

Advanced I/O Functions

- Socket timeouts
- recv and send functions
- readv and writev functions
- recvmsg and sendmsg functions
 - Ancillary data
 - Receiving ICMP error messages

Socket Timeouts

- Call alarm which generates SIGALRM before calling socket functions
- Block waiting for I/O in select which has a time limit built in
- Set SO_RCVTIMEO and SO_SNDFTIMEO socket options by setsockopt

recv and send Functions

~ read and write

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

```
ssize_t recv (int sockfd, void *buff, size_t nbytes, int flags);  
ssize_t send (int sockfd, const void *buff, size_t nbytes, int flags);
```

both return: number of bytes read or written if OK, -1 on error

| <i>flags</i> | Description | recv | send |
|---------------|------------------------------------|------|------|
| MSG_DONTROUTE | bypass routing table lookup | | X |
| MSG_DONTWAIT | only this operation is nonblocking | X | X |
| MSG_OOB | send or receive out-of-band data | X | X |
| MSG_PEEK | peek at incoming message | X | |
| MSG_WAITALL | wait for all the data | X | |

readv and writev functions

scatter read and gather write

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

```
ssize_t readv (int filedes, const struct iovec *iov, int iovcnt);
```

```
ssize_t writev (int filedes, const struct iovec *iov, int iovcnt);
```

both return: number of bytes read or written, -1 on error

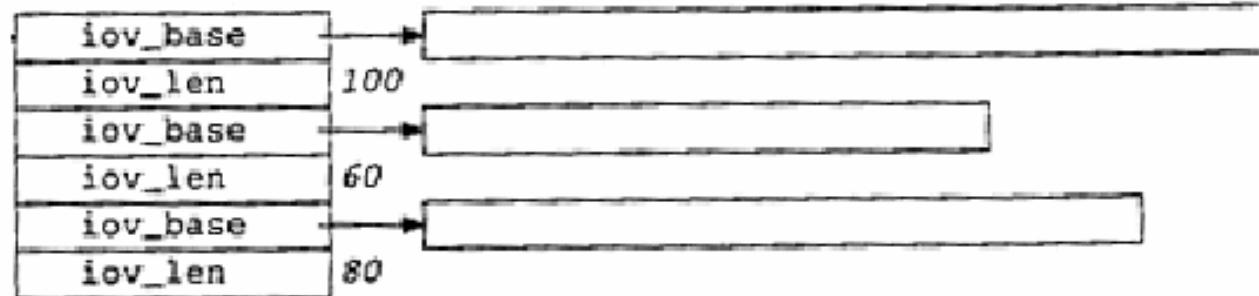
*iov: a pointer to an array of iovec structure

struct iovec {

void *iov_base; /* starting address of buffer */

size_t iov_len; /* size of buffer */

};



recvmsg and sendmsg Functions the most general socket I/O functions

```
#include <sys/socket.h>
ssize_t recvmsg (int sockfd, struct msghdr *msg, int flags);
ssize_t sendmsg (int sockfd, struct msghdr *msg, int flags);
```

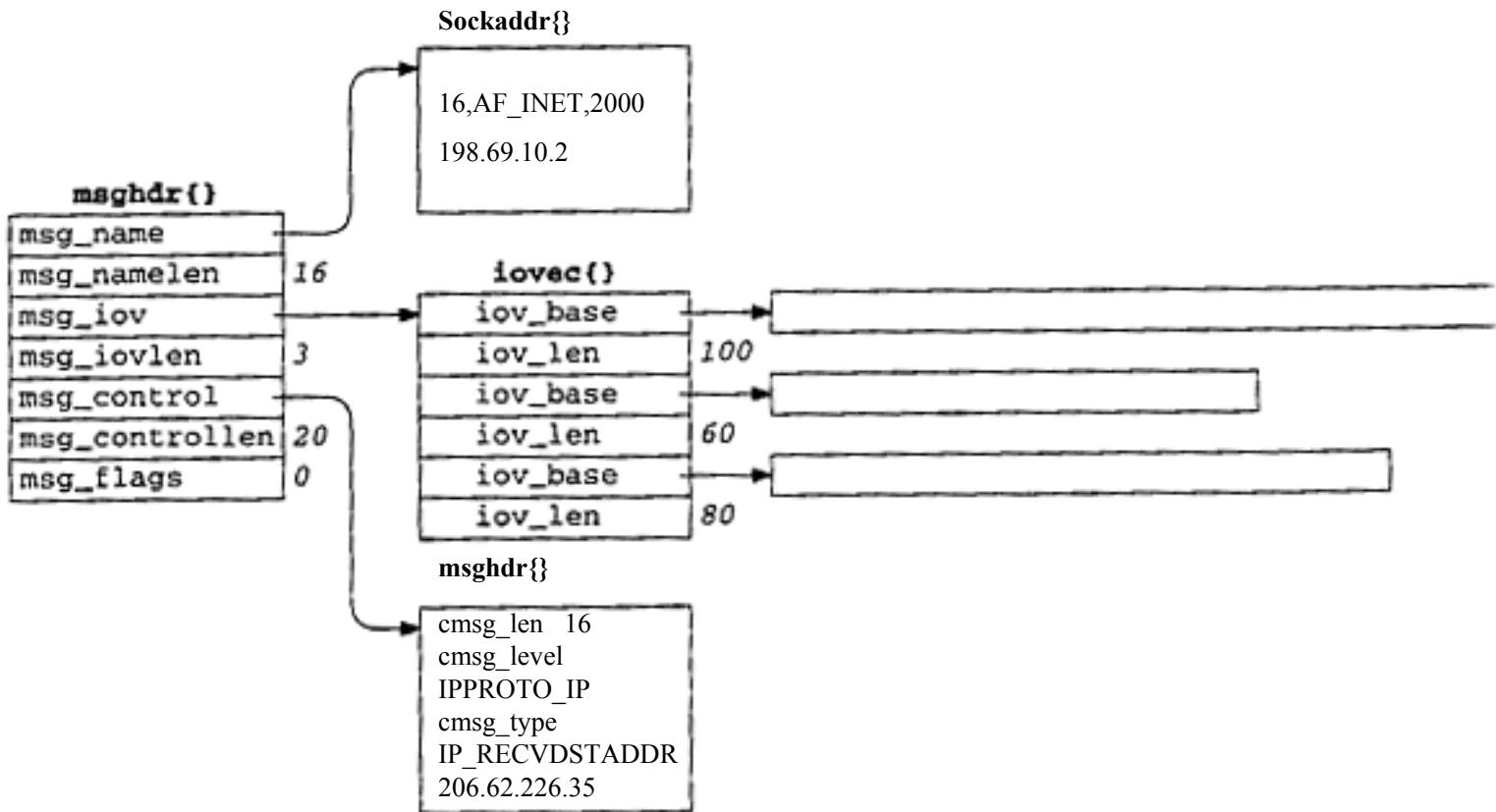
both return: number of bytes read or written if OK, -1 on error

```
struct msghdr {
    void    *msg_name;           /* protocol address */
    socklen_t msg_namelen;       /* size of protocol address */
    struct iovec  *msg iov;     /* scatter/gather array */
    size_t    msg iovlen;        /* # elements in msg iov */
    void    *msg_control;        /* ancillary data; must be aligned
                                    for a cmsghdr structure */
    socklen_t msg_controllen;    /* length of ancillary data */
    int     msg_flags;           /* flags returned by recvmsg ( ) */
};
```

Summary of I/O Flags by Various I/O Functions

| Flag | Examined by: <i>send flags</i> <i>sendto flags</i> <i>sendmsg flags</i> | Examined by: <i>recv flags</i> <i>recvfrom flags</i> <i>recvmsg flags</i> | Returned by: recvmsg msg_flags |
|---------------------|----------------------------------------------------------------------------------|------------------------------------------------------------------------------------|-----------------------------------|
| MSG_DONTROUTE | X | | |
| MSG_DONTWAIT | X | X | |
| MSG_PEEK | | X | |
| MSG_WAITALL | | X | |
| MSG_EOR | X | | X |
| MSG_OOB | X | X | X |
| MSG_BCAST | | | X |
| MSG_MCAST | | | X |
| MSG_TRUNC | | | X |
| MSG_CTRUNC | | | X |
| MSG_ERRQUEUE | | | X |

msghdr When recvmsg Returns



Comparison of Five Groups of I/O Functions

| Functions | Any descriptor | Only socket descriptor | Single buffer | Scatter buffer | Optional flags | Optional peer address | Optional control info |
|------------------|----------------|------------------------|---------------|----------------|----------------|-----------------------|-----------------------|
| read, write | x | | x | | | | |
| readv, writev | x | | | x | | | |
| recv, send | | x | x | | x | | |
| recvfrom, sendto | | x | x | | x | x | |
| recvmsg, sendmsg | | x | | x | x | x | x |

Ancillary Data (Control Info)

| Protocol | cmsg_level | cmsg_type | Description |
|-------------|--------------|----------------|------------------------------------|
| IPv4 | IPPROTO_IP | IP_RECVDSTADDR | receive dest addr with UDP data |
| | | IP_RECVIF | receive interface index with UDP |
| IPv6 | IPPROTO_IPV6 | IPV6_DSTOPTS | specify/receive dest options |
| | | IPV6_HOPLIMIT | specify/receive hop limit |
| | | IPV6_HOPOPTS | specify/receive hop-by-hop options |
| | | IPV6_NEXTHOP | specify next-hop addr |
| | | IPV6_PKTINFO | specify/receive packet info |
| | | IPV6_RTHDR | specify/receive routing header |
| Unix domain | SOL_SOCKET | SCM_RIGHTS | send/receive descriptors |
| | | SCM_CREDS | send/receive user credentials |

| | | | |
|------|--------------|------------|------------------------|
| IPv4 | IPPROTO_IP | IP_RECVERR | receive error messages |
| IPv6 | IPPROTO_IPV6 | IP_RECVERR | receive error messages |

extended error description

```
struct sock_extended_err{  
    __u32        ee_errno;  
    __u8         ee_origin;  
    __u8         ee_type;  
    __u8         ee_code;  
    __u8         ee_pad;  
    __u32        ee_info;  
    __u32        ee_data;  
};  
#define SO_EE_ORIGIN_NONE 0  
#define SO_EE_ORIGIN_LOCAL 1  
#define SO_EE_ORIGIN_ICMP  2  
#define SO_EE_ORIGIN_ICMP6 3  
#define SO_EE_ORIGIN_LOCAL_NOTIFY 4  
#define SO_EE_OFFENDER(ee) ((struct sockaddr*)((ee)+1))
```