```
 1: // $Id: strlist.c,v 1.11 2014-02-07 17:13:35-08 - - $
 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;
17: struct node {
18:    char *string;
19:    node *link;
20: };
21:
22: int main (int argc, char **argv) {
23:    (void) argc; // warning: unused parameter 'argc'
24:    char *progname = basename (argv[0]);
25:    node *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 *tmp = malloc (sizeof (struct node));
49:       assert (tmp != NULL);
50:       tmp->string = strdup (buffer);
51:       assert (tmp->string != NULL);
52:       tmp->link = head;
53:       head = tmp;
54:    };
```

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

```
    1: @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: starting strlist.c
    2: strlist.c:
    3:      $Id: strlist.c,v 1.11 2014-02-07 17:13:35-08 - - $
    4: gcc -g -O0 -Wall -Wextra -rdynamic -std=gnu11 strlist.c -o strlist -lglu
t -lGLU -lGL -lX11 -lrt -lm
    5: rm -f strlist.o
    6: @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: finished strlist.c
```

```
 1: ./strlist: head= 0x5cb7320
 2: strlist: 0x5cb7320-> node {
 3:     string= 0x5cb7370->"Last Line.",
 4:     link= 0x5cb7280}
 5: strlist: 0x5cb7280-> node {
 6:     string= 0x5cb72d0->"fit the buffer.",
 7:     link= 0x5cb71d0}
 8: strlist: 0x5cb71d0-> node {
 9:     string= 0x5cb7220->"the previous line has length 0.",
10:     link= 0x5cb7130}
11: strlist: 0x5cb7130-> node {
12:     string= 0x5cb7180->"",
13:     link= 0x5cb7090}
14: strlist: 0x5cb7090-> node {
15:     string= 0x5cb70e0->"this is line 1",
16:     link= (nil)}
```

```
    1: ==22480== Memcheck, a memory error detector
    2: ==22480== Copyright (C) 2002-2013, and GNU GPL'd, by Julian Seward et al
.
    3: ==22480== Using Valgrind-3.9.0 and LibVEX; rerun with -h for copyright i
nfo
    4: ==22480== Command: ./strlist
    5: ==22480== Parent PID: 22478
    6: ==22480==
    7: ==22480==
    8: ==22480== HEAP SUMMARY:
    9: ==22480==     in use at exit: 155 bytes in 10 blocks
   10: ==22480==   total heap usage: 11 allocs, 1 frees, 164 bytes allocated
   11: ==22480==
   12: ==22480== 155 (16 direct, 139 indirect) bytes in 1 blocks are definitely
 lost in loss record 3 of 3
   13: ==22480==    at 0x4C27AAA: malloc (in /opt/rh/devtoolset-2/root/usr/lib6
4/valgrind/vgpreload_memcheck-amd64-linux.so)
   14: ==22480==    by 0x400AF4: main (strlist.c:48)
   15: ==22480==
   16: ==22480== LEAK SUMMARY:
   17: ==22480==    definitely lost: 16 bytes in 1 blocks
   18: ==22480==    indirectly lost: 139 bytes in 9 blocks
   19: ==22480==      possibly lost: 0 bytes in 0 blocks
   20: ==22480==    still reachable: 0 bytes in 0 blocks
   21: ==22480==         suppressed: 0 bytes in 0 blocks
   22: ==22480==
   23: ==22480== For counts of detected and suppressed errors, rerun with: -v
   24: ==22480== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 6 from 6)
```