Left column:

```
 1: /*     Copyright (c) 1990, 1991, 1992, 1993 UNIX System Laboratories, Inc.    */
 2: /*     Copyright (c) 1988, 1990 AT&T    */
 3: /*       All Rights Reserved        */
 4:
 5: /*     THIS IS UNPUBLISHED PROPRIETARY SOURCE CODE OF         */
 6: /*     UNIX System Laboratories, Inc.                         */
 7: /*     The copyright notice above does not evidence any       */
 8: /*     actual or intended publication of such source code.    */
 9:
10: #ident    "@(#)libelf:common/strptr.c     1.6"
11:
12:
13: #ifdef __STDC__
14:     #pragma weak    elf_strptr = _elf_strptr
15: #endif
16:
17:
18: #include "syn.h"
19: #include "libelf.h"
20: #include "decl.h"
21: #include "error.h"
22:
23:
24: char *
25: elf_strptr(elf, ndx, off)
26:     Elf            *elf;
27:     size_t            ndx;
28:     size_t            off;
29: {
30:     register Elf_Scn    *s;
31:     register Elf_Data    *d;
32:
33:     if (elf == 0)
34:         return 0;
35:     if ((s = elf_getscn(elf, ndx)) == 0
36:     || s->s_shdr == 0
37:     || s->s_shdr->sh_type != SHT_STRTAB)
38:     {
39:         _elf_err = EREQ_STRSCN;
40:         return 0;
41:     }
42:
43:     /*    If the layout bit is set, use the offsets and
44:      *    sizes in the data buffers.  Otherwise, take
45:      *    data buffers in order.
46:      */
47:
48:     d = 0;
49:     if (elf->ed_uflags & ELF_F_LAYOUT)
50:     {
51:         while ((d = elf_getdata(s, d)) != 0)
52:         {
53:             if (d->d_buf == 0)
54:                 continue;
55:             if (off >= d->d_off
56:             && off < d->d_off + d->d_size)
57:                 return (char *)d->d_buf + off - d->d_off;
58:         }
59:     }
60:     else
61:     {
62:         size_t    sz = 0, j;
63:
64:         while ((d = elf_getdata(s, d)) != 0)
65:         {
66:             if ((j = d->d_align) > 1 && sz % j != 0)
67:             {
68:                 j -= sz % j;
69:                 sz += j;
70:                 if (off < j)
71:                     break;
72:                 off -= j;
73:             }
74:             if (d->d_buf != 0)
75:             {
76:                 if (off < d->d_size)
77:                     return (char *)d->d_buf + off;
78:             }
```

Right column:

```
 1: /*
 2: strptr.c - implementation of the elf_strptr(3) function.
 3: Copyright (C) 1995 Michael Riepe <riepe@ifwsn4.ifw.uni-hannover.de>
 4:
 5: This library is free software; you can redistribute it and/or
 6: modify it under the terms of the GNU Library General Public
 7: License as published by the Free Software Foundation; either
 8: version 2 of the License, or (at your option) any later version.
 9:
10: This library is distributed in the hope that it will be useful,
11: but WITHOUT ANY WARRANTY; without even the implied warranty of
12: MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
13: Library General Public License for more details.
14:
15: You should have received a copy of the GNU Library General Public
16: License along with this library; if not, write to the Free Software
17: Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.
18: */
19:
20: #include <private.h>
21:
22: char*
23: elf_strptr(Elf *elf, size_t section, size_t offset) {
24:     Elf_Scn *scn;
25:     Elf_Data *sd;
26:
27:     if (!elf) {
28:     return NULL;
29:     }
30:     elf_assert(elf->e_magic == ELF_MAGIC);
31:     if (!(scn = elf_getscn(elf, section))) {
32:     return NULL;
33:     }
34:     if (scn->s_type != SHT_STRTAB) {
35:     seterr(ERROR_NOSTRTAB);
36:     return NULL;
37:     }
38:     if (offset >= 0 && offset < scn->s_size) {
39:     sd = NULL;
40:     while ((sd = elf_getdata(scn, sd))) {
41:         if (sd->d_buf && offset >= (size_t)sd->d_off
42:             && offset < (size_t)sd->d_off + sd->d_size) {
43:         return (char*)sd->d_buf + (offset - sd->d_off);
44:         }
45:     }
46:     }
47:     seterr(ERROR_BADSTROFF);
48:     return NULL;
49: }
```

(For readability, not all matches are indicated.)

```
79:                sz += d->d_size;
80:                if (off < d->d_size)
81:                    break;
82:                off -= d->d_size;
83:            }
84:        }
85:    _elf_err = EREQ_STROFF;
86:    return 0;
87: }
```

(For readability, not all matches are indicated.)