

File and Network I/O

CS 113: Introduction to C

Instructor: Saikat Guha

Cornell University

Fall 2006, Lecture 10

Input and Output

- ▶ Keyboard I/O
- ▶ Disk I/O
- ▶ Network I/O

... Unix Posix 101 ...

User Input: Keyboard

Reading till end of line

```
char buf[128];  
fgets(buf, 128, stdin);
```

Reading formatted input

```
int i, j;  
char buf[128];  
scanf("%d %d %s", &i, &j, buf);
```

File I/O

Opening and closing files

```
int fd;           // File Descriptor
fd = open("/path/to/file", O_RDWR | O_CREAT);
close(fd);
```

Reading and Writing

```
char buf[4096]; int len;
len = read(fd, buf, 4096)
len = write(fd, buf, 4096);
```

WARNING: Size passed is only a **suggestion**. May read/write fewer than requested number of bytes. Return value is number of bytes actually read/written. **MUST** retry if not fully read/written.

File I/O

- ▶ `lseek(fd, numbytes, SEEK_CUR);`
Seek numbytes from from current location.
- ▶ `sync();`
Ensure bytes hit the disk. Not needed for the most part.
- ▶ `FILE *ffd = fdopen(fd, "r");`
Construct a stream from file-descriptor.
- ▶ `fprintf(ffd, "format", args);`
Write formatted text output to file.
- ▶ `fscanf(ffd, "format", args);`
Read formatted input from the file.
- ▶ `fclose(ffd);`
Close a stream.

Opening and closing network sockets

```
int sock;                // File Descriptor
sock = socket(PF_INET, SOCK_STREAM, IPPROTO_TCP);
close(sock);
```

Internet Addresses

```
struct sockaddr_in addr;
addr.sin_family = AF_INET;
addr.sin_addr.s_addr = ntohl(0x7F000001);
addr.sin_port = ntohs(8080);
```

Network I/O

Server

```
srv = socket(PF_INET, SOCK_STREAM, IPPROTO_TCP);  
err = bind(srv, (struct sockaddr *)&addr, sizeof(addr));  
if (err) ...  
err = listen(srv, 5);  
if (err) ...  
cli = accept(srv, NULL, 0);
```

Client

```
cli = socket(PF_INET, SOCK_STREAM, IPPROTO_TCP);  
err = connect(cli, (struct sockaddr *)&addr, sizeof(addr));  
if (err) ...
```

Read/Write data just as you would with file-descriptors.