CS 33

More Network Programming
Stream Relay

Source → Relay → Sink
Source → Relay → Sink
Sink → Relay → Source
Sink → Relay → Source
Solution?

```c
while (...) {
    size = read(left, buf, sizeof(buf));
    write(right, buf, size);
    size = read(right, buf, sizeof(buf));
    write(left, buf, size);
}
```
Select System Call

```c
int select(
    int nfds,      // size of fd_sets
    fd_set *readfds, // descriptors of interest
                  // for reading
    fd_set *writefds, // descriptors of interest
                   // for writing
    fd_set *excpfds, // descriptors of interest
                   // for exceptional events
    struct timeval *timeout
                  // max time to wait
);
```
Relay Sketch

```c
void relay(int left, int right) {
    fd_set rd, wr;
    int maxFD = max(left, right) + 1;
    FD_ZERO(&rd); FD_SET(left, &rd); FD_SET(right, &rd);
    FD_ZERO(&wr); FD_SET(left, &wr); FD_SET(right, &wr);
    while (1) {
        select(maxFD, &rd, &wr, 0, 0);
        if (FD_ISSET(left, &rd))
            read(left, bufLR, BSIZE);
        if (FD_ISSET(right, &rd))
            read(right, bufRL, BSIZE);
        if (FD_ISSET(right, &wr))
            write(right, bufLR, BSIZE);
        if (FD_ISSET(left, &rd))
            write(left, bufRL, BSIZE);
    }
}
```
Quiz 1

40 bytes have been read from the left-hand source. Select reports that it is ok to write to the right-hand sink.

a) You’re guaranteed you can immediately write all 40 bytes to the right-hand sink
b) All that’s guaranteed is that you can immediately write at least one byte to the right-hand sink
c) Nothing is guaranteed
void relay(int left, int right) {
    fd_set rd, wr;
    int left_read = 1, right_write = 0;
    int right_read = 1, left_write = 0;
    int sizeLR, sizeRL, wret;
    char bufLR[BSIZE], bufRL[BSIZE];
    char *bufpR, *bufpL;
    int maxFD = max(left, right) + 1;

    // set up file descriptors so they won’t
    // wait if I/O is not yet possible
    fcntl(left, F_SETFL, O_NONBLOCK);
    fcntl(right, F_SETFL, O_NONBLOCK);
while(1) {
    FD_ZERO(&rd);
    FD_ZERO(&wr);
    if (left_read)
        FD_SET(left, &rd);
    if (right_read)
        FD_SET(right, &rd);
    if (left_write)
        FD_SET(left, &wr);
    if (right_write)
        FD_SET(right, &wr);

    select(maxFD, &rd, &wr, 0, 0);
}
Relay (3)

```c
if (FD_ISSET(left, &rd)) {
    sizeLR = read(left, bufLR, BSIZE);
    left_read = 0;
    right_write = 1;
    bufpR = bufLR;
}
if (FD_ISSET(right, &rd)) {
    sizeRL = read(right, bufRL, BSIZE);
    right_read = 0;
    left_write = 1;
    bufpL = bufRL;
}
```
Relay (4)

if (FD_ISSET(right, &wr)) {
    if ((wret = write(right, bufpR, sizeLR)) == sizeLR) {
        left_read = 1; right_write = 0;
    } else {
        sizeLR -= wret; bufpR += wret;
    }
}

if (FD_ISSET(left, &wr)) {
    if ((wret = write(left, bufpL, sizeRL)) == sizeRL) {
        right_read = 1; left_write = 0;
    } else {
        sizeRL -= wret; bufpL += wret;
    }
}

return 0;