

Re: How to ping in Python?

Source: <http://coding.derkeiler.com/Archive/Python/comp.lang.python/2005-12/msg00909.html>

- *From:* Larry Bates <larry.bates@xxxxxxxxxxxx>
 - *Date:* Mon, 05 Dec 2005 14:45:14 -0600
-

Seems that you must change the following lines to work:

```
if prop.platform == 'darwin':
myChecksum = socket.htons(myChecksum) & 0xffff
else:
myChecksum = socket.htons(myChecksum)
```

to

```
if sys.platform == 'darwin':
myChecksum = socket.htons(myChecksum) & 0xffff
else:
myChecksum = socket.htons(myChecksum)
```

You also must import the following modules:

```
import socket
import os
import sys
import struct
import time
import select
```

-Larry Bates

dwelch wrote:

> Nico Grubert wrote:

>

>> Hi there,

>>

>> I could not find any "ping" Class or Handler in python (2.3.5) to ping
>> a machine.

>> I just need to "ping" a machine to see if its answering. What's the
>> best way to do it?

>>

>> Kind regards,

>> Nico

>

>

Re: How to ping in Python?

```
>
> # Derived from ping.c distributed in Linux's netkit. That code is
> # copyright (c) 1989 by The Regents of the University of California.
> # That code is in turn derived from code written by Mike Muuss of the
> # US Army Ballistic Research Laboratory in December, 1983 and
> # placed in the public domain. They have my thanks.
>
> # Bugs are naturally mine. I'd be glad to hear about them. There are
> # certainly word-size dependences here.
>
> # Copyright (c) Matthew Dixon Cowles, <http://www.visi.com/~mdc/>.
> # Distributable under the terms of the GNU General Public License
> # version 2. Provided with no warranties of any sort.
>
> # Note that ICMP messages can only be sent from processes running
> # as root.
>
> # Revision history:
> #
> # November 22, 1997
> # Initial hack. Doesn't do much, but rather than try to guess
> # what features I (or others) will want in the future, I've only
> # put in what I need now.
> #
> # December 16, 1997
> # For some reason, the checksum bytes are in the wrong order when
> # this is run under Solaris 2.X for SPARC but it works right under
> # Linux x86. Since I don't know just what's wrong, I'll swap the
> # bytes always and then do an htons().
> #
> # December 4, 2000
> # Changed the struct.pack() calls to pack the checksum and ID as
> # unsigned. My thanks to Jerome Poincheval for the fix.
> #
>
> # From /usr/include/linux/icmp.h; your milage may vary.
> ICMP_ECHO_REQUEST = 8 # Seems to be the same on Solaris.
>
> # I'm not too confident that this is right but testing seems
> # to suggest that it gives the same answers as in_cksum in ping.c
> def checksum(str):
>     csum = 0
>     countTo = (len(str) / 2) * 2
>     count = 0
>     while count < countTo:
>         thisVal = ord(str[count+1]) * 256 + ord(str[count])
>         csum = csum + thisVal
>         csum = csum & 0xffffffffL # Necessary?
>         count = count + 2
>
>
> if countTo < len(str):
```

Re: How to ping in Python?

Re: How to ping in Python?

```
> csum = csum + ord(str[len(str) - 1])
> csum = csum & 0xffffffffL # Necessary?
>
> csum = (csum >> 16) + (csum & 0xffff)
> csum = csum + (csum >> 16)
> answer = ~csum
> answer = answer & 0xffff
>
> # Swap bytes. Bugger me if I know why.
> answer = answer >> 8 | (answer << 8 & 0xff00)
>
> return answer
>
> def receiveOnePing(mySocket, ID, timeout):
> timeLeft = timeout
>
> while 1:
> startedSelect = time.time()
> whatReady = select.select([mySocket], [], [], timeLeft)
> howLongInSelect = (time.time() - startedSelect)
>
> if whatReady[0] == []: # Timeout
> return -1
>
> timeReceived = time.time()
> recPacket, addr = mySocket.recvfrom(1024)
> icmpHeader = recPacket[20:28]
> typ, code, checksum, packetID, sequence = struct.unpack("bbHHh",
> icmpHeader)
>
> if packetID == ID:
> bytesInDouble = struct.calcsize("d")
> timeSent = struct.unpack("d", recPacket[28:28 +
> bytesInDouble])[0]
> return timeReceived - timeSent
>
> timeLeft = timeLeft - howLongInSelect
>
> if timeLeft <= 0:
> return -1
>
> def sendOnePing(mySocket, destAddr, ID):
> # Header is type (8), code (8), checksum (16), id (16), sequence (16)
> myChecksum = 0
>
> # Make a dummy heder with a 0 checksum.
> header = struct.pack("bbHHh", ICMP_ECHO_REQUEST, 0, myChecksum, ID, 1)
> bytesInDouble = struct.calcsize("d")
> data = (192 - bytesInDouble) * "Q"
> data = struct.pack("d", time.time()) + data
>
```

Re: How to ping in Python?

```
> # Calculate the checksum on the data and the dummy header.
> myChecksum = checksum(header + data)
>
> # Now that we have the right checksum, we put that in. It's just easier
> # to make up a new header than to stuff it into the dummy.
> if prop.platform == 'darwin':
> myChecksum = socket.htons(myChecksum) & 0xffff
> else:
> myChecksum = socket.htons(myChecksum)
>
> header = struct.pack("bbHHh", ICMP_ECHO_REQUEST, 0,
> myChecksum, ID, 1)
>
> packet = header + data
> mySocket.sendto(packet, (destAddr, 1)) # Don't know about the 1
>
> def doOne(destAddr, timeout=10):
> # Returns either the delay (in seconds) or none on timeout.
> icmp = socket.getprotobyname("icmp")
> mySocket = socket.socket(socket.AF_INET, socket.SOCK_RAW, icmp)
> myID = os.getpid() & 0xFFFF
> sendOnePing(mySocket, destAddr, myID)
> delay = receiveOnePing(mySocket, myID, timeout)
> mySocket.close()
>
> return delay
>
>
>
> def ping(host, timeout=1):
> dest = socket.gethostbyname(host)
> delay = doOne(dest, timeout)
> return delay
>
.
```

• *References:*

◆ *How to ping in Python?*

◇ *From:* Nico Grubert

◆ *Re: How to ping in Python?*

◇ *From:* dwelch

• Prev by Date: *Re: ANN: Dao Language v.0.9.6-beta is release!*

• Next by Date: *Re: Bitching about the documentation...*

• Previous by thread: *Re: How to ping in Python?*

• Next by thread: *Re: How to ping in Python?*

• Index(es):

◆ *Date*

◆ *Thread*