

# Computation structures

## Problem-solving lesson 8

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?

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <sys/msg.h>
4
5 #define MSGLEN 128
6 #define KEY 345782
7
8 struct {
9     long mtype;
10    char buf[MSGLEN];
11 } msg;
12
13 int main() {
14     int qid;
15
16     if ((qid=msgget(KEY,IPC_CREAT|0666))<0)
17         die("couldn't access the queue");
18
19     if (msgrcv(qid,&msg,MSGLEN,1,0)<0)
20         die("failed to receive");
21     printf("got '%s'\n",msg.buf);
22
23     if (msgctl(qid,IPC_RMID,0) < 0)
24         die("warning: trailing queue");
25
26     return EXIT_SUCCESS;
27 }
28
29 /** --> reuse lines 1..11 of receiver **/
30
31 int main(int argc, char ** argv) {
32
33     if (argc != 2) {
34         fprintf(stderr,
35             "Usage : %s <message>\n",
36             argv[0]);
37         return EXIT_FAILURE;
38     }
39
40     int qid;
41
42     if ((qid=msgget(KEY,IPC_CREAT|0666))<0)
43         die("couldn't access the queue");
44
45     msg.mtype = 1;
46     strncpy(msg.buf, argv[1], MSGLEN);
47     if (msgsnd(qid,&msg,MSGLEN,0xCAFE) < 0)
48         die("failed to send");
49
50     return EXIT_SUCCESS;
51 }
52
53 int die(char *msg) {
54     perror(msg); exit(EXIT_FAILURE);
55 }
```

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.