NAME

stab - symbol table types

SYNOPSIS

#include <stab.h>

DESCRIPTION

The file <*stab.h*> defines some of the symbol table *n_type* field values for a.out files. These are the types for permanent symbols (i.e., not local labels, etc.) used by the old debugger *sdb* and the Berkeley Pascal compiler pc(1). Symbol table entries can be produced by the *.stabs* assembler directive. This allows one to specify a double-quote delimited name, a symbol type, one char and one short of information about the symbol, and an unsigned long (usually an address). To avoid having to produce an explicit label for the address field, the *.stabd* directive can be used to implicitly address the current location. If no name is needed, symbol table entries can be generated using the *.stabn* directive. The loader promises to preserve the order of symbol table entries produced by *.stab* directives. As described in a.out(5), an element of the symbol table consists of the following structure:

```
/*
* Format of a symbol table entry.
*/

struct nlist {
        union {
                const char *n_name;             /* for use when in-core */
                long        n_strx;             /* index into file string table */
        } n_un;
        unsigned char       n_type;             /* type flag */
        char                n_other;   /* unused */
        short               n_desc;             /* see struct desc, below */
        unsigned  n_value;  /* address or offset or line */
};
```

The low bits of the *n_type* field are used to place a symbol into at most one segment, according to the following masks, defined in <*a.out.h*>. A symbol can be in none of these segments by having none of these segment bits set.

```
/*
* Simple values for n_type.
*/
```

```
#define    N_UNDF 0x0      /* undefined */
#define    N_ABS  0x2      /* absolute */
#define    N_TEXT 0x4      /* text */
#define    N_DATA 0x6      /* data */
#define    N_BSS  0x8      /* bss */

#define    N_EXT  01       /* external bit, or'ed in */
```

The *n_value* field of a symbol is relocated by the linker, ld(1) as an address within the appropriate segment. *N_value* fields of symbols not in any segment are unchanged by the linker. In addition, the linker will discard certain symbols, according to rules of its own, unless the *n_type* field has one of the following bits set:

```
/*
* Other permanent symbol table entries have some of the N_STAB bits set.
* These are given in <stab.h>
*/

#define    N_STAB 0xe0     /* if any of these bits set, don't discard */
```

This allows up to 112 (7 * 16) symbol types, split between the various segments. Some of these have already been claimed. The old symbolic debugger, *sdb*, uses the following n_type values:

```
#define    N_GSYM      0x20    /* global symbol: name,,0,type,0 */
#define    N_FNAME     0x22    /* procedure name (f77 kludge): name,,0 */
#define    N_FUN  0x24 /* procedure: name,,0,linenumber,address */
#define    N_STSYM     0x26    /* static symbol: name,,0,type,address */
#define    N_LCSYM     0x28    /* .lcomm symbol: name,,0,type,address */
#define    N_RSYM      0x40    /* register sym: name,,0,type,register */
#define    N_SLINE     0x44    /* src line: 0,,0,linenumber,address */
#define    N_SSYM 0x60 /* structure elt: name,,0,type,struct_offset */
#define    N_SO   0x64 /* source file name: name,,0,0,address */
#define    N_LSYM      0x80    /* local sym: name,,0,type,offset */
#define    N_SOL  0x84 /* #included file name: name,,0,0,address */
#define    N_PSYM 0xa0 /* parameter: name,,0,type,offset */
#define    N_ENTRY     0xa4    /* alternate entry: name,linenumber,address */
#define    N_LBRAC     0xc0    /* left bracket: 0,,0,nesting level,address */
#define    N_RBRAC     0xe0    /* right bracket: 0,,0,nesting level,address */
#define    N_BCOMM     0xe2    /* begin common: name,, */
#define    N_ECOMM     0xe4    /* end common: name,, */
```

```
#define N_ECOML 0xe8 /* end common (local name): ,,address */
#define N_LENG 0xfe /* second stab entry with length information */
```

where the comments give *sdb* conventional use for *.stab s* and the *n_name*, *n_other*, *n_desc*, and *n_value* fields of the given *n_type*. *Sdb* uses the *n_desc* field to hold a type specifier in the form used by the Portable C Compiler, cc(1); see the header file *pcc.h* for details on the format of these type values.

The Berkeley Pascal compiler, pc(1), uses the following *n_type* value:

```
#define N_PC 0x30 /* global pascal symbol: name,,0,subtype,line */
```

and uses the following subtypes to do type checking across separately compiled files:

1	source file name
2	included file name
3	global label
4	global constant
5	global type
6	global variable
7	global function
8	global procedure
9	external function
10	external procedure
11	library variable
12	library routine

SEE ALSO

as(1), ld(1), a.out(5)

HISTORY

The **stab** file appeared in 4.0BSD.

BUGS

More basic types are needed.