

socket communication: send & receive doesn't work right

Source:

<http://www.tech-archive.net/Archive/Development/microsoft.public.win32.programmer.networks/2007-04/msg00282>

- *From:* Ananya <Ananya@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Sat, 28 Apr 2007 02:38:00 -0700
-

I am trying to establish socket communication between my Java and C++ program.

I called my Java program from my C++ program with ShellExecuteEx.
I created a C++ Server and a Java Client, which is accepted by the Server.

I did a test of sending two doubles:

1.23 & 4.5

from my Java program to my C++ program, however I always received the following 2 different doubles:

1.1648250968930678e-302 & -6.4627233651951511e-086.

Here is my Java sending method:

```
public void send_doubles(double vals[], int len) throws IOException
{
    // convert our array of doubles into an array of bytes
    ByteArrayOutputStream bytestream;
    bytestream = new ByteArrayOutputStream(len*8);

    DataOutputStream out;
    out = new DataOutputStream(bytestream);

    for (int i=0; i<len; i++)
    {
        out.writeDouble(vals[i]);
    }

    output.write(bytestream.toByteArray(), 0, bytestream.size());
    output.flush();

    recv_ack();
    send_ack();
}
```

and my Java acknowledgement methods:

```
// send a short acknowledgement to the server
private void send_ack() throws IOException
{
```

socket communication: send & receive doesn't work right

```
int ack;

ack = 0;

output.write(ack);
output.flush();
}

// recv a short acknowledgment from the server
private void recv_ack() throws IOException
{
int ack;

ack = (int)input.read();
}
```

And here is my C++ receiving method:

```
int Server::recv_doubles(double *val, int maxlen) throw (string)
{
int i, j;
int numbytes = 0;
int end = 0;
int total_bytes = 0;
char *temp;
char *result;

temp = (char *)buffer;
result = (char *)buffer2;

j = 0;

// we are receiving the incoming doubles one byte at a time
while (!end)
{
if ((numbytes=recv(new_fd, temp, BUFSIZE, 0))==-1)
{
throw string("help!");
}

for (i=0; i<numbytes; i++)
{
result[j] = temp[i];
j++;
}

total_bytes = total_bytes + numbytes;
if (total_bytes==maxlen*sizeof(double) + 1)
{
end = 1;
}
}
```

socket communication: send & receive doesn't work right

socket communication: send & receive doesn't work right

```
// now we need to put the array of bytes into the array of doubles
char *ptr;
int num = (j - 1)/sizeof(double);

ptr = (char *)val;

// going from Java to C++, we need to reverse the order of each set of bytes
for (i = 0; i < num; i++)
{
for (j=0; j<sizeof(double); j++)
{
ptr[i*sizeof(double)+j] = (char)result[(i+1)*sizeof(double)-j-1];
}
}

send_ack();
recv_ack();

return num;
}
```

and my C++ acknowledgement methods:

```
// receive a short acknowledgement from the client
void Server::recv_ack()
{
char temp[1];
int total = 0;

while (total<1)
{
total += recv(new_fd, temp, 1, 0);
}
}

// send a short acknowledgement to the client
void Server::send_ack()
{
char temp[1];
temp[0] = 42;

send(new_fd, temp, 1, 0);
}
```

Why does my C++ program receive incorrect doubles?

Thanks for looking at my code!

.