```
 1: // $Id: queue.h,v 1.6 2014-02-13 18:59:56-08 - - $
 2:
 3: #ifndef __QUEUE_H__
 4: #define __QUEUE_H__
 5:
 6: #include <stdbool.h>
 7:
 8: typedef struct queue queue;
 9: typedef char *queue_item_t;
10:
11: queue *new_queue (void);
12:
13: void free_queue (queue*);
14:
15: void insert_queue (queue*, queue_item_t);
16:
17: queue_item_t remove_queue (queue*);
18:
19: bool isempty_queue (queue*);
20:
21: #endif
22:
```

```
 1: // $Id: main.c,v 1.10 2013-02-15 17:17:42-08 - - $
 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 *the_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 (the_queue, linepos);
33:    }
34: }
35:
36: void putfileinqueue (queue *the_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 (the_queue, input, filename);
46:       fclose (input);
47:    }
48: }
49:
```

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

```
 1: // $Id: queue.c,v 1.9 2013-05-01 13:22:44-07 - - $
 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: typedef struct queue_node queue_node;
13: struct queue_node {
14:    queue_item_t item;
15:    queue_node *link;
16: };
17:
18: struct queue {
19:    queue_node *front;
20:    queue_node *rear;
21: };
22:
23: queue *new_queue (void) {
24:    STUBPRINTF ("return NULL\n");
25:    return NULL;
26: }
27:
28: void free_queue (queue *this) {
29:    assert (isempty_queue (this));
30:    free (this);
31: }
32:
33: void insert_queue (queue *this, queue_item_t item) {
34:    STUBPRINTF ("item =\n\t\"%s\"\n", item);
35: }
36:
37: queue_item_t remove_queue (queue *this) {
38:    assert (! isempty_queue (this));
39:    STUBPRINTF ("return NULL\n");
40:    return NULL;
41: }
42:
43: bool isempty_queue (queue *this) {
44:    return this->front == NULL;
45: }
46:
```

```
# $Id: Makefile,v 1.6 2014-02-13 18:58:45-08 - - $

MKFILE    = Makefile
DEPSFILE  = ${MKFILE}.deps
NOINCLUDE = ci clean spotless
NEEDINCL  = ${filter ${NOINCLUDE}, ${MAKECMDGOALS}}

GCC       = gcc -g -O0 -Wall -Wextra -std=gnu99
MKDEPS    = gcc -MM
GRIND     = valgrind --leak-check=full

CSOURCE   = main.c queue.c
CHEADER   =        queue.h
OBJECTS   = ${CSOURCE:.c=.o}
EXECBIN   = catqueue
SOURCES   = ${CHEADER} ${CSOURCE} ${MKFILE}
LISTSRC   = ${SOURCES} ${DEPSFILE}
LISTING   = Listing.catqueue.ps
OUTPUT    = output*.lis

all : ${EXECBIN}

${EXECBIN} : ${OBJECTS}
        ${GCC} -o $@ ${OBJECTS}

%.o : %.c
        cid + $<
        ${GCC} -c $<

ci : ${SOURCES}
        cid + ${SOURCES} test*.data

lis : ${SOURCES} test
        mkpspdf ${LISTING} ${LISTSRC} ${OUTPUT}

clean :
        - rm ${OBJECTS} ${DEPSFILE} core ${OUTPUT}

spotless : clean
        - rm ${EXECBIN} ${LISTING} ${LISTING:.ps=.pdf}

test : ${EXECBIN}
        - ${EXECBIN} <test1.data >output1.lis 2>&1
        - ${EXECBIN} test*.data >output2.lis 2>&1
        - ${GRIND} ${EXECBIN} <test1.data >output3.lis 2>&1

deps : ${CSOURCE} ${CHEADER}
        @ echo "# ${DEPSFILE} created `date`" >${DEPSFILE}
        ${MKDEPS} ${CSOURCE} | sort | uniq >>${DEPSFILE}

${DEPSFILE} :
        @ touch ${DEPSFILE}
        ${MAKE} --no-print-directory deps


again :
        gmake spotless deps ci all lis

```

```
59: ifeq "${NEEDINCL}" ""
60: include ${DEPSFILE}
61: endif
62:
```

```
1: # Makefile.deps created Thu Feb 13 18:59:56 PST 2014
2: main.o: main.c queue.h
3: queue.o: queue.c queue.h
```

```
1: return NULL
2: item =
3:         "$Id: test1.data,v 1.1 2012-02-14 20:32:33-08 - - $"
4: item =
5:         "Test data 1 line 1."
6: item =
7:         "Test data 1 line 2."
8: item =
9:         "Test data 1 line 3."
```

```
 1: return NULL
 2: item =
 3:          "$Id: test1.data,v 1.1 2012-02-14 20:32:33-08 - - $"
 4: item =
 5:          "Test data 1 line 1."
 6: item =
 7:          "Test data 1 line 2."
 8: item =
 9:          "Test data 1 line 3."
10: item =
11:          "$Id: test2.data,v 1.1 2012-02-14 20:32:33-08 - - $"
12: item =
13:          "Test data 2 line 1."
14: item =
15:          "Test data 2 line 2."
16: item =
17:          "Test data 2 line 3."
18: item =
19:          "$Id: test3.data,v 1.1 2012-02-14 20:32:33-08 - - $"
20: item =
21:          "Test data 3 line 1."
22: item =
23:          "Test data 3 line 2."
24: item =
25:          "Test data 3 line 3."
```

```
    1: ==1206== Memcheck, a memory error detector
    2: ==1206== Copyright (C) 2002-2012, and GNU GPL'd, by Julian Seward et al.
    3: ==1206== Using Valgrind-3.8.1 and LibVEX; rerun with -h for copyright in
fo
    4: ==1206== Command: catqueue
    5: ==1206==
    6: return NULL
    7: item =
    8:         "$Id: test1.data,v 1.1 2012-02-14 20:32:33-08 - - $"
    9: item =
   10:         "Test data 1 line 1."
   11: item =
   12:         "Test data 1 line 2."
   13: item =
   14:         "Test data 1 line 3."
   15: ==1206== Invalid read of size 8
   16: ==1206==    at 0x400D3C: isempty_queue (queue.c:44)
   17: ==1206==    by 0x400C27: main (main.c:67)
   18: ==1206==  Address 0x0 is not stack'd, malloc'd or (recently) free'd
   19: ==1206==
   20: ==1206==
   21: ==1206== Process terminating with default action of signal 11 (SIGSEGV)
   22: ==1206==  Access not within mapped region at address 0x0
   23: ==1206==    at 0x400D3C: isempty_queue (queue.c:44)
   24: ==1206==    by 0x400C27: main (main.c:67)
   25: ==1206==  If you believe this happened as a result of a stack
   26: ==1206==  overflow in your program's main thread (unlikely but
   27: ==1206==  possible), you can try to increase the size of the
   28: ==1206==  main thread stack using the --main-stacksize= flag.
   29: ==1206==  The main thread stack size used in this run was 10485760.
   30: ==1206==
   31: ==1206== HEAP SUMMARY:
   32: ==1206==     in use at exit: 111 bytes in 4 blocks
   33: ==1206==   total heap usage: 4 allocs, 0 frees, 111 bytes allocated
   34: ==1206==
   35: ==1206== 111 bytes in 4 blocks are definitely lost in loss record 1 of 1
   36: ==1206==    at 0x4A06A2E: malloc (in /opt/rh/devtoolset-2/root/usr/lib64
/valgrind/vgpreload_memcheck-amd64-linux.so)
   37: ==1206==    by 0x3753681081: strdup (strdup.c:43)
   38: ==1206==    by 0x400A43: putinqueue (main.c:30)
   39: ==1206==    by 0x400B80: main (main.c:56)
   40: ==1206==
   41: ==1206== LEAK SUMMARY:
   42: ==1206==    definitely lost: 111 bytes in 4 blocks
   43: ==1206==    indirectly lost: 0 bytes in 0 blocks
   44: ==1206==      possibly lost: 0 bytes in 0 blocks
   45: ==1206==    still reachable: 0 bytes in 0 blocks
   46: ==1206==         suppressed: 0 bytes in 0 blocks
   47: ==1206==
   48: ==1206== For counts of detected and suppressed errors, rerun with: -v
   49: ==1206== ERROR SUMMARY: 2 errors from 2 contexts (suppressed: 6 from 6)
```