

UDP SOCKETS — SERVER

```
int main(int argc, char *argv[])
{
    int sfd;
    struct sockaddr_saddr, caddr;

    sfd = socket(AF_INET, SOCK_DGRAM, 0);

    bzero(&saddr, sizeof(saddr));
    saddr.sin_family = AF_INET;
    saddr.sin_addr.s_addr = htonl(INADDR_ANY);
    saddr.sin_port = htons(SERV_PORT);

    bind(sfd, &saddr, sizeof(saddr));

    dg_echo(sfd, &caddr, sizeof(caddr));
}
```

SERVER ...

```
void dg_echo(int sfd, sockaddr *caddr, int clen)
{
    int n, len;
    char msg[MAXLINE];

    for ( ; ; ) {
        len = clen;
        n = recvfrom(sfd, msg, MAXLINE, 0,
                    caddr, &len);
        sendto(sfd, msg, n, 0, caddr, len);
    }
}
```

CLIENT

```
int main(int argc, char *argv[])
{
    int sfd;
    struct sockaddr saddr;

    sfd = socket(AF_INET, SOCK_DGRAM, 0);

    bzero(&saddr, sizeof(saddr));
    saddr.sin_family = AF_INET;
    saddr.sin_port = htons(SERV_PORT);
    inet_pton(AF_INET, argv[1], &saddr,
              &saddr.sin_addr);

    dg_client(stdin, sfd, &saddr, sizeof(saddr));
}
```

CLIENT ...

```
void dg_client(FILE *fp, int sfd,  
              sockaddr *saddr, int slen)  
{  
    int n;  
    char sendline[MAXLINE], recvline[MAXLINE];  
  
    while (fgets(sendline, MAXLINE, fp) != NULL) {  
        sendto(sfd, sendline, strlen(sendline),  
              0, saddr, slen);  
        n = recvfrom(sfd, recvline, MAXLINE, 0,  
                    NULL, NULL);  
        recvline[n] = 0;  
        fputs(recvline, stdout);  
    }  
}
```