Computer Network Laboratory

Basic Network Programming (II)

Jiawei Chang

Dept. of Computer Science and Information Engineering

National Taichung University of Science and Technology

Outline

- Ubuntu OS

1. modify_buff_size
2. socket_modes
3. reuse_socket_address
4. print_machine_time

modify_buff_size

```python
import socket

SEND_BUF_SIZE = 4096
RECV_BUF_SIZE = 4096

def modify_buff_size():
    sock = socket.socket( socket.AF_INET, socket.SOCK_STREAM )

    # Get the size of the socket's send buffer
    bufsize = sock.getsockopt(socket.SOL_SOCKET, socket.SO_SNDBUF)
    print ("Buffer size [Before]:%d" %bufsize)

    sock.setsockopt(socket.SOL_TCP, socket.TCP_NODELAY, 1)
    sock.setsockopt(
            socket.SOL_SOCKET,
            socket.SO_SNDBUF,
            SEND_BUF_SIZE)
    sock.setsockopt(
            socket.SOL_SOCKET,
            socket.SO_RCVBUF,
            RECV_BUF_SIZE)
    bufsize = sock.getsockopt(socket.SOL_SOCKET, socket.SO_SNDBUF)
    print ("Buffer size [After]:%d" %bufsize)

if __name__ == '__main__':
    modify_buff_size()
```

```
Buffer size [Before]:65536
Buffer size [After]:4096
```

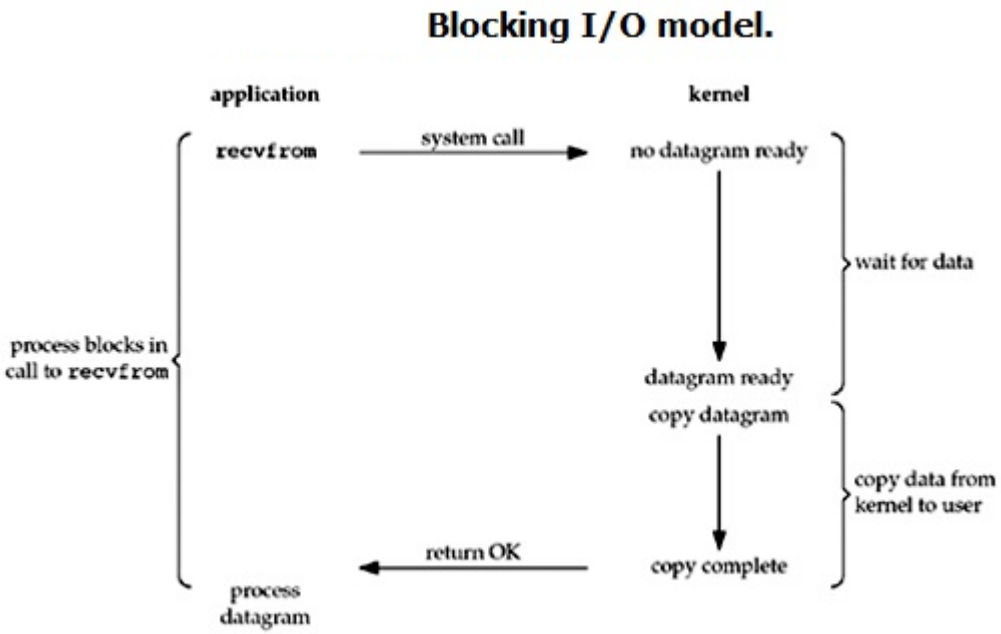
socket_modes

```python
import socket

def test_socket_modes():
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.setblocking(0) #0 = non-blocking mode, 1 = blocking mode and default value
    s.settimeout(0.5)
    s.bind(("127.0.0.1", 0))

    socket_address = s.getsockname()
    print ("Trivial Server launched on socket: %s" %str(socket_address))
    while(1):
        s.listen(1)

if __name__ == '__main__':
    test_socket_modes()
```

```
Trivial Server launched on socket: ('127.0.0.1', 64604)
```

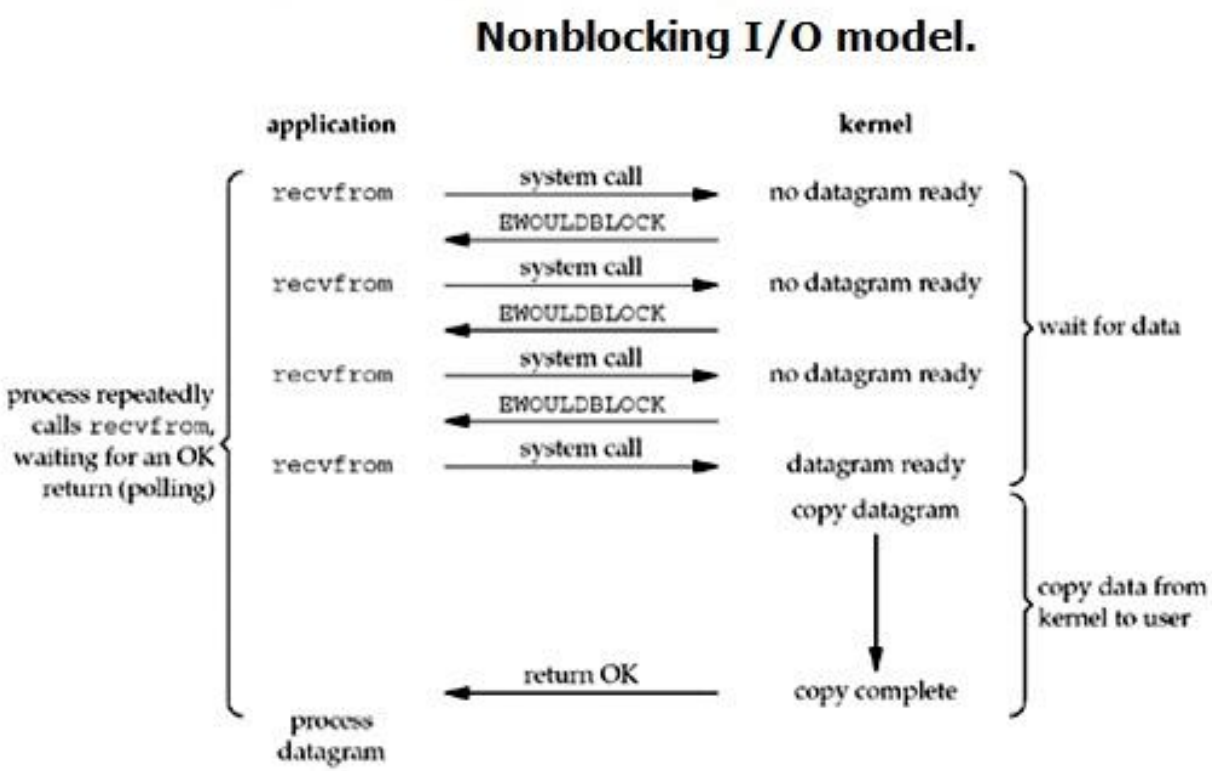
Blocking and non-Blocking

- 在預設的狀況，Server 通常是開啟 block 機制(同步)，將導致程式在執行時會被阻塞住，導致暫停執行。
- 非同步 I/O 的想法其實很單純，假如程式在執行過程中因為 I/O 暫停，但如果不會被阻塞住就能暫時把控制權切換給其它程式，這樣就不會浪費執行時間。



Blocking and non-Blocking

1. 當使用者程序發出read操作時，如果kernel中的資料還沒有準備好，那麼它並不會block使用者程序，而是立刻返回一個error。
2. 從使用者程序角度講 ，它發起一個read操作後，並不需要等待，而是馬上就得到了一個結果。
3. 使用者程序判斷結果是一個error時，它就知道資料還沒有準備好，於是它可以再次傳送read操作。
4. 一旦kernel中的資料準備好了，並且又再次收到了使用者程序的system call，那麼它馬上就將資料拷貝到了使用者記憶體，然後返回。
5. 所以，使用者程序其實是需要不斷的主動詢問kernel資料好了沒有。



reuse_socket_address

```python
import socket
import sys


def reuse_socket_addr():
    sock = socket.socket( socket.AF_INET, socket.SOCK_STREAM )

    # Get the old state of the SO_REUSEADDR option
    old_state = sock.getsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR )
    print ("Old sock state: %s" %old_state)

    # Enable the SO_REUSEADDR option
    sock.setsockopt( socket.SOL_SOCKET, socket.SO_REUSEADDR, 1 )
    new_state = sock.getsockopt( socket.SOL_SOCKET, socket.SO_REUSEADDR )
    print ("New sock state: %s" %new_state)

    local_port = 8282

    srv = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    srv.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
    srv.bind( ('', local_port) )
    srv.listen(1)
    print ("Listening on port: %s " %local_port)
    while True:
        try:
            connection, addr = srv.accept()
            print ('Connected by %s:%s' % (addr[0], addr[1]))
        except KeyboardInterrupt:
            break
        except socket.error as msg:
            print ('%s' % (msg,))


if __name__ == '__main__':
    reuse_socket_addr()
```

```
Old sock state: 0
New sock state: 1
Listening on port: 8282
Connected by 127.0.0.1:36272
Connected by 127.0.0.1:36406
Connected by 127.0.0.1:36412
```

reuse_socket_address

```
jwchang@ubuntu: ~
File  Edit  View  Search  Terminal  Help
jwchang@ubuntu:~$ telnet localhost 8282
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
qq
qq^]
'^]'
'Q



Connection closed by foreign host.
jwchang@ubuntu:~$ telnet localhost 8282
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
```

```
jwchang@ubuntu: ~
File  Edit  View  Search  Terminal  Help
jwchang@ubuntu:~$ telnet localhost 8282
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Connection closed by foreign host.
jwchang@ubuntu:~$
```

print_machine_time

```python
import ntplib
from time import ctime


def print_time():
    ntp_client = ntplib.NTPClient()
    response = ntp_client.request('pool.ntp.org')
    print (ctime(response.tx_time))


if __name__ == '__main__':
    print_time()
```

```
Sun Mar  3 22:43:32 2019
```

延伸閱讀

- Socket Programming in Python (Guide)
  - https://realpython.com/python-sockets/#socket-api-overview

- Python 网络编程
  - http://www.runoob.com/python/python-socket.html

Resource is available by
https://jiaweichang.github.io/biography/

THANKS