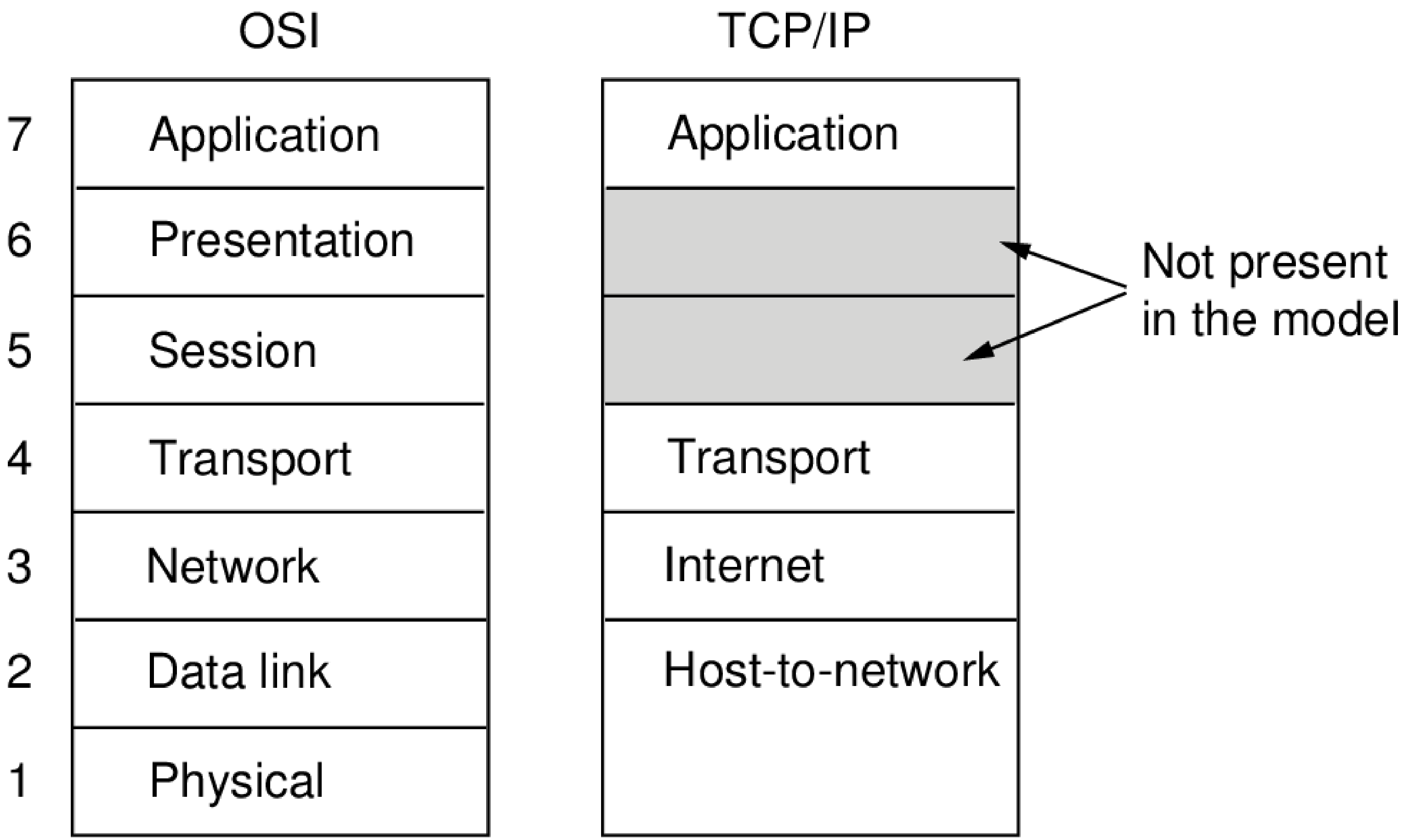
Lecture 3

Introduction to socket API

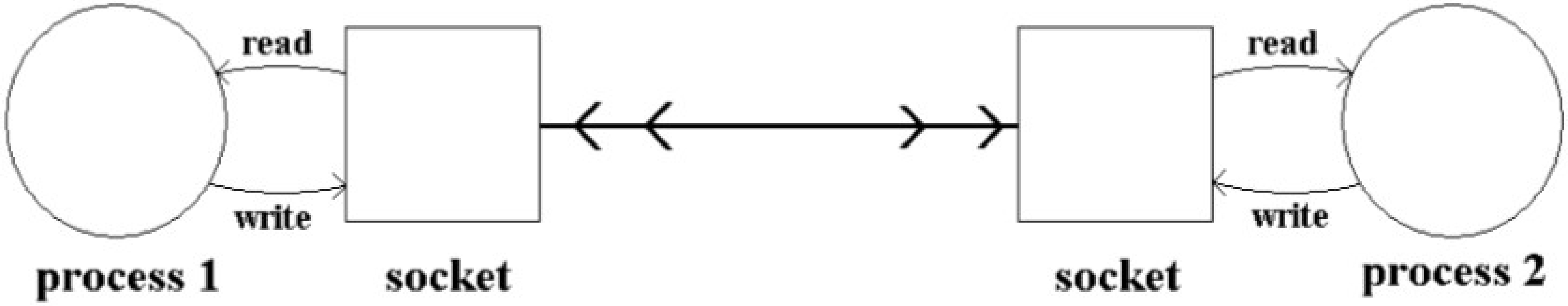
Review



	OSI	TCP/IP
7	Application	Application
6	Presentation	
5	Session	
4	Transport	Transport
3	Network	Internet
2	Data link	Host-to-network
1	Physical	

Not present in the model

Socket Idea in Transport Layer



Set Up an Connection

Server	Client
create socket	
bind to local address	
convert to listening socket	
loop	create socket
accept connection	request connection
get request	get reply
send response	send request
close connection	get response
end loop	close connection

System Calls

```
int connect(
    int socket, // file descriptor of a (local) socket

    struct sockaddr * saddr, // pointer to a structure containing the address of a remote socket

    int saddrlen // length (in bytes) of the sockaddr struct
);
```

System Calls

```
int socket(
     int protofamily, // PF_INET or PF_UNIX
     int type,//  SOCK_STREAM or SOCK_DGRAM
     int protocol//  IPPROTO_TCP or
  IPPROTO_UDP or 0 (use default)
);
```

Two Important structs

```
struct sockaddr {
    u_char   sa_len; // length of the struct

    u_char   sa_family; // address family

    char       sa_data[14];    // actual
  address; format depends on address family
};
```

Two Important structs

```
struct sockaddr_in {
     u_char sin_len; //  length of the struct

     u_char sin_family; //  address family
  (AF_INET)

     u_short sin_port; //  port number

     struct in_addr sin_addr; //  32-bit binary IP
  address

     char sin_zero[8]; //  set to zero
};
```

System Calls

```
int bind(
     int sockfd, //file descriptor for a local
  socket

     struct sockaddr * saddr,  //socket
  address

     int addrlen //length of sockaddr struct
);
```

System Calls

```
int listen(
    int sockfd,//file descriptor for a local socket
    int backlog//length of queue for waiting clients
);
```

System Calls

```
int accept(
    int sockfd, //file descriptor for a listening socket
    struct sockaddr * saddr, //address of local variable to be filled in with client's socket address
    int * addrlen //address of local variable to be filled in with length of client's socket address
);
```

System Calls

```
int read(
      int fd, //  file descriptor to read
      char * buffer, // address of a local variable to be
                 filled with incoming data
      int len //number of bytes to read
);


int write(
      int fd, //file descriptor to write
      char * buffer, //address of a local variable
                 containing data to be written
      int len //number of bytes to write
);
```

System Calls

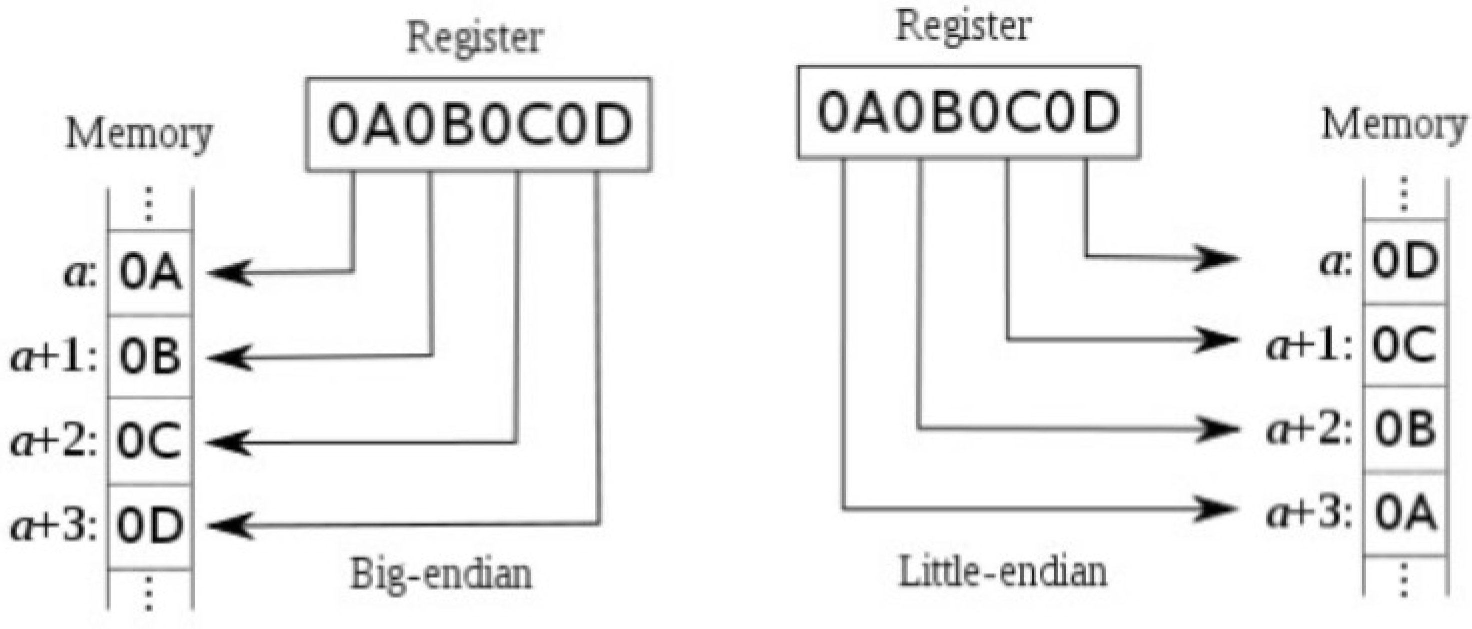
```
int send(int fd, char * buffer, int len, int
  flags);

int recv(int fd, char * buffer, int len, int
  flags);
```

System Calls

```
ssize_t sendto(int s, const void *buf, size_t
  len, int flags, const struct sockaddr
  *to,socklen_t tolen);
```

```
ssize_t recvfrom(int s, void *buf, size_t
 len, int flags,struct sockaddr *from,
 socklen_t *fromlen);
```

Network Byte Order


Register
0A0B0C0D
Memory
a: 0A
a+1: 0B
a+2: 0C
a+3: 0D
Big-endian
Register
0A0B0C0D
Memory
a: 0D
a+1: 0C
a+2: 0B
a+3: 0A
Little-endian