Socket Programming Guideline

- ✓ **The overall structure of the Socket Server is organized as follows**

  ① Create a TCP socket

  ② Assign a port to socket

  ③ Set socket to listen

  ④ Repeatedly :

  A. Accept new connection

  B. Communicate

  C. Close the connection

- ✓ **The overall structure of the Socket Client is organized as follows**

  ① Create a TCP socket

  ② Establish connection

  ③ Communicate

  ④ Close the connection

- ✓ **Functions of Socket**

  ① Create : int sockid = socket(family, type, protocol);

  sockid :socket descriptor

  family : communication domain

  type : communication type

  protocol : specifies protocol

  ② Bind : int status = bind(sockid, &addrport, size);

  sockid : socket descriptor

  addrport : struct sockaddr, the address and port of the machine

  size : the size of the addrport structure

  status : upon failure -1 is returned

③ **Listen : int status = listen(sockid, queueLimit);**

sockid : socket descripto

queuelen : active participants that can “wait” for a connection

status : 0 if listening, -1 if error

④ **Accept : int status = connect(sockid, &foreignAddr, addrlen);**

Sockid: socket to be used in connection

foreignAddr : address of the passive participant

addrlen: sizeof(name)

status : 0 if successful connect, -1 otherwise

⑤ **Send : int count = send(sockid, msg, msgLen, flags);**

Msg : message to be transmitted

⑥ **Receive : int count = recv(sockid, recvBuf, bufLen, flags);**

recvBuf : stores received bytes

bufLen : bytes received

flags : usually just 0

count : bytes received(-1 if error)

⑦ **Close : status = close(sockid);**

sockid : the file descriptor(being closed)

status : 0 if successful, -1 if error