

Network Programming

Lecture 3—Elementary Sockets II: I/O Multiplexing

Lei Wang

lei.wang@dlut.edu.cn

Dalian University of Technology

Dec 3, 2008

Part 2. Elementary Sockets II: I/O Multiplexing

1 I/O Multiplexing: `select` and `poll` function

- Introduction
- I/O Models
- `select` Function
- `pselect` Function
- `poll` Function

Introduction

I/O multiplexing is typically used in the following scenarios:

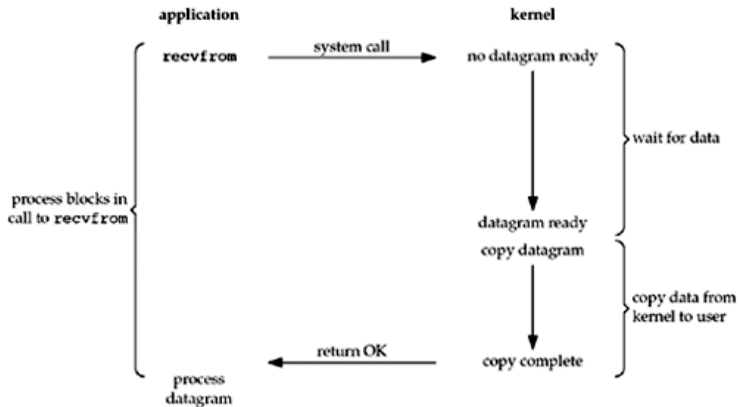
- When a client is handling multiple descriptors (normally interactive input and a network socket).
- It is possible, but rare, for a client to handle multiple sockets at the same time.
- If a TCP server handles both a listening socket and its connected sockets.
- If a server handles both TCP and UDP.
- If a server handles multiple services and perhaps multiple protocols (e.g., the inetd daemon).

Note: I/O multiplexing is not limited to network programming.

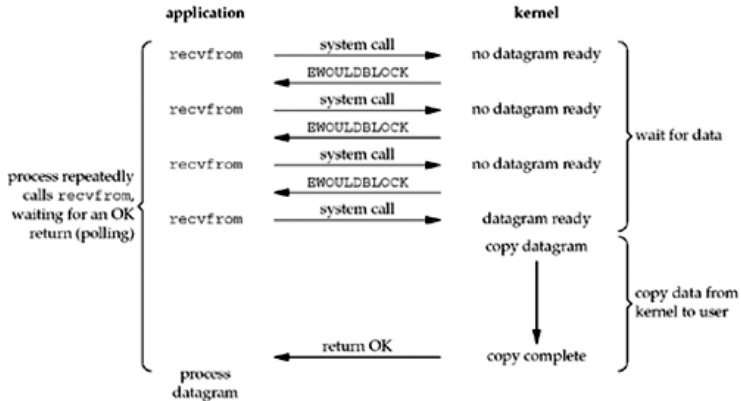
I/O Models

- blocking I/O
- nonblocking I/O
- I/O multiplexing (`select` and `poll`)
- signal driven I/O (`SIGIO`)
- asynchronous I/O (the POSIX `aio_` functions)

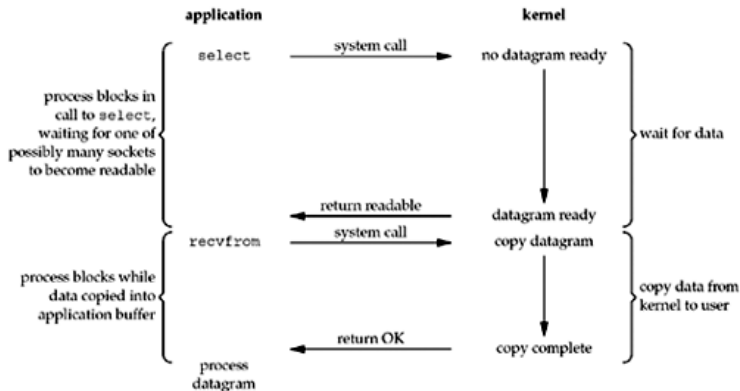
Blocking I/O Model



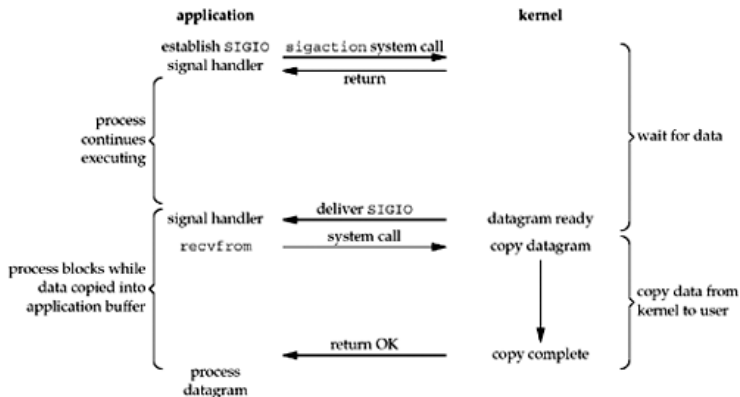
Nonblocking I/O Model



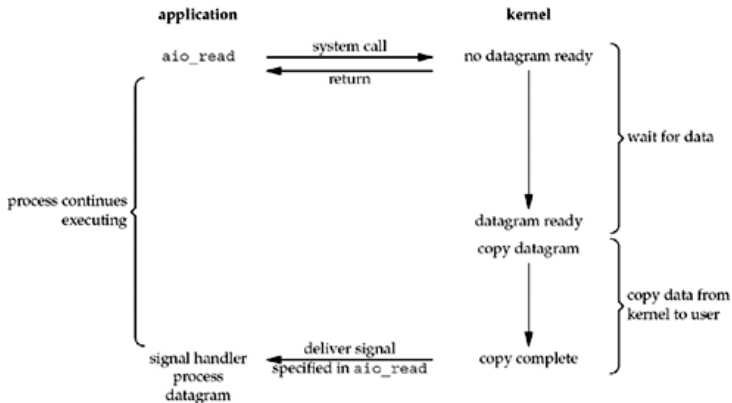
I/O Multiplexing Model



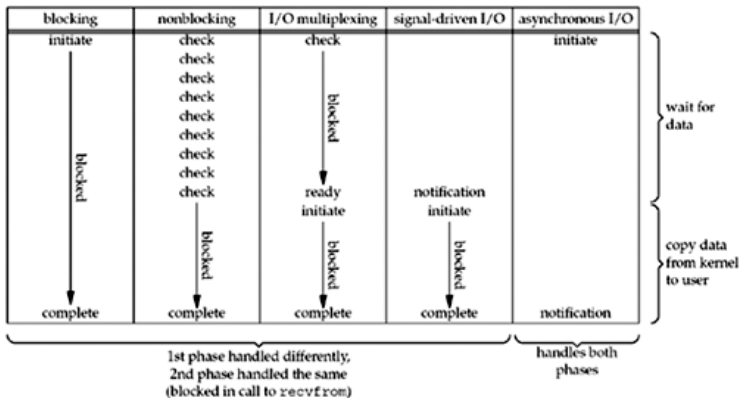
Signal driven I/O Model



Asynchronous I/O Model



Comparison of I/O Models



select Function

- Allows the process to instruct the kernel to wait for any one of multiple events to occur and to wake up the process only when one or more of these events occurs or when a specified amount of time has passed.
- (readable, writable, expired time)

```
#include <sys/select.h>
#include <sys/time.h>
```

```
int select(int maxfdpl, fd_set *readset, fd_set *writeset,
           fd_set *exceptset,
           const struct timeval *timeout);
```

Descriptor sets

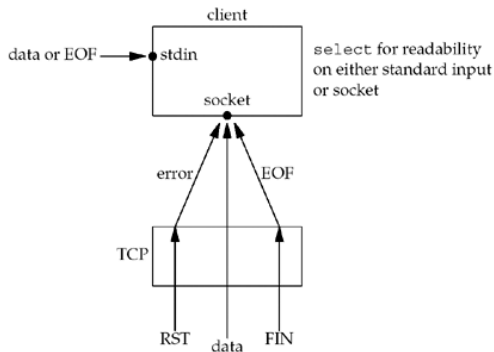
- Array of integers: each bit in each integer correspond to a descriptor.
- `fd_set`: an array of integers, with each bit in each integer corresponding to a descriptor.

```
/* clear all bits in fdset */  
Void FD_ZERO(fd_set *fdset);  
/* turn on the bit for fd in fdset */  
Void FD_SET(int fd, fd_set *fdset);  
/* turn off the bit for fd in fdset*/  
Void FD_CLR(int fd, fd_set *fdset);  
/* is the bit for fd on in fdset ? */  
int FD_ISSET(int fd, fd_set *fdset);
```

Conditions Summary

Condition	Readable?	Writable?	Exception?
Data to read	•		
Read half of the connection closed	•		
New connection ready for listening socket	•		
Space available for writing		•	
Write half of the connection closed		•	
Pending error	•	•	
TCP out-of-band data			•

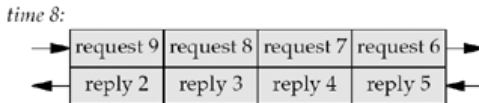
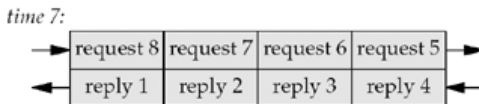
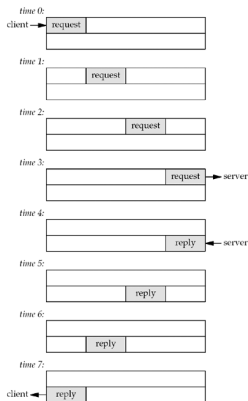
str_cli Function (Revisited)



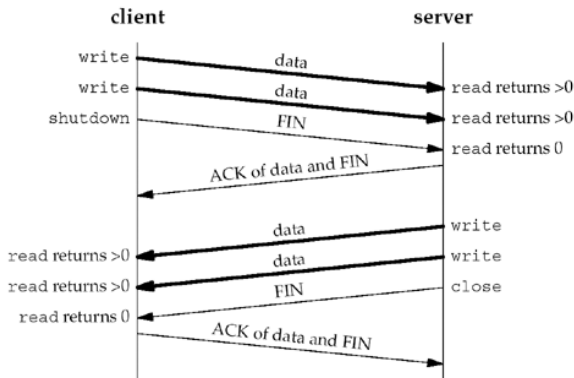
Conditions handled by select in `str_cli`.

Batch Input and Buffering

- Network pipe is not fully used



shutdown Function



pselect Function

- pselect function was invented by Posix.1g.

```
#include <sys/select.h>
#include <signal.h>
#include <time.h>

int pselect(int maxfdpl, fd_set *readset, fd_set *writeset,
            fd_set *exceptset, const struct timespec *timeout,
            const sigset_t *sigmask)
```

poll Function

- Similar to select, but provide additional information when dealing with streams devices

```
#include <poll.h>
int poll(struct pollfd *fdarray, unsigned long nfds, int timeout);
/*return : count of ready descriptors, 0 on timeout, -1 on error*/
```