

CSE 451 Sectional

08 April, 2015

Interprocess Communications

Process to Process

Parent Process to Child Process

Uni-directional? Bi-directional? What are the data models?

Lets start from the Shell

```
/bin/ls -al / | /usr/bin/tr a-z A-Z | /usr/bin/head
```

Pipes are unidirectional FIFO streaming

```
sort < file_list.txt > sorted_file_list.txt
```

IO redirection streams file data to programs.

Programs can move data between themselves and the system (as launched from the shell)

Named Pipes

```
mkfifo fifo0 fifo1
```

```
prog1 > fifo0
```

```
prog2 < fifo0 > fifo1
```

```
cat < fifo1
```

```
==
```

```
prog1 | prog2 | cat
```

Named Pipes look like files with piping behavior, and persist

Programatic Pipes

```
int main(void) {
    int    fd[2], nbytes;
    pid_t  childpid;
    char   string[] = "Hello, world!\n";
    char   readbuffer[80];

    pipe(fd);

    if((childpid = fork()) == -1) {
        perror("fork"); exit(1);
    }

    if(childpid == 0) {
        close(fd[0]); /* Child process closes up input side of pipe */
        write(fd[1], string, (strlen(string)+1)); /* Send "string" through the output side of pipe */
        exit(0);
    }
    else {
        close(fd[1]); /* Parent process closes up output side of pipe */
        nbytes = read(fd[0], readbuffer, sizeof(readbuffer)); /* Read in a string from the pipe */
        printf("Received string: %s", readbuffer);
    }
    return(0);
}
```

Sockets

Unix Domain Sockets, UDP, TCP

Can be Bi-directional

More similar to a file on disk than pipes

TCP socket (Blocking Example)

```
#!/usr/bin/env python
import socket

TCP_IP = '127.0.0.1'
TCP_PORT = 5005
BUFFER_SIZE = 256

s = socket.socket(socket.AF_INET,
socket.SOCK_STREAM)
s.bind((TCP_IP, TCP_PORT))
s.listen(1)

conn, addr = s.accept()
print 'Connection address:', addr
while 1:
    data = conn.recv(BUFFER_SIZE)
    if not data: break
    print "received data:", data
    conn.send(data) # echo
conn.close()
```

```
#!/usr/bin/env python

import socket

TCP_IP = '127.0.0.1'
TCP_PORT = 5005
BUFFER_SIZE = 1024
MESSAGE = "Hello, World!"

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((TCP_IP, TCP_PORT))
s.send(MESSAGE)
data = s.recv(BUFFER_SIZE)
s.close()

print "received data:", data
```