|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | #includ<stdio.h> | |  | #include <mqueue.h> | |  | #include <fcntl.h> | |  | #include <sys/stat.h> | |  | #include <sys/types.h> | |  | #include <string.h> | |  | #define MAX\_MSG\_SIZE 2048 | |  |  | |  | int main(int argc, char \*argv[]) | |  | { | |  | int md; /\* msg queue descriptor \*/ | |  |  | |  | /\* attributes for the message queue \*/ | |  | struct mq\_attr msgattr; | |  | msgattr.mq\_maxmsg = 10; | |  | msgattr.mq\_msgsize = MAX\_MSG\_SIZE; | |  |  | |  | if ( argc != 2) | |  | { | |  | fprintf(stderr, "Usage: %s 'my message'\n", | |  | argv[0]); | |  | return 1; | |  | } | |  |  | |  | /\* the first arg : the name of the queue, here it's /my-queue | |  | the second arg : the flags are O\_CREATE and O\_RDWR | |  | the third arg : the permission mode; this is the same as for files | |  | the fourth arg : the struct msgattr | |  | \*/ | |  |  | |  | md = mq\_open("/my\_queue", O\_CREAT|O\_RDWR, 0644, &msgattr); | |  | if ( md == -1 ) | |  | { | |  | perror("Creating message queue"); | |  | return 1; | |  | } | |  | if ( (mq\_send(md, argv[1], strlen(argv[1]), 1)) == -1 ) | |  | { | |  | perror("Message queue send"); | |  | return 1; | |  | } | |  | mq\_close(md); | |  | return 0; | |  | } | | |  | | --- | | #include<stdio.h> | |  | #include <mqueue.h> | |  | #include <fcntl.h> | |  | #include <sys/stat.h> | |  | #include <sys/types.h> | |  | #include <stdlib.h> | |  | #include <string.h> | |  |  | |  | int main(void) | |  | { | |  | int md; /\* msg queue descriptor \*/ | |  | char \*buffer; | |  | struct mq\_attr msgattr; | |  |  | |  | /\* to open the message queue mq\_open() | |  | the first arg : the name of the queue | |  | the second arg : the flags , only want to read the queue | |  | \*/ | |  | md = mq\_open("/my\_queue", O\_RDONLY); | |  |  | |  | if (md == -1 ) | |  | { | |  | perror("Open message queue"); | |  | return 1; | |  | } | |  |  | |  | /\* to get the attributes of the message queue | |  | using mq\_getattr(). | |  | Once we have the attributes of the message, we use | |  | its mq\_msgsize() member to allocate memory for a message | |  | of that size using calloc() | |  | \*/ | |  | if ( (mq\_getattr(md, &msgattr)) == -1 ) | |  | { | |  | perror("Get message attribute"); | |  | return 1; | |  | } | |  |  | |  |  | |  | /\* return a pointer to that memory, that's buffer \*/ | |  | buffer = calloc(msgattr.mq\_msgsize, sizeof(char)); | |  |  | |  | if (buffer == NULL) | |  | { | |  | fprintf(stderr, "Couldn't allocate memory"); | |  | return 1; | |  | } | |  |  | |  | /\* | |  | msgattr.mq\_curmsgs : the number of messages currently | |  | in the queue | |  | \*/ | |  | printf("%ld messages in queue\n", msgattr.mq\_curmsgs); | |  |  | |  | for (int i = 0; i < msgattr.mq\_curmsgs; i++) | |  | { // we receive a message using mq\_receive | |  | if ( (mq\_receive(md, buffer, msgattr.mq\_msgsize, NULL)) == -1 ) | |  | { | |  | perror("Message receive"); | |  | return 1; | |  | } | |  | printf("%s\n", buffer); // print the message | |  | // reset the entire memory to NULL characters using memset() | |  | memset(buffer, '\0', msgattr.mq\_msgsize); | |  | } | |  |  | |  | free(buffer); | |  | mq\_close(md); | |  | mq\_unlink("/my\_queue"); | |  | return 0; | |  | } | |