

NAME

bits – convert bit masks from/to various formats

SYNOPSIS

bits [**-h**] [**-V**] [**-w** <NUM>] [<MODE>] [<MASK_OR_LIST>...]

DESCRIPTION

The **bits** utility converts bit masks into various formats. It supports combining multiple masks together using bitwise operations.

POSITIONAL ARGUMENTS

<MASK_OR_LIST>

A set of bits specified as a hexadecimal mask value (e.g. *0xec2*) or as a comma-separated list of bit IDs.

If no argument is specified, the sets of bits will be read from standard input; one group per line.

Consecutive ids can be compressed as ranges (e.g. *5,6,7,8,9,10* → *5-10*).

Optionally, if an argument starts with a comma, it will be parsed as a single hexadecimal mask split in 32bit groups (e.g. *,00014000,00000000,00020000* → *17,78,80*).

By default all groups will be OR'ed together. If a group has one of the following prefixes, it will be combined with the resulting mask using a different binary operation:

&<MASK_OR_LIST>

The group will be combined with a binary AND operation. I.e. all bits that are set to 1 in the group AND the combined groups so far will be preserved to 1. All other bits will be reset to 0.

^<MASK_OR_LIST>

The group will be combined with a binary XOR operation. I.e. all bits that are set to 1 in the group AND to 0 the combined groups so far (or the other way around) will be set to 1. Bits that are both to 1 or both to 0 will be reset to 0.

~<MASK_OR_LIST>

All bits set to 1 in the group will be cleared (reset to 0) in the combined groups so far.

OPTIONS

-w <NUM>, **--width** <NUM>

Maximum number of bits in the masks handled by **bits** (default *8192*). Any bit larger than this number will be truncated.

-h, **--help**

Display help text and exit.

-V, **--version**

Display version and exit.

CONVERSION MODE

One of the following conversion modes can be specified. If not specified, it defaults to **-m**, **--mask**.

-m, **--mask**

Print the combined args as a hexadecimal mask value (default).

-g, **--grouped-mask**

Print the combined args as a hexadecimal mask value in 32bit comma separated groups.

-b, --binary

Print the combined args as a binary mask value.

-l, --list

Print the combined args as a list of bit IDs. Consecutive IDs are compressed as ranges.

EXAMPLES

```
~$ bits --mask 4,5-8 16,30
0x400101f0
```

```
~$ bits --list 0xeec2
1,6,7,9-11,13-15
```

```
~$ bits --binary 4,5-8 16,30
0b100_0000_0000_0001_0000_0001_1111_0000
```

```
~$ bits --list ,00300000,03000000,30000003
0,1,28,29,56,57,84,85
```

```
~$ bits --list 1,2,3,4 ~3-10
1,2
```

```
~$ bits --list 1,2,3,4 ^3-10
1,2,5-10
```

```
~$ bits --grouped-mask 2,22,74,79
8400,00000000,00400004
```

```
~$ bits --width 64 --list 2,22,74,79
2,22
```

AUTHORS

Robin Jarry.

REPORTING BUGS

For bug reports, use the issue tracker <<https://github.com/util-linux/util-linux/issues>>.

AVAILABILITY

The **bits** command is part of the util-linux package which can be downloaded from Linux Kernel Archive <<https://www.kernel.org/pub/linux/utils/util-linux/>>.