|  |  |
| --- | --- |
| ***name*** |  |
| ***id*** |  |

Birzeit University - Faculty of Engineering Technology

Electrical & Computer Engineering Department – ENCS531

Real-Time Applications and Embedded Systems – 2nd semester - 2015/16

Quiz #2

|  |
| --- |
| Real-Time Applications and Embedded Systems  |

We only have one license of Matlab installed on the server. We need to use Matlab from other machines on the network. So we have designed a solution as follows. The server will have a TCP server that will get requests from different clients. And send the requests to the Matalab process using a message queue. The the TCP server get the result from the Matlab process and return it to the client over TCP then close the connection.

You task is to write the TCP server to get requests from clients send it to Matalab over Message Queue and get results and send it to Client.

The structure used between TCP client and server is

Struct \_call {

Char name[40]; // function name

Int argc; // number of arguments for function

Int\* args; // array of arguments

Int retVal; // return value from server to client;

}

The message queue message from tcp server to Matlab should have:

* Type 1
* All data from client
* Anything else you need to add to in order to know which reply belong to which client.

Note: you can use your notes from the lectures in the quiz, you can ask for the prototype of any function you need.

Struct mat\_call

{ long type;

 Struct \_call data;

 Int client\_fd;

Int pid;

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Int fd;

Fd=socket(AF\_INET,SOCK\_STREAM,0);

Sockaddr\_in srv;

Srv.sin\_family-AF\_INET;

Srv.sin\_port= ?? ;

Srv.sin\_addr.s\_addr= ????;

Bind (fd,&srv, sizeof(srv));

Listen(fd,??);

Int qid;

Qid=msgget(KEY, IPC\_CREATE|0666);

While (1)

{ \_call req;

 Mat\_call matReq;

 Int clienfd=accept (fd, ….. );

// clients must not block each other using threads fork , select , stc…

Read(clientfd,&req,sizeof(req));

matReq.type=1;

matReq.data.name=req.name;

matReq.data.argc=req.argc;

matReq.data.argv=req.argv; // this should be an array copy not just pointer

matReq.pid=getpid();

matReq.client\_fd=clientfd;

msgsend(qid, &matReq,sizeof(matReq)-4,0);

if (msgrcv(msgsend(qid, &matReq,sizeof(matReq)-4, getpid(),IPC\_NOWAIT) >0)

{

 Write (matReq.client\_fd,&(matReq.data), sizeof (matReq.data));

Close (matReq.client\_fd);

}

}

}