin
            \noexpand\cygwinname
          \else
            \noexpand\unknownplatform
          \fi
        \fi
      \fi
    \else
      \noexpand\notwindowsname
    \fi
  \fi
}
```

Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.