

IPC and Networking

The background of the slide features a photograph of a large, classical-style statue, likely the 'Alma Mater' statue at the University of Illinois. The statue is positioned in the center-right of the frame. The entire image is overlaid with a semi-transparent orange color, which serves as a background for the white text.

CS 240 - The University of Illinois

Wade Fagen-Ulmschneider

September 28, 2021

Threads vs. Processes

A photograph of a crowd of people gathered around a statue of a woman in a long dress, set against a background of bare trees. The entire image is overlaid with a semi-transparent orange filter. The text "Threads vs. Processes" is centered in white.

Synchronization Technique #1

- **Threads** within a process share nearly all resources (exceptions are few, like the PC and their stack frames).

Synchronization Technique #1

- **Threads** within a process share nearly all resources (exceptions are few, like the PC and their stack frames).

AND

- **Processes** are almost _____ from other processes.

	Threads	Processes
Creation		
Overhead		
Context Switching		
Virtual Memory		

Case Study: Chrome

IPC



Inter-Process Communication (IPC)

IPC is the broad terminology for all technologies that facilitate real-time communication between threads.

Inter-Process Communication (IPC)

1.

Pipes

```
$ ps -aux | grep waf
```

Pipes

```
int pipe(int pipefd[2]);
```

Inter-Process Communication (IPC)

2.

Shared Memory

```
void *mmap(void *addr, size_t length, int prot,  
int flags, int fd, off_t offset);
```

Inter-Process Communication (IPC)

3.

Signals

```
$ kill -TERM <pid>
```

Signals

```
$ kill -1
```


Signals

```
int kill(pid_t pid, int sig);
```

Signals

```
struct sigaction {  
    void (*sa_handler)(int);  
    ...  
}
```

Inter-Process Communication (IPC)

4.

Message Queues

```
mqd_t mq_open(const char *name, int oflag);
```

```
int mq_send(mqd_t mqdes, const char *msg_ptr,  
            size_t msg_len, unsigned int msg_prio);
```

```
ssize_t mq_receive(mqd_t mqdes, char *msg_ptr,  
                  size_t msg_len, unsigned int *msg_prio);
```

```
int mq_close(mqd_t mqdes);
```

Semaphores

5.

Sockets

6.

Sockets

```
int socket(int domain, int type, int protocol);
```

Sockets

```
int bind(int sockfd, const struct sockaddr *addr,  
         socklen_t addrlen);
```


Sockets

```
int connect(int sockfd,  
            const struct sockaddr *addr,  
            socklen_t addrlen);
```

Sockets

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

Sockets

```
int accept(int sockfd,  
           struct sockaddr *restrict addr,  
           socklen_t *restrict addrlen);
```

Networking

A photograph of a crowd of people gathered around a statue of Alma Mater, overlaid with a semi-transparent orange filter. The word "Networking" is written in large white text across the center of the image. The background shows a large group of people, some looking towards the camera and others looking towards the statue. The statue is a large, classical-style figure standing on a pedestal. The overall scene is outdoors, with trees and foliage visible in the background.

Networking

Q: What do we expect out of networking?

Hosts Routers Links Applications	Protocols Hardware Software Bit Errors	Packet Errors Link Failures Node Failures Message Delays	Out-of-Order Packets Eavesdropping ...and more...
---	---	---	--

We define common _____ -- a message format and rules for exchanging messages. You know many protocols already:

```
00 4500 00c6 1e1f 4000 4006 152e ac16 b24c
10 12dc 95a6 bafa 0050 0f60 c9b4 356a 523f
20 8018 01f6 079e 0000 0101 080a 8146 30a0
30 31d4 daac 4745 5420 2f20 4854 5450 2f31
40 2e31 0d0a 5573 6572 2d41 6765 6e74 3a20
50 5767 6574 2f31 2e32 302e 3320 286c 696e
60 7578 2d67 6e75 290d 0a41 6363 6570 743a
70 202a 2f2a 0d0a 4163 6365 7074 2d45 6e63
80 6f64 696e 673a 2069 6465 6e74 6974 790d
90 0a48 6f73 743a 2077 6166 2e63 732e 696c
a0 6c69 6e6f 6973 2e65 6475 0d0a 436f 6e6e
b0 6563 7469 6f6e 3a20 4b65 6570 2d41 6c69
c0 7665 0d0a 0d0a
```



```
00 4500 00c6 1e1f 4000 4006 152e ac16 b24c
10 12dc 95a6 bafa 0050 0f60 c9b4 356a 523f
20 8018 01f6 079e 0000 0101 080a 8146 30a0
30 31d4 daac 4745 5420 2f20 4854 5450 2f31
40 2e31 0d0a 5573 6572 2d41 6765 6e74 3a20
50 5767 6574 2f31 2e32 302e 3320 286c 696e
60 7578 2d67 6e75 290d 0a41 6363 6570 743a
70 202a 2f2a 0d0a 4163 6365 7074 2d45 6e63
80 6f64 696e 673a 2069 6465 6e74 6974 790d
90 0a48 6f73 743a 2077 6166 2e63 732e 696c
a0 6c69 6e6f 6973 2e65 6475 0d0a 436f 6e6e
b0 6563 7469 6f6e 3a20 4b65 6570 2d41 6c69
c0 7665 0d0a 0d0a
```

```
00 4500 00c6 1e1f 4000 4006 152e ac16 b24c
10 12dc 95a6 bafa 0050 0f60 c9b4 356a 523f
20 8018 01f6 079e 0000 0101 080a 8146 30a0
30 31d4 daac 4745 5420 2f20 4854 5450 2f31
40 2e31 0d0a 5573 6572 2d41 6765 6e74 3a20
50 5767 6574 2f31 2e32 302e 3320 286c 696e
60 7578 2d67 6e75 290d 0a41 6363 6570 743a
70 202a 2f2a 0d0a 4163 6365 7074 2d45 6e63
80 6f64 696e 673a 2069 6465 6e74 6974 790d
90 0a48 6f73 743a 2077 6166 2e63 732e 696c
a0 6c69 6e6f 6973 2e65 6475 0d0a 436f 6e6e
b0 6563 7469 6f6e 3a20 4b65 6570 2d41 6c69
c0 7665 0d0a 0d0a
```

```
00 4500 00c6 1e1f 4000 4006 152e ac16 b24c
10 12dc 95a6 bafa 0050 0f60 c9b4 356a 523f
20 8018 01f6 079e 0000 0101 080a 8146 30a0
30 31d4 daac 4745 5420 2f20 4854 5450 2f31
40 2e31 0d0a 5573 6572 2d41 6765 6e74 3a20
50 5767 6574 2f31 2e32 302e 3320 286c 696e
60 7578 2d67 6e75 290d 0a41 6363 6570 743a
70 202a 2f2a 0d0a 4163 6365 7074 2d45 6e63
80 6f64 696e 673a 2069 6465 6e74 6974 790d
90 0a48 6f73 743a 2077 6166 2e63 732e 696c
a0 6c69 6e6f 6973 2e65 6475 0d0a 436f 6e6e
b0 6563 7469 6f6e 3a20 4b65 6570 2d41 6c69
c0 7665 0d0a 0d0a
```