

```
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 1.10 2014-02-27 12:12:24-08 - - $
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: 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 = \"%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:
```

```
1: # $Id: Makefile,v 1.6 2014-02-13 18:58:45-08 - - $
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: GRIND        = valgrind --leak-check=full
11:
12: CSOURCE      = main.c queue.c
13: CHEADER      =          queue.h
14: OBJECTS      = ${CSOURCE:.c=.o}
15: EXECBIN      = catqueue
16: SOURCES      = ${CHEADER} ${CSOURCE} ${MKFILE}
17: LISTSRC      = ${SOURCES} ${DEPSFILE}
18: LISTING      = Listing.catqueue.ps
19: OUTPUT       = output*.lis
20:
21: all : ${EXECBIN}
22:
23: ${EXECBIN} : ${OBJECTS}
24:             ${GCC} -o $@ ${OBJECTS}
25:
26: %.o : %.c
27:         cid + $<
28:         ${GCC} -c $<
29:
30: ci : ${SOURCES}
31:     cid + ${SOURCES} test*.data
32:
33: lis : ${SOURCES} test
34:     mkpspdf ${LISTING} ${LISTSRC} ${OUTPUT}
35:
36: clean :
37:     - rm ${OBJECTS} ${DEPSFILE} core ${OUTPUT}
38:
39: spotless : clean
40:     - rm ${EXECBIN} ${LISTING} ${LISTING:.ps=.pdf}
41:
42: test : ${EXECBIN}
43:     - ${EXECBIN} <test1.data >output1.lis 2>&1
44:     - ${EXECBIN} test*.data >output2.lis 2>&1
45:     - ${GRIND} ${EXECBIN} <test1.data >output3.lis 2>&1
46:
47: deps : ${CSOURCE} ${CHEADER}
48:     @ echo "# ${DEPSFILE} created `date`" >${DEPSFILE}
49:     ${MKDEPS} ${CSOURCE} | sort | uniq >>${DEPSFILE}
50:
51: ${DEPSFILE} :
52:     @ touch ${DEPSFILE}
53:     ${MAKE} --no-print-directory deps
54:
55:
56: again :
57:     gmake spotless deps ci all lis
58:
```

01/30/15  
17:42:41

\$cmpps012b-wm/Labs-cmps012m/lab7c-headers-adts/catqueue/  
Makefile

2/2

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

**01/30/15**  
**17:42:41**

\$cmpps012b-wm/Labs-cmps012m/lab7c-headers-adts/catqueue/  
Makefile.deps

**1/1**

```
1: # Makefile.deps created Fri Jan 30 17:42:41 PST 2015
2: main.o: main.c queue.h
3: queue.o: queue.c queue.h
```

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



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

```
1: ==27878== Memcheck, a memory error detector
2: ==27878== Copyright (C) 2002-2013, and GNU GPL'd, by Julian Seward et al
.
3: ==27878== Using Valgrind-3.9.0 and LibVEX; rerun with -h for copyright i
nfo
4: ==27878== Command: catqueue
5: ==27878==
6: return NULL
7: item = "$Id: test1.data,v 1.1 2012-02-14 20:32:33-08 - - $"
8: item = "Test data 1 line 1."
9: item = "Test data 1 line 2."
10: item = "Test data 1 line 3."
11: ==27878== Invalid read of size 8
12: ==27878==    at 0x400D3C: isempty_queue (queue.c:44)
13: ==27878==    by 0x400C27: main (main.c:67)
14: ==27878== Address 0x0 is not stack'd, malloc'd or (recently) free'd
15: ==27878==
16: ==27878==
17: ==27878== Process terminating with default action of signal 11 (SIGSEGV)
18: ==27878== Access not within mapped region at address 0x0
19: ==27878==    at 0x400D3C: isempty_queue (queue.c:44)
20: ==27878==    by 0x400C27: main (main.c:67)
21: ==27878== If you believe this happened as a result of a stack
22: ==27878== overflow in your program's main thread (unlikely but
23: ==27878== possible), you can try to increase the size of the
24: ==27878== main thread stack using the --main-stacksize= flag.
25: ==27878== The main thread stack size used in this run was 10485760.
26: ==27878==
27: ==27878== HEAP SUMMARY:
28: ==27878==    in use at exit: 111 bytes in 4 blocks
29: ==27878==    total heap usage: 4 allocs, 0 frees, 111 bytes allocated
30: ==27878==
31: ==27878== 111 bytes in 4 blocks are definitely lost in loss record 1 of
1
32: ==27878==    at 0x4C27AAA: malloc (in /opt/rh/devtoolset-2/root/usr/lib6
4/valgrind/vgpreload_memcheck-amd64-linux.so)
33: ==27878==    by 0x4EAFE91: strdup (in /lib64/libc-2.12.so)
34: ==27878==    by 0x400A43: putinqueue (main.c:30)
35: ==27878==    by 0x400B80: main (main.c:56)
36: ==27878==
37: ==27878== LEAK SUMMARY:
38: ==27878==    definitely lost: 111 bytes in 4 blocks
39: ==27878==    indirectly lost: 0 bytes in 0 blocks
40: ==27878==    possibly lost: 0 bytes in 0 blocks
41: ==27878==    still reachable: 0 bytes in 0 blocks
42: ==27878==    suppressed: 0 bytes in 0 blocks
43: ==27878==
44: ==27878== For counts of detected and suppressed errors, rerun with: -v
45: ==27878== ERROR SUMMARY: 2 errors from 2 contexts (suppressed: 6 from 6)
```