

CS 744

Lecture 14

- network & storage IO handling & usage
- synchronization & multi-threading
- networked appl^{ns}. / dist^r. systems

- lab 4 ← a ⑦
 b
 c

⑧ Project (Lab 5++)

~ Lab exam 2 (?)

— Endsem exam

the networking API / interface

⑩ OS is a glorified paper weight unless,

(i) it supports user processes & the sys call interface

(ii) supports primitives / abstraction for IO

⑩ communication primitive via networked
interconnected
computers / machines
for communication
between processes
or process related abstractions / devices.

— KBD, mouse,
monitor
— disk
— n/w
printer

Socket abstraction of the OS facilitates / sets up
the communication.

— similar to a pipe

— naming / setup process & systems calls different.

⑩ what is a socket?

Q. what is a socket?

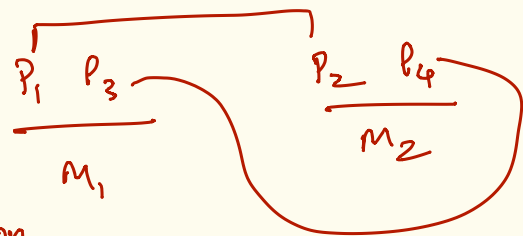
- └ endpoint of a process. (process create endpoints.
- └ pair of sockets communicate w/ each other.
- └ P1 opens a socket (server)
- └ P2 opens a socket (client)
- └ connection establishment for sockets to send/recv.

types of sockets

(i) ~ local vs on network

unix domain
sockets
(via a file)

internet domain sockets
(IP address, port number)



(ii) connection-oriented vs connection-less
(TCP) (UDP)

- after connection is established, send/recv msg's w/ sockfd reference.

- in-order, reliable delivery.

- each msg. has address details of (socket) other endpoint.

- no-order guarantees
- unreliable.

system call

sockfd = socket (domain, type, protocol)

(int) index
in fd table

PF_LOCAL
PF_INET/INET6

SOCK_STREAM
SOCK_DGRAM

0 < TCP
UDP

⑧ typical sequence of system calls.

connection
oriented

Server process

client process

```
sockfd = socket();
```

port
number

```
clientfd = socket ( );
```

bind (sockfd, serv_addr details)

connect (clientId, server
details,
...)

```
listen(sockfd, backlog);
```

#outstanding
connection
requests

```
newfd, = accept(sockfd, sock_addr, ...)
```

filled up
with socket
details of
connecting socket

sock $\nearrow$
fd that refers
to the connection

<serverip, server port

client ip, client port

~~not~~ same
as that
on bind.

any port
as long as
it is unique

A red spiral graphic, starting from a small circle in the center and winding outwards in a clockwise direction.

action/communication

```
read(newfd, buf, size);
```

write (new fd, buf, size);

```
close(newfd1);
```

```
write(clientfd, ...)
```

```
read(clientfd, ...)
```

```
close(clientfd);
```

client fd

<4-tuple> clients

new server fd,

 <4-tuple>
 ↘ server fd

 N/w

sip Sport Cip Sport
payload

data

data