

# Computation structures

## Problem-solving lesson 7

1. Give an implementation of the mutual exclusion between 2 processes using only the blocking message queues of size 0 as synchronization mechanism.
2. Comment the following programs. Can they be use to implement a rendezvous between two scripts?

<pre>1 #include &lt;stdio.h&gt;   #include &lt;stdlib.h&gt;   #include &lt;sys/msg.h&gt;    #define MSGLEN 128 6 #define KEY 345782    struct {     long mtype;     char buf[MSGLEN]; 11 } msg;    int main() {     int qid;  16    if ((qid=msgget(KEY,IPC_CREAT 0666)) &lt; 0)       die("couldn't access the queue");      if (msgrcv(qid,&amp;msg,MSGLEN,1,0) &lt; 0) 21    die("failed to receive");     printf("got '%s'\n",msg.buf);      if (msgctl(qid,IPC_RMID,0) &lt; 0) 22    die("warning: trailing queue");  26    return EXIT_SUCCESS;   }</pre>	<pre>2 /* --&gt; reuse lines 1..11 of receiver */    int main(int argc, char ** argv) {      if (argc != 2) {       fprintf(stderr, 7        "Usage : %s &lt;message&gt;\n",         argv[0]);       return EXIT_FAILURE;     }  12    int qid;      if ((qid=msgget(KEY,IPC_CREAT 0666)) &lt; 0)       die("couldn't access the queue");      msg.mtype = 1;     strncpy(msg.buf, argv[1], MSGLEN);     if (msgsnd(qid,&amp;msg,MSGLEN,0xCAFE) &lt; 0)       die("failed to send");  22    return EXIT_SUCCESS;   }    int die(char *msg) {     perror(msg); exit(EXIT_FAILURE); 27 }</pre>
---	--

3. Modify the above programs in order to design a reader and a writer that communicate through a message queue:
  - The writer sends messages coming from the standard input (**stdin**) on the queue and ends by sending the “.” symbol.
  - The reader displays the messages from the queue on the standard output (**stdout**) and stops when it receives the “.” symbol .
4. Simulate a message queue using only semaphores and shared memory. For simplicity, we consider the case of only two processes sending each other integer values as messages.