

```
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 *progrname = 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:                     progrname, 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:
```

[illegible]

```
1: ./strlist: head= 0x4c282d0
2: strlist: 0x4c282d0-> node {
3:     string= 0x4c28320->"Last Line.",
4:     link= 0x4c28230}
5: strlist: 0x4c28230-> node {
6:     string= 0x4c28280->"fit the buffer.",
7:     link= 0x4c28180}
8: strlist: 0x4c28180-> node {
9:     string= 0x4c281d0->"the previous line has length 0.",
10:    link= 0x4c280e0}
11: strlist: 0x4c280e0-> node {
12:    string= 0x4c28130->"",
13:    link= 0x4c28040}
14: strlist: 0x4c28040-> node {
15:    string= 0x4c28090->"this is line 1",
16:    link= (nil)}
```

```
1: ==17346== Memcheck, a memory error detector
2: ==17346== Copyright (C) 2002-2012, and GNU GPL'd, by Julian Seward et al
.
3: ==17346== Using Valgrind-3.8.1 and LibVEX; rerun with -h for copyright i
nfo
4: ==17346== Command: ./strlist
5: ==17346== Parent PID: 17344
6: ==17346==
7: ==17346==
8: ==17346== HEAP SUMMARY:
9: ==17346==      in use at exit: 155 bytes in 10 blocks
10: ==17346==    total heap usage: 10 allocs, 0 frees, 155 bytes allocated
11: ==17346==
12: ==17346== 155 (16 direct, 139 indirect) bytes in 1 blocks are definitely
lost in loss record 3 of 3
13: ==17346==    at 0x4A06A2E: malloc (in /opt/rh/devtoolset-2/root/usr/lib6
4/valgrind/vgpreload_memcheck-amd64-linux.so)
14: ==17346==    by 0x400884: main (strlist.c:48)
15: ==17346==
16: ==17346== LEAK SUMMARY:
17: ==17346==    definitely lost: 16 bytes in 1 blocks
18: ==17346==    indirectly lost: 139 bytes in 9 blocks
19: ==17346==    possibly lost: 0 bytes in 0 blocks
20: ==17346==    still reachable: 0 bytes in 0 blocks
21: ==17346==    suppressed: 0 bytes in 0 blocks
22: ==17346==
23: ==17346== For counts of detected and suppressed errors, rerun with: -v
24: ==17346== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 6 from 6)
```