NAME

bit_alloc, bit_clear, bit_decl, bit_ffs, bit_nclear, bit_nset, bit_set, bitstr_size, bit_test — bit-string manipulation macros

LIBRARY

Utility functions from BSD systems (libbsd, −lbsd)

SYNOPSIS

#include <bitstring.h>
(See *libbsd*(7) for include usage.)

bitstr_t ∗
bit_alloc(*int nbits*);

void
bit_decl(*bitstr_t ∗name*, *int nbits*);

void
bit_clear(*bitstr_t ∗name*, *int bit*);

void
bit_ffc(*bitstr_t ∗name*, *int nbits*, *int ∗value*);

void
bit_ffs(*bitstr_t ∗name*, *int nbits*, *int ∗value*);

void
bit_nclear(*bitstr_t ∗name*, *int start*, *int stop*);

void
bit_nset(*bitstr_t ∗name*, *int start*, *int stop*);

void
bit_set(*bitstr_t ∗name*, *int bit*);

int
bitstr_size(*int nbits*);

int
bit_test(*bitstr_t ∗name*, *int bit*);

DESCRIPTION

These macros operate on strings of bits.

The macro **bit_alloc**() returns a pointer of type "*bitstr_t* ∗" to sufficient space to store *nbits* bits, or NULL if no space is available.

The macro **bit_decl**() allocates sufficient space to store *nbits* bits on the stack.

The macro **bitstr_size**() returns the number of elements of type *bitstr_t* necessary to store *nbits* bits. This is useful for copying bit strings.

The macros **bit_clear**() and **bit_set**() clear or set the zero-based numbered bit *bit*, in the bit string *name*.

The **bit_nset**() and **bit_nclear**() macros set or clear the zero-based numbered bits from *start* through *stop* in the bit string *name*.

The **bit_test**() macro evaluates to non-zero if the zero-based numbered bit *bit* of bit string *name* is set, and zero otherwise.

The **bit_ffs**() macro stores in the location referenced by *value* the zero-based number of the first bit set in the array of *nbits* bits referenced by *name*. If no bits are set, the location referenced by*value* is set to −1.

The macro **bit_ffc**() stores in the location referenced by *value* the zero-based number of the first bit not set in the array of *nbits* bits referenced by *name*. If all bits are set, the location referenced by *value* is set to −1.

The arguments to these macros are evaluated only once and may safely have side effects.

EXAMPLES

```
#include <limits.h>
#include <bsd/bitstring.h>

...
#define LPR_BUSY_BIT          0
#define LPR_FORMAT_BIT        1
#define LPR_DOWNLOAD_BIT      2
...
#define LPR_AVAILABLE_BIT     9
#define LPR_MAX_BITS          10

make_lpr_available()
{
        bitstr_t bit_decl(bitlist, LPR_MAX_BITS);
        ...
        bit_nclear(bitlist, 0, LPR_MAX_BITS - 1);
        ...
        if (!bit_test(bitlist, LPR_BUSY_BIT)) {
                bit_clear(bitlist, LPR_FORMAT_BIT);
                bit_clear(bitlist, LPR_DOWNLOAD_BIT);
                bit_set(bitlist, LPR_AVAILABLE_BIT);
        }
}
```

SEE ALSO

malloc(3)

HISTORY

The **bitstring** functions first appeared in 4.4BSD.