Network Programming

Lecture 4—Elementary Sockets III: Socket Options, UDP Sockets, Name and Address Conversions

Lei Wang
lei.wang@dlut.edu.cn

Dalian University of Technology

Dec 8, 2008

Part 2. Elementary Sockets III: Socket Options, Name and Address Conversions

1. Socket Options: `getsockopt` and `setsockopt` function
    - Introduction
    - `getsockopt` and `setsockopt` Functions
    - `checkopts.c`
    - `fcntl` Function
2. UDP Sockets Introduciton
    - UDP Sockets Introduciton
    - `recvfrom` and `sendto` Functions
    - A UDP Example
3. Name and Address Conversions
    - DNS
    - `gethostbyname` and `gethostbyaddr` Functions
    - `getservbyname` and `getservbyport` Function
    - `getaddrinfo` Function

Introduction

There are three ways to get and set the options that affect a socket:

- The `getsockopt` and `setsockopt` functions
- The `fcntl` function
- The `ioctl` function (chapter 17)

getsockopt and setsockopt Functions

```
#include <sys/socket.h>

int getsockopt(int sockfd, int level, int optname,
               void *optval, socklen_t *optlen);

int setsockopt(int sockfd, int level, int optname,
               const void *optval socklen_t optlen);
```

Both return: 0 if OK, -1 on error

Summary of Socket Options

You may refer back to them later—not required at this time.

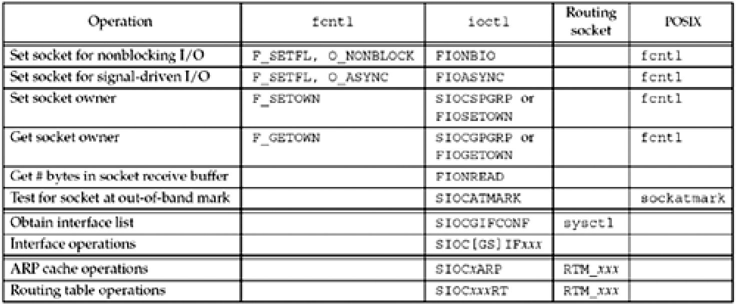
- Figure 7.1
- Figure 7.2

checkopts.c—experiment assignment

sockopt/checkopts.c

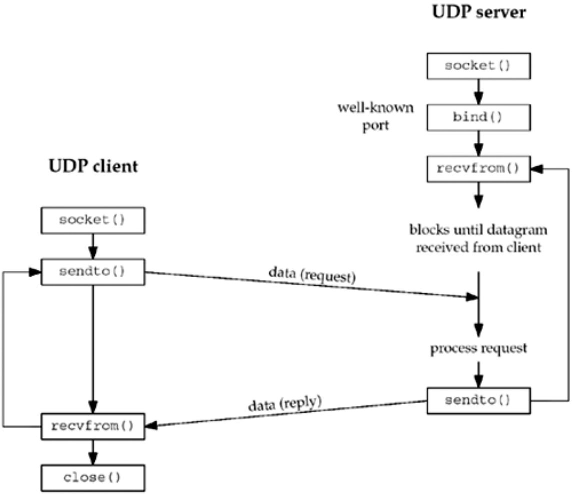
fcntl Function

fcntl: stands for "file control" and performs various descriptor control operations.



Operation	fcntl	ioctl	Routing socket	POSIX
Set socket for nonblocking I/O	F_SETFL, O_NONBLOCK	FIONBIO		fcntl
Set socket for signal-driven I/O	F_SETFL, O_ASYNC	FIOASYNC		fcntl
Set socket owner	F_SETOWN	SIOCSPGRP or FIOSETOWN		fcntl
Get socket owner	F_GETOWN	SIOCGPGRP or FIOGETOWN		fcntl
Get # bytes in socket receive buffer		FIONREAD		
Test for socket at out-of-band mark		SIOCATMARK		sockatmark
Obtain interface list		SIOCGIFCONF	sysctl	
Interface operations		SIOC[GS]IFxxx		
ARP cache operations		SIOCxARP	RTM_xxx	
Routing table operations		SIOCxxxRT	RTM_xxx	

UDP Sockets Introduciton

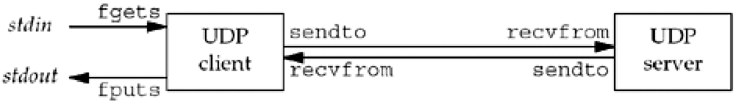


recvfrom and sendto Functions

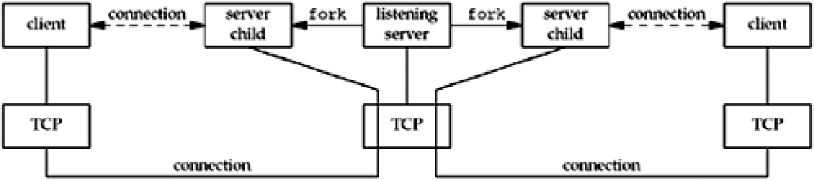
```
#include <sys/socket.h>

ssize_t recvfrom(int sockfd, void *buff, size_t nbytes,
                 int flags, struct sockaddr *from,
                 socklen_t *addrlen);

ssize_t sendto(int sockfd, const void *buff, size_t nbytes,
               int flags, const struct sockaddr *to,
               socklen_t addrlen);
```

Both return: number of bytes read or written if OK, ¨C1 on error

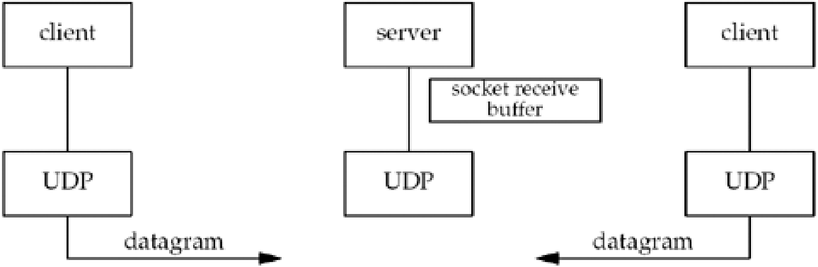
UDP Echo Example: UDP Echo Server



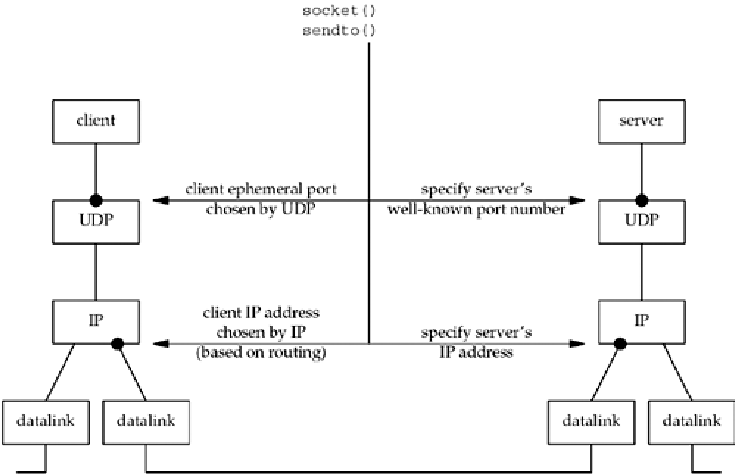
UDP Echo Example: TCP example revisited (2 clients)



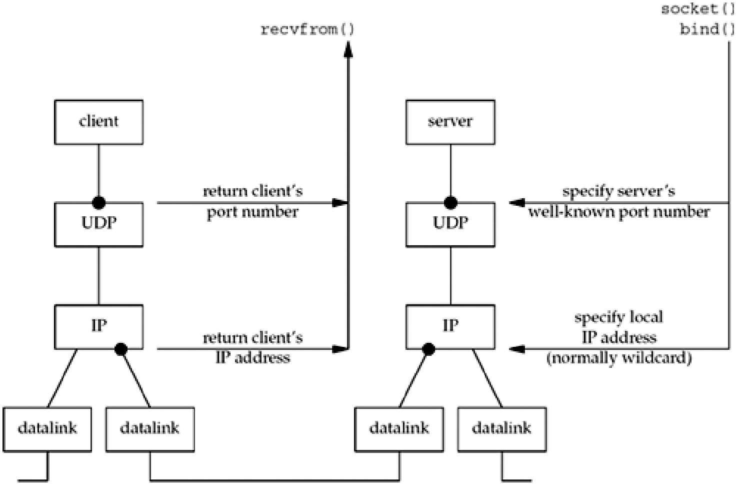
UDP Echo Example: UDP (2 clients)



UDP Echo Example: Summary (from client's perspective)



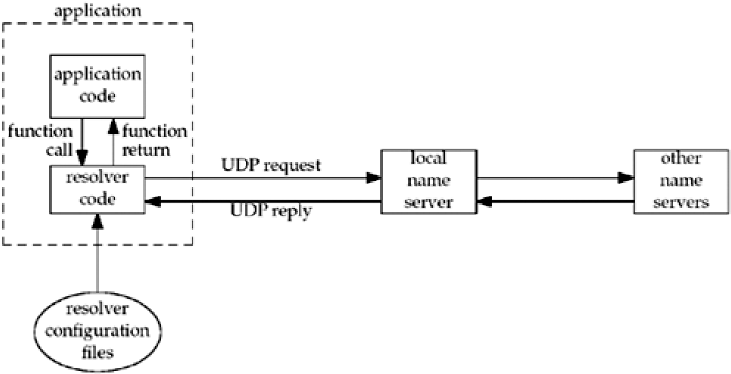
UDP Echo Example: Summary (from server's perspective)


recvfrom()
socket()
bind()
client
server
return client's port number
specify server's well-known port number
UDP
UDP
IP
IP
return client's IP address
specify local IP address (normally wildcard)
datalink
datalink
datalink
datalink

Name and Address Conversions

- Domain Name System (DNS)
- `gethostbyname` and `gethostbyaddr` Functions
- `getservbyname` and `getservbyport` Functions
- `getaddrinfo` Function

Resolvers and Name Servers



gethostbyname Function

```
#include <netdb.h>

struct hostent *gethostbyname (const char *hostname);
```

Returns: non-null pointer if OK, `NULL` on error with `h_errno` set

```
struct hostent {
  char   *h_name;       /* official (canonical) name of host */
  char  **h_aliases;    /* pointer to array of pointers to alias names*/
  int     h_addrtype;   /* host address type: AF_INET */
  int     h_length;     /* length of address: 4 */
  char  **h_addr_list; /* ptr to array of ptrs with IPv4 addrs */
};
```

Example: `names/hostent.c`

gethostbyaddr Function

```
#include <netdb.h>

struct hostent *gethostbyaddr (const char *addr, socklen_t len, int fa
```

Returns: non-null pointer if OK, NULL on error with h_errno set

Example: names/hostent.c

getservbyname Function

```
#include <netdb.h>

struct servent *
    getservbyname (const char *servname, const char *protoname);
```

Returns: non-null pointer if OK, NULL on error

```
struct servent {
  char   *s_name;       /* official service name */
  char  **s_aliases;    /* alias list */
  int     s-port;       /* port number, network-byte order */
  char   *s_proto;      /* protocol to use */
};
```

Examples:

```
struct servent *sptr;

sptr = getservbyname("domain", "udp"); /* DNS using UDP */
sptr = getservbyname("ftp", NULL);     /* FTP using TCP */
sptr = getservbyname("ftp", "udp");    /* this call will fail */
```

getaddrinfo Function

```
#include <netdb.h>

int getaddrinfo (const char *hostname, const char *service,
                 const struct addrinfo *hints,
                 struct addrinfo **result) ;
```

Returns: 0 if OK, nonzero on error (see Figure 11.7)

```
struct addrinfo {
   int           ai_flags;          /* AI_PASSIVE, AI_CANONNAME */
   int           ai_family;         /* AF_xxx */
   int           ai_socktype;       /* SOCK_xxx */
   int           ai_protocol;       /* 0 or IPPROTO_xxx for IPv4 and IPv6 */
   socklen_t     ai_addrlen;        /* length of ai_addr */
   char         *ai_canonname;      /* ptr to canonical name for host */
   struct sockaddr    *ai_addr;     /* ptr to socket address structure */
   struct addrinfo    *ai_next;     /* ptr to next structure in linked list */
};
```