

• A Simple Stream Server & Client

- The server sends the string **"Hello, World!\n"** out over a stream connection.

- You can test this server: run it in one machine, and telnet to it from another with:

- `$ telnet ServerName 3490`

- where *ServerName* is the name of the machine on which you're running the server.

- Or use our client

• The server code (Skeleton):

• #include

• #define MYPORT 3490 /* the port
clients will be connecting to */

• #define BACKLOG 10 /* how
many pending connections queue I
will hold */

• **main()**

• {

• int **sockfd, new_sockfd;**
/* listen on **sock_fd**, new connection
on **new_sockfd** */

• struct sockaddr_in **my_addr;**
/* my address information */

• struct sockaddr_in **their_addr;**
/* Client's address information */

• int **sin_size, pid;**

```
• sockfd = socket(AF_INET,  
                     SOCK_STREAM, 0) ;  
  
• my_addr.sin_family = AF_INET;  
  
• my_addr.sin_port = MYPORT;  
  
  my_addr.sin_addr.s_addr =  
                     INADDR_ANY;  
  /* auto-fill with my IP */  
  
• bind(sockfd, &my_addr,  
      sizeof(struct sockaddr));  
  
• listen(sockfd, BACKLOG)
```

```
• while(1) { /* main accept() loop */
• new_sockfd = accept(sockfd,
                        &their_addr);
  }
  printf("server: got connection from
%s\n", their_addr.sin_addr);

  pid = fork();

• if ( pid == 0) { /* this is the child
process */
• send(new_sockfd, "Hello, world!\n",
      14,0)
• close(new_sockfd);
• exit(0);/* End Child Process */
• }
• close(new_sockfd); /* parent doesn't
need this */
• }
```

• A Simple, Skeleton Stream Client

- This client connects to the host you specify on the command line, port 3490.
- It gets the string that the server sends and prints it on the screen.
- usage: client *hostname*

• The client code (skeleton):

- #include <stdio.h>

- #include

```
#define PORT 3490  /* the port I will  
                    be connecting to */
```

- #define MAXDATASIZE 100 /* Max
 number of bytes we can get at once */

- int **main**(int argc, char *argv[])

- {

- int **sockfd**, numbytes;

- char **buf**[MAXDATASIZE];

- struct hostent ***he**;

- struct sockaddr_in **their_addr**;

```
    /* Server's address information */
```

- **he = gethostbyname**(argv[1])

- **sockfd = socket(AF_INET, SOCK_STREAM, 0)**
- **their_addr.sin_family = AF_INET;**
- **their_addr.sin_port = PORT;**
- **their_addr.sin_addr = he->h_addr);**
- **connect(sockfd, &their_addr, sizeof(struct sockaddr));**
- **numbytes = recv(sockfd, buf, MAXDATASIZE, 0);**
- **buf[numbytes] = '\0';**
- **printf("Received: %s", buf);**
- **close(sockfd);**
- **return 0;**