The `binhex.tex` package for expansible conversion into binary-based number systems

David Kastrup*

July 22, 2005

Contents

1 Usage

This is a file for expandably converting numbers into binary, octal and hexadecimal. All constructs TEX accepts as an argument to its `\number` primitive are valid. This holds for all numeric arguments of the macros presented in here.

You use this package by simply inputting it with

```
\input binhex
```

It will work equally well under LATEX and plain TEX. It does not even use plain TEX, but only TEX primitives. Simply setting the correct `\catcode` values for `{}#` and end of line will make it load and work under iniTEX.

`\binary` The following macros are defined: `\binary{`⟨*number*⟩`}` will convert ⟨*number*⟩ into its binary representation.

```
\binary{0}          →  0
\binary{\maxdimen}  →  11111111111111111111111111111
\binary{-"7EE6}     →  -111111011100110
```

`\nbinary` `\nbinary{`⟨*size*⟩`}{`⟨*number*⟩`}` will convert ⟨*number*⟩ into a binary representation of at least ⟨*size*⟩ digits length, filling up with leading zeros where necessary. The `-` sign of negative numbers is not counted. If both ⟨*size*⟩ and ⟨*number*⟩ are zero, an empty string is generated. This should please some computer scientists in some situations.

```
\nbinary{3}{3}   →  011
\nbinary{3}{-2}  →  -010
\nbinary{3}{-12} →  -1100
```

`\hex` `\hex{`⟨*number*⟩`}` converts ⟨*number*⟩ into its hexadecimal representation, using

*David.Kastrup@neuroinformatik.ruhr-uni-bochum.de

uppercase letters.

```
\hex{34}    →   22
\hex{-4711}   →   -1267
```

`\nhex` `\nhex{`⟨*size*⟩`}{`⟨*number*⟩`}` will convert ⟨*number*⟩ into a hexadecimal representation of at least ⟨*size*⟩ digits length, filling up with leading zeros where necessary. The `-` sign of negative numbers is not counted. If both ⟨*size*⟩ and ⟨*number*⟩ are zero, an empty string is generated. This should please some computer scientists in some situations.

```
\nhex{3}{3}    →   003
\nhex{3}{-\maxdimen}   →   -3FFFFFFF
```

`\oct` `\oct{`⟨*number*⟩`}` converts ⟨*number*⟩ into its octal representation.

```
\oct{34}    →   42
\oct{-4711}   →   -11147
```

`\noct` `\noct{`⟨*size*⟩`}{`⟨*number*⟩`}` will do the right thing.

```
\noct{3}{13}    →   015
\noct{3}{-\maxdimen}   →   -7777777777
```

`\tetra` `\tetra{`⟨*number*⟩`}` is for people counting with arms and legs instead of fingers, or for quadrupeds.

```
\tetra{34}    →   202
\tetra{-4711}   →   -1021213
```

`\ntetra` `\ntetra{`⟨*size*⟩`}{`⟨*number*⟩`}` is for those of the same count which have minimum requirements.

```
\ntetra{3}{3}    →   003
\ntetra{3}{-\maxdimen}   →   -333333333333333
```

`\nbinbased` `\nbinbased{`⟨*logbase*⟩`}{`⟨*size*⟩`}{`⟨*number*⟩`}` will convert ⟨*number*⟩ into number base $2^{\langle logbase \rangle}$ and generate at least ⟨*size*⟩ digits. Only supported values of ⟨*logbase*⟩ are 1, 2, 3, 4. This is called by all other macros except of the faster binary conversion macros.

```
\nbinbased{3}{3}{13}    →   015
\nbinbased{3}{3}{-\maxdimen}   →   -7777777777

\nbinbased{2}{4}{13}    →   0031
\nbinbased{2}{4}{-\maxdimen}   →   -333333333333333
```

That's it, have fun!