```
 1: /* $Id: queue.h,v 1.2 2012-02-14 20:43:22-08 - - $ */
 2:
 3: #ifndef __QUEUE_H__
 4: #define __QUEUE_H__
 5:
 6: #include <stdbool.h>
 7:
 8: typedef struct queue *queue_ref;
 9: typedef char *queue_item_t;
10:
11: queue_ref new_queue (void);
12:
13: void free_queue (queue_ref);
14:
15: void insert_queue (queue_ref, queue_item_t);
16:
17: queue_item_t remove_queue (queue_ref);
18:
19: bool isempty_queue (queue_ref);
20:
21: bool is_queue (queue_ref);
22:
23: #endif
24:
```

```
 1: /* $Id: main.c,v 1.6 2012-02-22 19:23:35-08 - - $ */
 2:
 3: #include <assert.h>
 4: #include <errno.h>
 5: #include <libgen.h>
 6: #include <stdio.h>
 7: #include <stdlib.h>
 8: #include <string.h>
 9:
10: #include "queue.h"
11:
12: char *execname = NULL;
13: int exit_status = EXIT_SUCCESS;
14:
15: void putinqueue (queue_ref queue, FILE *input, char *filename) {
16:     char buffer[1024];
17:     for (int linenr = 1; ; ++linenr) {
18:         char *linepos = fgets (buffer, sizeof buffer, input);
19:         if (linepos == NULL) break;
20:         linepos = strchr (buffer, '\n');
21:         if (linepos == NULL) {
22:             fflush (NULL);
23:             fprintf (stderr, "%s: %s[%d]: unterminated line\n",
24:                      execname, filename, linenr);
25:             fflush (NULL);
26:             exit_status = EXIT_FAILURE;
27:         }else {
28:             *linepos = '\0';
29:         }
30:         linepos = strdup (buffer);
31:         assert (linepos != NULL);
32:         insert_queue (queue, linepos);
33:     }
34: }
35:
36: void putfileinqueue (queue_ref queue, char *filename) {
37:     FILE *input = fopen (filename, "r");
38:     if (input == NULL) {
39:         fflush (NULL);
40:         fprintf (stderr, "%s: %s: %s\n",
41:                  execname, filename, strerror (errno));
42:         fflush (NULL);
43:         exit_status = EXIT_FAILURE;
44:     }else {
45:         putinqueue (queue, input, filename);
46:         fclose (input);
47:     }
48: }
49:
```

```
50:
51: int main (int argc, char **argv) {
52:     execname = basename (argv[0]);
53:     queue_ref queue = new_queue();
54:
55:     if (argc < 2) {
56:         putinqueue (queue, stdin, "-");
57:     }else {
58:         for (int argi = 1; argi < argc; ++argi) {
59:             if (strcmp (argv[argi], "-") == 0) {
60:                 putinqueue (queue, stdin, "-");
61:             }else {
62:                 putfileinqueue (queue, argv[argi]);
63:             }
64:         }
65:     }
66:
67:     while (! isempty_queue (queue)) {
68:         printf ("%s\n", remove_queue (queue));
69:     }
70:
71:     return exit_status;
72: }
73:
```

```
 1: /* $Id: queue.c,v 1.2 2012-02-21 14:57:52-08 - - $ */
 2:
 3: #include <assert.h>
 4: #include <stdio.h>
 5: #include <stdlib.h>
 6: #include <string.h>
 7:
 8: #include "queue.h"
 9:
10: #define STUBPRINTF(...) fprintf (stderr, __VA_ARGS__);
11:
12: static char *queue_tag = "struct queue";
13: static char *queuenode_tag = "struct queuenode";
14:
15: typedef struct queuenode *queuenode_ref;
16:
17: struct queuenode {
18:     char *tag;
19:     queue_item_t item;
20:     queuenode_ref link;
21: };
22:
23: struct queue {
24:     char *tag;
25:     queuenode_ref front;
26:     queuenode_ref rear;
27: };
28:
29: queue_ref new_queue (void) {
30:     STUBPRINTF ("return NULL\n");
31:     return NULL;
32: }
33:
34: void free_queue (queue_ref queue) {
35:     assert (is_queue (queue));
36:     assert (isempty_queue (queue));
37:     memset (queue, 0, sizeof (struct queue));
38:     free (queue);
39: }
40:
41: void insert_queue (queue_ref queue, queue_item_t item) {
42:     assert (is_queue (queue));
43:     STUBPRINTF ("item =\n\t\"%s\"\n", item);
44: }
45:
46: queue_item_t remove_queue (queue_ref queue) {
47:     assert (is_queue (queue));
48:     assert (! isempty_queue (queue));
49:     STUBPRINTF ("return NULL\n");
50:     return NULL;
51: }
52:
53: bool isempty_queue (queue_ref queue) {
54:     assert (is_queue (queue));
55:     // LINTED (assignment of 32-bit integer to 8-bit integer)
56:     return queue->front == NULL;
57: }
58:
59: bool is_queue (queue_ref queue) {
60:     // LINTED (assignment of 32-bit integer to 8-bit integer)
61:     return queue != NULL && queue->tag == queue_tag;
62: }
63:
```

```
 1: # $Id: Makefile,v 1.3 2012-02-14 20:43:22-08 - - $
 2:
 3: MKFILE     = Makefile
 4: DEPSFILE   = ${MKFILE}.deps
 5: NOINCLUDE  = ci clean spotless
 6: NEEDINCL   = ${filter ${NOINCLUDE}, ${MAKECMDGOALS}}
 7:
 8: GCC        = gcc -g -O0 -Wall -Wextra -std=gnu99
 9: MKDEPS     = gcc -MM
10: LINT       = lint -Xa -fd -m -u -x -errchk=%all
11: GRIND      = valgrind --leak-check=full
12:
13: CSOURCE    = main.c queue.c
14: CHEADER    =        queue.h
15: OBJECTS    = ${CSOURCE:.c=.o}
16: EXECBIN    = catqueue
17: SOURCES    = ${CHEADER} ${CSOURCE} ${MKFILE}
18: LISTSRC    = ${SOURCES} ${DEPSFILE}
19: LISTING    = Listing.catqueue.ps
20: OUTPUT     = output*.lis
21:
22: all : ${EXECBIN}
23:
24: ${EXECBIN} : ${OBJECTS}
25:         ${GCC} -o $@ ${OBJECTS}
26:
27: %.o : %.c
28:         cid + $<
29:         ${GCC} -c $<
30:
31: lint : ${CSOURCE}
32:         ${LINT} ${CSOURCE}
33:
34: ci : ${SOURCES}
35:         cid + ${SOURCES} test*.data
36:
37: lis : ${SOURCES} test
38:         mkpspdf ${LISTING} ${LISTSRC} ${OUTPUT}
39:
40: clean :
41:         - rm ${OBJECTS} ${DEPSFILE} core ${OUTPUT}
42:
43: spotless : clean
44:         - rm ${EXECBIN}
45:
46: test : ${EXECBIN}
47:         - runprogram.perl -x output1.lis -0test1.data ${EXECBIN}
48:         - runprogram.perl -x output2.lis ${EXECBIN} test*.data
49:         - ${GRIND} ${EXECBIN} <test1.data >output3.lis 2>&1
50:
51: deps : ${CSOURCE} ${CHEADER}
52:         @ echo "# ${DEPSFILE} created 'date'" >${DEPSFILE}
53:         ${MKDEPS} ${CSOURCE} | sort | uniq >>${DEPSFILE}
54:
55: ${DEPSFILE} :
56:         @ touch ${DEPSFILE}
57:         ${MAKE} --no-print-directory deps
58:
59:
60: again :
61:         gmake spotless deps ci lint all lis
62:
63: ifeq "${NEEDINCL}" ""
64: include ${DEPSFILE}
```

```
65: endif
66:
```

```
1: # Makefile.deps created Wed Feb 22 19:23:35 PST 2012
2: main.o: main.c queue.h
3: queue.o: queue.c queue.h
```

```
 1:
 2: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
 3: log: output1.log
 4: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
 5:
 6:      1  Script  : /afs/cats.ucsc.edu/courses/cmps012b-wm/bin/runprogram.perl
 7:      2  limit c :     0 max core file size (KB)
 8:      3  limit f : 4194303 max output file size (KB)
 9:      4  limit t : 4294967295 max CPU time (sec)
10:      5  stdin   : test1.data
11:      6  stdout  : output1.out
12:      7  stderr  : output1.err
13:      8  log     : output1.log
14:      9  listing : output1.lis
15:     10  Command : catqueue
16:     11  starting: pid 10788: 19:23:35.00
17:     12  finished: pid 10788: 19:23:35.00, real 0.00, user 0.00, sys 0.00
18:     13  pstatus: 0x0006 TERMINATED 6: Aborted
19:
20: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
21: stdin: test1.data
22: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
23:
24:      1  $Id: test1.data,v 1.1 2012-02-14 20:32:33-08 - - $
25:      2  Test data 1 line 1.
26:      3  Test data 1 line 2.
27:      4  Test data 1 line 3.
28:
29: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
30: stdout: output1.out
31: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
32:
33:
34: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
35: stderr: output1.err
36: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
37:
38:      1  return NULL
39:      2  catqueue: queue.c:42: insert_queue: Assertion 'is_queue (queue)' failed.
```

```
 1:
 2: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
 3: log: output2.log
 4: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
 5:
 6:       1   Script  : /afs/cats.ucsc.edu/courses/cmps012b-wm/bin/runprogram.perl
 7:       2   limit c :    0 max core file size (KB)
 8:       3   limit f : 4194303 max output file size (KB)
 9:       4   limit t : 4294967295 max CPU time (sec)
10:       5   stdin   : /dev/null
11:       6   stdout  : output2.out
12:       7   stderr  : output2.err
13:       8   log     : output2.log
14:       9   listing : output2.lis
15:      10   Command : catqueue test1.data test2.data test3.data
16:      11   starting: pid 10795: 19:23:35.00
17:      12   finished: pid 10795: 19:23:35.00, real 0.00, user 0.00, sys 0.00
18:      13   pstatus: 0x0006 TERMINATED 6: Aborted
19:
20: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
21: stdin: /dev/null
22: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
23:
24:
25: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
26: stdout: output2.out
27: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
28:
29:
30: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
31: stderr: output2.err
32: ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
33:
34:       1   return NULL
35:       2   catqueue: queue.c:42: insert_queue: Assertion 'is_queue (queue)' failed.
```

```
 1: ==10802== Memcheck, a memory error detector
 2: ==10802== Copyright (C) 2002-2009, and GNU GPL'd, by Julian Seward et al.
 3: ==10802== Using Valgrind-3.5.0 and LibVEX; rerun with -h for copyright info
 4: ==10802== Command: catqueue
 5: ==10802==
 6: return NULL
 7: catqueue: queue.c:42: insert_queue: Assertion 'is_queue (queue)' failed.
 8: ==10802==
 9: ==10802== HEAP SUMMARY:
10: ==10802==     in use at exit: 51 bytes in 1 blocks
11: ==10802==   total heap usage: 3 allocs, 2 frees, 225 bytes allocated
12: ==10802==
13: ==10802== LEAK SUMMARY:
14: ==10802==    definitely lost: 0 bytes in 0 blocks
15: ==10802==    indirectly lost: 0 bytes in 0 blocks
16: ==10802==      possibly lost: 0 bytes in 0 blocks
17: ==10802==    still reachable: 51 bytes in 1 blocks
18: ==10802==         suppressed: 0 bytes in 0 blocks
19: ==10802== Reachable blocks (those to which a pointer was found) are not shown.
20: ==10802== To see them, rerun with: --leak-check=full --show-reachable=yes
21: ==10802==
22: ==10802== For counts of detected and suppressed errors, rerun with: -v
23: ==10802== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 4 from 4)
```