

Rewrite the programs for program 3 using a UNIX message queue.
Here's how you get the queue:
Place the following includes in your program:

```
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/msg.h>
```

(You should have ipc.h and types.h already. You don't need the shared memory stuff any more.)
Get a handle to the message queue using the following function call:

```
int MyMsgId = msgget(MyLastFour, IPC_CREAT|0x1c0);
```

The first parameter is the last four digits of your ssn as before. The second parameter is the same as the third parameter of the shmget function call. To destroy the message queue (which is done at the same time that you would normally destroy the shared memory segment) use the following function call:

```
msgctl(MyMsgId, IPC_RMID, NULL);
```

To send a message do the following:
Create a structure with the following format:

```
struct MyMsg
{
    long MType;
    long RandNum;
};
```

Create a variable of this type:

```
MyMsg Message;
```

Fill it in as follows: (do this in your message-sending loop.)

```
Message.MType = 1; // required
RandNum = rand(); // this is your random number
```

Send the message using the following function call.

```
msgsnd(MyMsgId,&Message,sizeof(long),0);
```

To receive a message use the same structure in the reader process and declare the same variable, Message. Use the following function call to obtain the message data.

```
msgrcv(MyMsgId,&Message,sizeof(long),0,0); // two zero arguments at end.
```

The random number will be in the RandNum component of the Message variable.