```
// $Id: strlist.c,v 1.12 2016-01-28 16:02:07-08 - - $

// Reads in a sequence of lines and then prints them out in debug
// format.  strdup(3) copies these lines onto the heap.  Read the
// comments in the file "numlist.c" first.

#include <assert.h>
#include <libgen.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

//
// Declaration for linked list of nodes.
//
typedef struct node node;
struct node {
   char *string;
   node *link;
};

int main (int argc, char **argv) {
   (void) argc; // warning: unused parameter ’argc’
   char *progname = basename (argv[0]);
   node *head = NULL;
   char buffer[256];
   int linenr;
   for (linenr = 1; ; ++linenr) {

      // Read a line of input and check to see if it ends with
      // a newline character.  Print a message if not.

      char *gotline = fgets (buffer, sizeof buffer, stdin);
      if (gotline == NULL) break;

      char *nlpos = strchr (buffer, ’\n’);
      if (nlpos != NULL) {
         *nlpos = ’\0’;
      }else {
         fprintf (stderr, "%s: %d: unterminated line: %s\n",
                  progname, linenr, buffer);
      }

      // Allocate a node and initialize it to point a a heap copy
      // of the input line.  Note that strdup(3) contains a call
      // to malloc(3), so we need the NULL check there as well.

      node *tmp = malloc (sizeof (struct node));
      assert (tmp != NULL);
      tmp->string = strdup (buffer);
      assert (tmp->string != NULL);
      tmp->link = head;
      head = tmp;
   }

   // Print the results in debug mode.

   printf ("%s: head= %p\n", argv[0], head);
   while (head != NULL) {
      node *old = head;
      head = head->link;
      printf ("%s: %p-> node {\n"
              "    string= %p->\"%s\",\n"
              "    link= %p}\n",
              progname, old, old->string, old->string, old->link);
   }

   return EXIT_SUCCESS;
}

/*
//TEST// (echo "this is line 1" \
//TEST// ;echo "" \
//TEST// ;echo "the previous line has length 0." \
//TEST// ;echo "fit the buffer." \
//TEST// ;echo "Last Line." \
//TEST// ) | valgrind --leak-check=full --log-file=strlist.lisval \
//TEST// ./strlist >strlist.lisout 2>&1
//TEST// mkpspdf strlist.ps strlist.c* strlist.lis*
*/

```

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: starting strlist.c
strlist.c:
     $Id: strlist.c,v 1.12 2016-01-28 16:02:07-08 - - $
gcc -g -O0 -Wall -Wextra -rdynamic -std=gnu11 strlist.c
        -o strlist -lglut -lGLU -lGL -lX11 -lrt -lm
rm -f strlist.o
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: finished strlist.c
```

```
./strlist: head= 0x9c9a320
strlist: 0x9c9a320-> node {
    string= 0x9c9a370->"Last Line.",
    link= 0x9c9a280}
strlist: 0x9c9a280-> node {
    string= 0x9c9a2d0->"fit the buffer.",
    link= 0x9c9a1d0}
strlist: 0x9c9a1d0-> node {
    string= 0x9c9a220->"the previous line has length 0.",
    link= 0x9c9a130}
strlist: 0x9c9a130-> node {
    string= 0x9c9a180->"",
    link= 0x9c9a090}
strlist: 0x9c9a090-> node {
    string= 0x9c9a0e0->"this is line 1",
    link= (nil)}
```

```
==27055== Memcheck, a memory error detector
==27055== Copyright (C) 2002-2013, and GNU GPL'd, by Julian Seward et al.
==27055== Using Valgrind-3.10.1 and LibVEX; rerun with -h for copyright info
==27055== Command: ./strlist
==27055== Parent PID: 27053
==27055== 
==27055== 
==27055== HEAP SUMMARY:
==27055==     in use at exit: 155 bytes in 10 blocks
==27055==   total heap usage: 11 allocs, 1 frees, 171 bytes allocated
==27055== 
==27055== 155 (16 direct, 139 indirect) bytes in 1 blocks are definitely lost in loss record 3 of 3
==27055==    at 0x4C29BBD: malloc (in /opt/rh/devtoolset-3/root/usr/lib64/valgrind/vgpreload_memcheck-amd64-linux.so)
==27055==    by 0x400B4A: main (strlist.c:48)
==27055== 
==27055== LEAK SUMMARY:
==27055==    definitely lost: 16 bytes in 1 blocks
==27055==    indirectly lost: 139 bytes in 9 blocks
==27055==      possibly lost: 0 bytes in 0 blocks
==27055==    still reachable: 0 bytes in 0 blocks
==27055==         suppressed: 0 bytes in 0 blocks
==27055== 
==27055== For counts of detected and suppressed errors, rerun with: -v
==27055== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 1 from 1)
```