```
 1: // $Id: strlist.c,v 1.3 2012-02-15 20:45:26-08 - - $
 2:
 3: // Reads in a sequence of lines and then prints them out in debug
 4: // format.  strdup(3) copies these lines onto the heap.  Read the
 5: // comments in the file 'numlist.c' first.
 6:
 7: #include <assert.h>
 8: #include <libgen.h>
 9: #include <stdio.h>
10: #include <stdlib.h>
11: #include <string.h>
12:
13: //
14: // Declaration for linked list of nodes.
15: //
16: typedef struct node *node_ref;
17: struct node {
18:    char *string;
19:    node_ref link;
20: };
21:
22: // LINTED (argument unused in function)
23: int main (int argc, char **argv) {
24:    char *progname = basename (argv[0]);
25:    node_ref head = NULL;
26:    char buffer[256];
27:    int linenr;
28:    for (linenr = 1; ; ++linenr) {
29:
30:       // Read a line of input and check to see if it ends with
31:       // a newline character.  Print a message if not.
32:
33:       char *gotline = fgets (buffer, sizeof buffer, stdin);
34:       if (gotline == NULL) break;
35:
36:       char *nlpos = strchr (buffer, '\n');
37:       if (nlpos != NULL) {
38:          *nlpos = '\0';
39:       }else {
40:          fprintf (stderr, "%s: %d: unterminated line: %s\n",
41:                   progname, linenr, buffer);
42:       };
43:
44:       // Allocate a node and initialize it to point a a heap copy
45:       // of the input line.  Note that strdup(3) contains a call
46:       // to malloc(3), so we need the NULL check there as well.
47:
48:       node_ref new = malloc (sizeof (struct node));
49:       assert (new != NULL);
50:       new->string = strdup (buffer);
51:       assert (new->string != NULL);
52:       new->link = head;
53:       head = new;
54:    };
```

```
55:
56:    // Print the results in debug mode.
57:
58:    printf ("%s: head= %p\n", argv[0], (void*) head);
59:    while (head != NULL) {
60:       node_ref old = head;
61:       head = head->link;
62:       printf ("%s: %p-> node {\n"
63:               "    string= %p->\"%s\",\n"
64:               "    link= %p}\n",
65:               progname, (void*) old, (void*) old->string,
66:               old->string, (void*) old->link);
67:    };
68:
69:    return EXIT_SUCCESS;
70: }
71:
72: /*
73: //TEST// (echo "this is line 1" \
74: //TEST// ;echo "" \
75: //TEST// ;echo "the previous line has length 0." \
76: //TEST// ;echo "fit the buffer." \
77: //TEST// ;echo "Last Line." \
78: //TEST// ) | valgrind --leak-check=full --log-file=strlist.lisval \
79: //TEST// ./strlist >strlist.lisout 2>&1
80: //TEST// mkpspdf strlist.ps strlist.c* strlist.lis*
81: */
82:
```

```
 1: @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: starting strlist.c
 2: strlist.c: $Id: strlist.c,v 1.3 2012-02-15 20:45:26-08 - - $
 3: gcc -g -O0 -Wall -Wextra -std=gnu99 strlist.c -o strlist -lm
 4: strlist.c:23: warning: unused parameter 'argc'
 5: lint -Xa -fd -m -u -x -errchk=%all strlist.c
 6:
 7: function returns value which is always ignored
 8:     fprintf             printf
 9: rm -f strlist.o
10: @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: finished strlist.c
```

```
 1: ./strlist: head= 0x4c302d0
 2: strlist: 0x4c302d0-> node {
 3:     string= 0x4c30320->"Last Line.",
 4:     link= 0x4c30230}
 5: strlist: 0x4c30230-> node {
 6:     string= 0x4c30280->"fit the buffer.",
 7:     link= 0x4c30180}
 8: strlist: 0x4c30180-> node {
 9:     string= 0x4c301d0->"the previous line has length 0.",
10:     link= 0x4c300e0}
11: strlist: 0x4c300e0-> node {
12:     string= 0x4c30130->"",
13:     link= 0x4c30040}
14: strlist: 0x4c30040-> node {
15:     string= 0x4c30090->"this is line 1",
16:     link= (nil)}
```

```
   1: ==27729== Memcheck, a memory error detector
   2: ==27729== Copyright (C) 2002-2009, and GNU GPL'd, by Julian Seward et al.
   3: ==27729== Using Valgrind-3.5.0 and LibVEX; rerun with -h for copyright info
   4: ==27729== Command: ./strlist
   5: ==27729== Parent PID: 27727
   6: ==27729==
   7: ==27729==
   8: ==27729== HEAP SUMMARY:
   9: ==27729==     in use at exit: 155 bytes in 10 blocks
  10: ==27729==   total heap usage: 10 allocs, 0 frees, 155 bytes allocated
  11: ==27729==
  12: ==27729== 155 (16 direct, 139 indirect) bytes in 1 blocks are definitely lost in
loss record 3 of 3
  13: ==27729==    at 0x4A05E1C: malloc (vg_replace_malloc.c:195)
  14: ==27729==    by 0x4007FE: main (strlist.c:48)
  15: ==27729==
  16: ==27729== LEAK SUMMARY:
  17: ==27729==    definitely lost: 16 bytes in 1 blocks
  18: ==27729==    indirectly lost: 139 bytes in 9 blocks
  19: ==27729==      possibly lost: 0 bytes in 0 blocks
  20: ==27729==    still reachable: 0 bytes in 0 blocks
  21: ==27729==         suppressed: 0 bytes in 0 blocks
  22: ==27729==
  23: ==27729== For counts of detected and suppressed errors, rerun with: -v
  24: ==27729== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 4 from 4)
```