

Re: socket communication: send & receive doesn't work right

Source:

<http://www.tech-archive.net/Archive/Development/microsoft.public.win32.programmer.networks/2007-04/msg00332>

- *From:* Ananya <Ananya@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Mon, 30 Apr 2007 12:20:02 -0700
-

Ok, so why did you say that I can eliminate all issues of endian-ness by sending strings, if that also sends bytes?

"Michael K. O'Neill" wrote:

Sorry for looseness in terminology, but you always send bytes. The interpretation of the data (as a string or as a double) is up to you and your code.

Mike

"Ananya" <Ananya@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in message news:AA6F501C-7A17-4D06-8435-ACF3A401E8DB@xxxxxxxxxxxxxxxxxxxx

How can I send the string as string (not bytes) in Java and how can I receive it in C++?

"Ananya" wrote:

Ok, but Michael suggested to send a string for getting around the bytes endian problem.

So I don't want to send a string as bytes. Can I send a string as a string?

Re: socket communication: send & receive doesn't work right

"Scherbina Vladimir" wrote:

If I recall correctly you can convert "string"
data type on Java side

to a

bytes array using .getBytes(...) or something
like that.

--

--Vladimir, Windows SDK MVP

"Ananya"

<Ananya@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>

wrote in message

news:21045337-B7CE-4FC2-86B0-4631DCBFF7D1@xxxxxxxxxxxxxxxxxxxx

That sounds great! But can I
actually send and receive
strings? I
thought
that I only can send and
receive bytes. How do I
send and receive
strings?

Thanks in advance for your
answer!

"Michael K. O'Neill" wrote:

You can
eliminate all
issues of
endian-ness
by
converting
the

doubles to

strings
(sprintf-style)
and sending
the string.
On the
receiving

Re: socket communication: send & receive doesn't work right

side,

use
an atof()
function to
convert
back.

Note that
this
approach
also
resolves all
issues
concerning
whether

the

sending and
receiving
side are
both using
the same
implementation

of

IEEE-754
for storing
double
precision
numbers.
Since string

equivalents

are
sent, the
precise
implementation
simply
doesn't
matter.

Finally, it's
often easier
to debug,
since you
can "see"
the number

Re: socket communication: send & receive doesn't work right

that's
being sent.

Mike

"Scherbina
Vladimir"
<v_scherbina@xxxxxxxxxxxxxxxxxx>

wrote in
message

news:EB69C218-6DAE-4776-ABC0-F601C7878116@xxxxxxxxxx

In
Windows
everything
is
little
endian,
<http://support.microsoft.com/kb/q102025/>.
I
suggest
you
to
send

simple

data

from
java
program,
and
analyze
what
you
obtain
on
c++
side.
For
example,
send
an
integer
value
10,
and
tell
us
here

Re: socket communication: send & receive doesn't work right

what
you
recieve
in

C++

application.

--

--Vladimir,

Windows

SDK

MVP

"Ananya"

<Ananya@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>

wrote

in

message

news:7E0E8354-A53A-4B8F-A0B4-33A51785FEE4@xxx

Well,

I

tried

not

to

reverse

bytes

by

saying:

for

(i

=

0;

i

<

j;

i++)

{

ptr[i]

=

result[i];

}

instead

of:

for

(i

=

0;

i

Re: socket communication: send & receive doesn't work right

```
<
num;
i++)
{
for
(j=0;
j<sizeof(double);
j++)
{
ptr[i*sizeof(double)+j]
=
(char)result[(i+1)*sizeof(double)-j-1];
}
}
in
my
C++
receiving
method.

But
now
the
doubles:
1.23
&
4.5
which
I
send
from
my
Java
program
always
become:
1.8584604523406555e+038
&
5.910042899492e-318#DEN
in
my
C++
program.

Please
help!
By
the
way,
```

Re: socket communication: send & receive doesn't work right

Re: socket communication: send & receive doesn't work right

how
can
I
test
if
my
server
needs

reversing

bytes
or
not?

"Scherbina
Vladimir"
wrote:

I
am
saying
that
if
your
server
is
big
endian,
then
you

_do_not_

need

to

reverse
bytes.

--

--Vladimir,
Windows
SDK
MVP
"Ananya"

<Ananya@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>
wrote
in

Re: socket communication: send & receive doesn't work right

message

news:8B29C2D6-060A-41BA-8180-F4AA

Thanks!

Are

you

saying

that

in

my

C++

receiving

method

the

code

for

reversing

the

byte

order

is

incorrect?

Where

can

I

find

the

correct

code?

"Scherbina

Vladimir"

wrote:

The

problem

with

this

approach

is

that

Java

stores

the

binary

Re: socket communication: send & receive doesn't work right

Re: socket communication: send & receive doesn't work right

data

as

big
endians
only
(no
matter
what
CPU
architecture
is),
if
your

C++

client

is
little
endian
then
unpredictable
results
may
be
obtained.

Check

this

issue.

--
--Vladimir,

Windows

SDK

MVP

"Ananya"

<Ananya@xxxxxxxxxxxxxxxxxxxxxx

wrote

in

message

news:2951DED5-4DDF-40

Re: socket communication: send & receive doesn't work right

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

Re: socket communication: send & receive doesn't work right

sending
two
doubles:
1.23
&
4.5
from
my
Java
program
to
my
C++
program,
however
I
always
received
the
following
2
different
doubles:
1.164825096893067
&
-6.46272336519515

Here
is
my
Java
sending
method:
public
void
send_doubles(double
vals[],
int
len)
throws

IOException

```
{  
//  
convert  
our  
array  
of  
doubles  
into
```

Re: socket communication: send & receive doesn't work right

```
bytestream.size());
```

```
an  
array  
of  
bytes  
ByteArrayOutputStr  
bytestream;  
bytestream  
=  
new  
ByteArrayOutputStr  
  
DataOutputStream  
out;  
out  
=  
new  
DataOutputStream(b  
  
for  
(int  
i=0;  
i<len;  
i++)  
{  
out.writeDouble(vals  
}  
  
output.write(bytestre  
0,  
  
output.flush();  
  
recv_ack();  
send_ack();  
}  
  
and  
my  
Java  
acknowledgement  
methods:  
//  
send  
a  
short  
acknowledgement  
to  
the  
server
```

Re: socket communication: send & receive doesn't work right

```
private
void
send_ack()
throws
IOException
{
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_double(
*val,
int
maxlen)
```

Re: socket communication: send & receive doesn't work right

```
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

Re: socket communication: send & receive doesn't work right

```
while
(!end)
{
if
((numbytes=recv(new
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==maxle
+
1)
{
end
=
1;
}
}

//
now
we
need
to
put
the
array
of
bytes
into
```

Re: socket communication: send & receive doesn't work right

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);
```

Re: socket communication: send & receive doesn't work right

```
j++)
{
ptr[i*sizeof(double)-
=
(char)result[(i+1)*si
}
}

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);
}
}

//
```

Re: socket communication: send & receive doesn't work right

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

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