

SER 321 C Session

SI Session

Thursday, June 13th 2024

6:00 pm - 7:00 pm MST

Agenda



OSI Model Review

Threads!

Threading Pitfalls

Dining Philosophers

Concurrency Structures

Threading your Code

SI Session Expectations

Thanks for coming to the **SER 321** SI session. We have a packed agenda and we are going to try to get through as many of our planned example problems as possible. This session will be recorded and shared with others.

- If after this you want to see additional examples, please visit the drop-in tutoring center.
- We will post the link in the chat now and at the end of the session.
 - tutoring.asu.edu
- Please keep in mind we are recording this session and it will be made available for you to review 24-48 hours after this session concludes.
- Finally, please be respectful to each other during the session.

Interact with us:

Zoom Features



Zoom Chat

- Use the chat feature to interact with the presenter and respond to presenter's questions.
- Annotations are encouraged

SER 321

OSI Model

TCP/UDP

Data	Application	
Data	Presentation	
Data	Session	
Segment	Transport	
Packet	Network	
Frame	Data Link	
Bits	Physical	

SER 321

OSI Model

HTTP(s), SMTP, FTP, IMAP,
POP, etc.

Data	Application	
Data	Presentation	
Data	Session	
Segment	Transport	TCP/UDP
Packet	Network	
Frame	Data Link	
Bits	Physical	

SER 321

OSI Model

LLC, MAC, data transmission
in LAN

Data	Application	HTTP(s), SMTP, FTP, IMAP, POP, etc.
Data	Presentation	
Data	Session	
Segment	Transport	TCP/UDP
Packet	Network	
Frame	Data Link	
Bits	Physical	

SER 321

OSI Model

Signal, Binary transmission

Data	Application	HTTP(s), SMTP, FTP, IMAP, POP, etc.
Data	Presentation	
Data	Session	
Segment	Transport	TCP/UDP
Packet	Network	
Frame	Data Link	LLC, MAC, data transmission in LAN
Bits	Physical	

SER 321

OSI Model

AuthN, authZ, session mgmt

Data	Application	HTTP(s), SMTP, FTP, IMAP, POP, etc.
Data	Presentation	
Data	Session	
Segment	Transport	TCP/UDP
Packet	Network	
Frame	Data Link	LLC, MAC, data transmission in LAN
Bits	Physical	Signal, Binary transmission

SER 321

OSI Model

IP address, routing and
delivery

Data	Application	HTTP(s), SMTP, FTP, IMAP, POP, etc.
Data	Presentation	
Data	Session	AuthN, authZ, session mgmt
Segment	Transport	TCP/UDP
Packet	Network	
Frame	Data Link	LLC, MAC, data transmission in LAN
Bits	Physical	Signal, Binary transmission

SER 321

OSI Model

Translation, compression,
encryption

Data	Application	HTTP(s), SMTP, FTP, IMAP, POP, etc.
Data	Presentation	
Data	Session	AuthN, authZ, session mgmt
Segment	Transport	TCP/UDP
Packet	Network	IP address, routing and delivery
Frame	Data Link	LLC, MAC, data transmission in LAN
Bits	Physical	Signal, Binary transmission

SER 321

OSI Model

Data	Application	HTTP(s), SMTP, FTP, IMAP, POP, etc.
Data	Presentation	Translation, compression, encryption
Data	Session	AuthN, authZ, session mgmt
Segment	Transport	TCP/UDP
Packet	Network	IP address, routing and delivery
Frame	Data Link	LLC, MAC, data transmission in LAN
Bits	Physical	Signal, Binary transmission

SER 321

OSI Model

Data	Application	HTTP(s), SMTP, FTP, IMAP, POP, etc.
Data	Presentation	Translation, compression, encryption
Data	Session	AuthN, authZ, session mgmt
Segment	Transport	TCP/UDP
Packet	Network	IP address, routing and delivery
Frame	Data Link	LLC, MAC, data transmission in LAN
Bits	Physical	Signal, Binary transmission

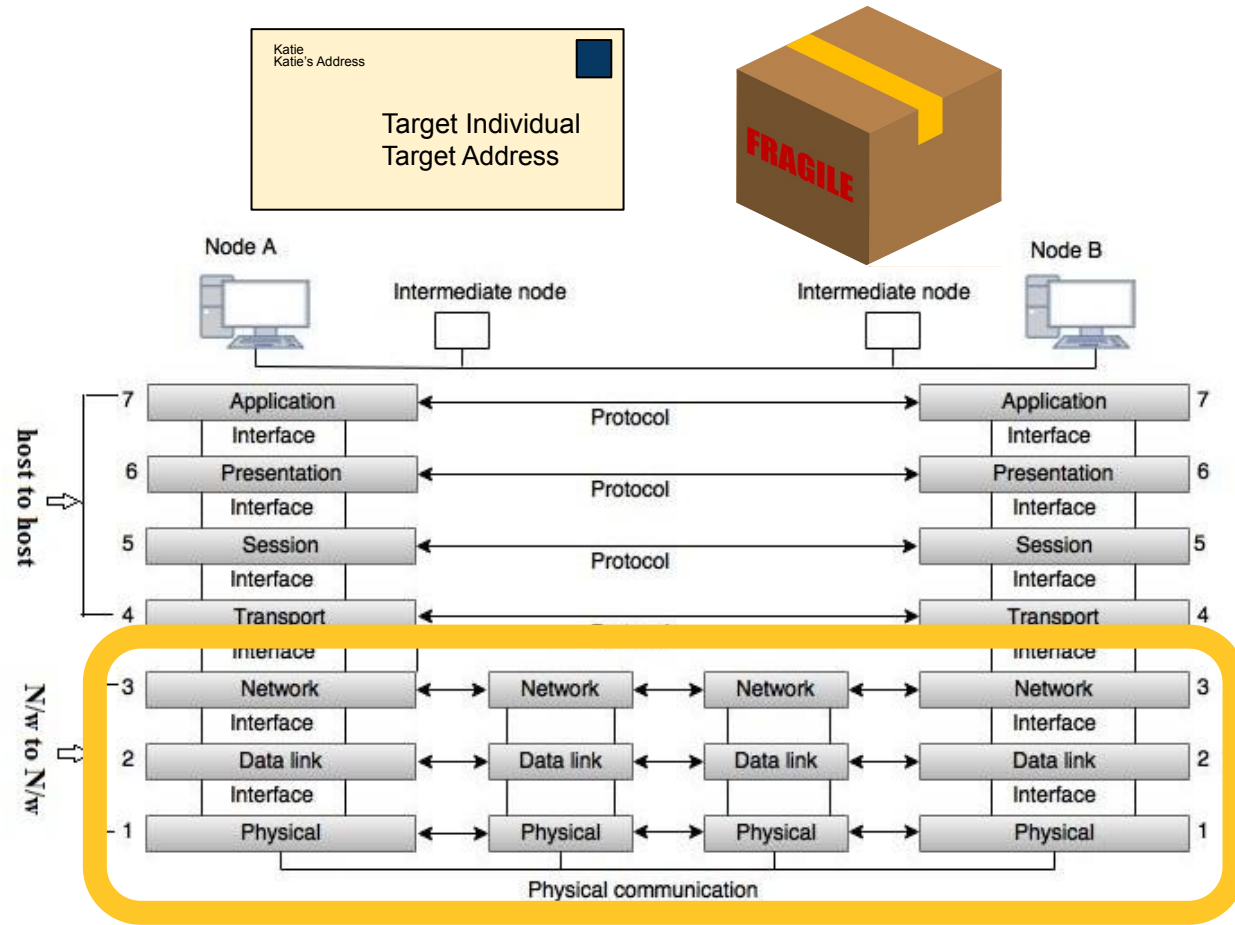


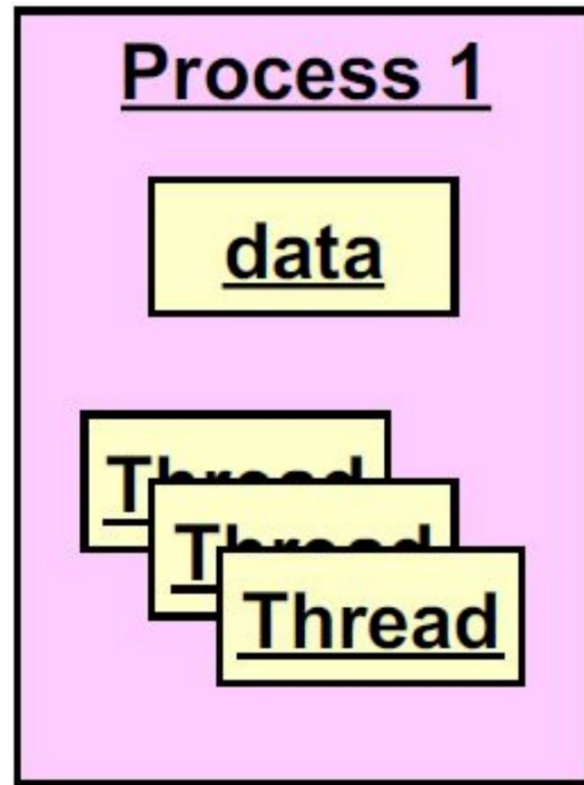
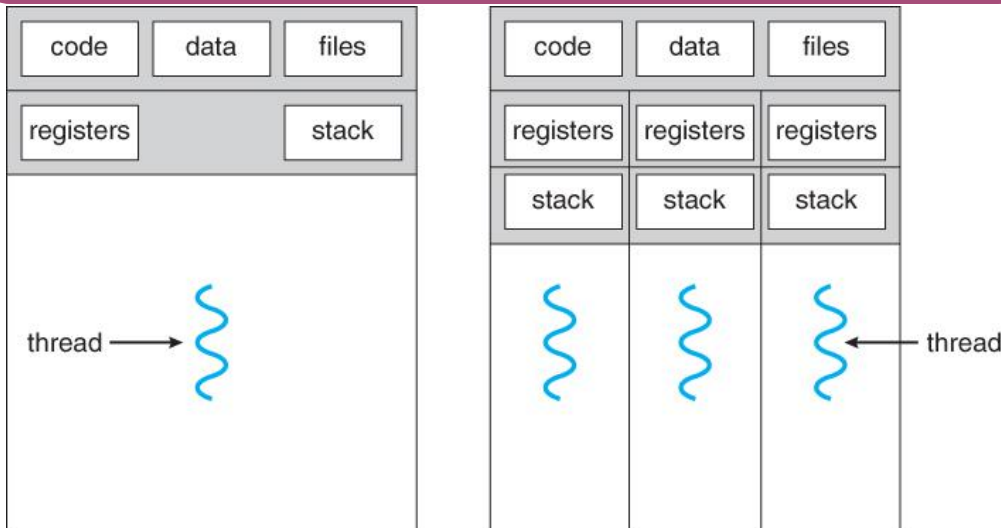
Fig: OSI Model

SER 321

Threads

What does that imply?

Remember that they exist *within* the parent process



SER 321

Threading Pitfalls

Race Condition

A thread never gains access to the resource it needs

Starvation

A thread is only able to acquire some of the resources it needs

Deadlock

More than one thread accesses a single resource at the same time

SER 321

Threading Pitfalls

Race Condition

A thread never gains access to the resource it needs

Starvation

A thread is only able to acquire some of the resources it needs

Deadlock

More than one thread accesses a single resource at the same time

SER 321

Threading Pitfalls

As the project name implies, we encounter a **deadlock**.

But what happened?

```
class SockClient {  
    public static void main (String args[]) throws Exception {  
        Socket      sock = new Socket( host: "localhost", port: 8888);    //Any IP name  
  
        ObjectInputStream in = new ObjectInputStream(sock.getInputStream());  
        ObjectOutputStream out = new ObjectOutputStream(sock.getOutputStream());  
  
        String s = (String) in.readObject();  
        out.writeObject("Back at you");  
  
        in.close();  
        out.close();  
        sock.close();  
    }  
}
```

Client

```
class SockServer {  
    public static void main (String args[]) throws Exception {  
  
        int count = 0;  
        ServerSocket      serv = new ServerSocket( port: 8888);  
  
        Socket sock = serv.accept();  
  
        ObjectInputStream in = new ObjectInputStream(sock.getInputStream());  
        ObjectOutputStream out = new ObjectOutputStream(sock.getOutputStream());  
  
        String s = (String) in.readObject();  
        System.out.println("Received " + s);  
        out.writeObject("Back at you");  
        System.out.println("Received " + s);  
  
        in.close();  
        out.close();  
        sock.close();  
    }  
}
```

Server

```
PS C:\ASU\SER321\examples_repo\ser321examples\Threads\NetworkDeadlock> gradle  
server  
<=====--> 75% EXECUTING [1m 33s]  
> :server  
█
```

```
PS C:\ASU\SER321\examples_repo\ser321examples\Threads\NetworkDeadlock> gradle  
client  
Starting a Gradle Daemon, 1 busy and 1 stopped Daemons could not be reused, use  
--status for details  
<=====--> 75% EXECUTING [53s]  
> :client  
█
```

SER 321

Threading Pitfalls

What does *Spaghetti Consumed* represent?

What does *Thinking* represent?

What does *Hungry* represent?

powered by NetLogo

Dining Philosophers

File: New Revert to Original
Export: NetLogo HTML

Mode: Interactive Commands and Code: Bottom

model speed

ticks: 6712

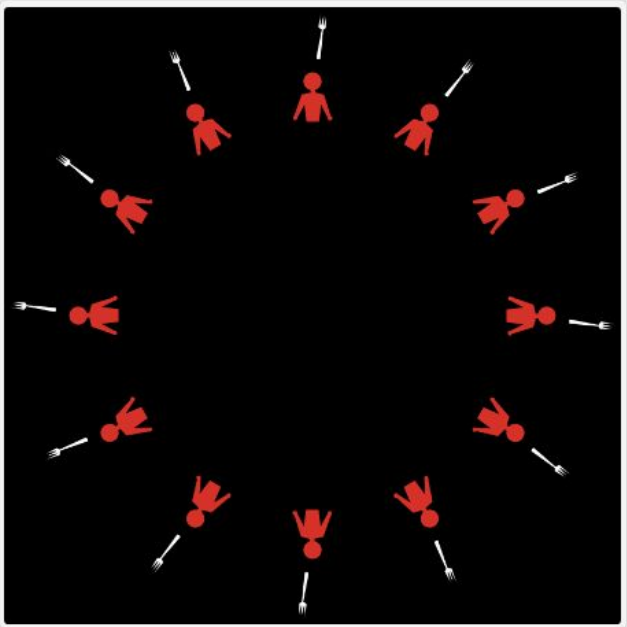
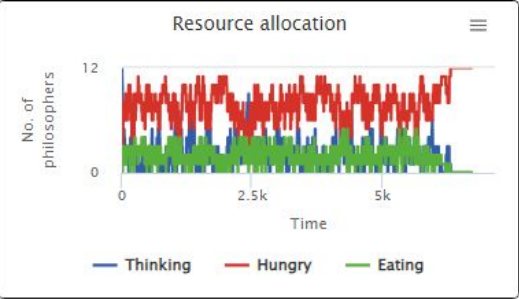
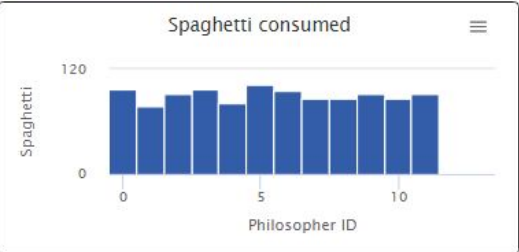
num-philosophers 12

setup go go once

hungry-chance 0.5

full-chance 0.5

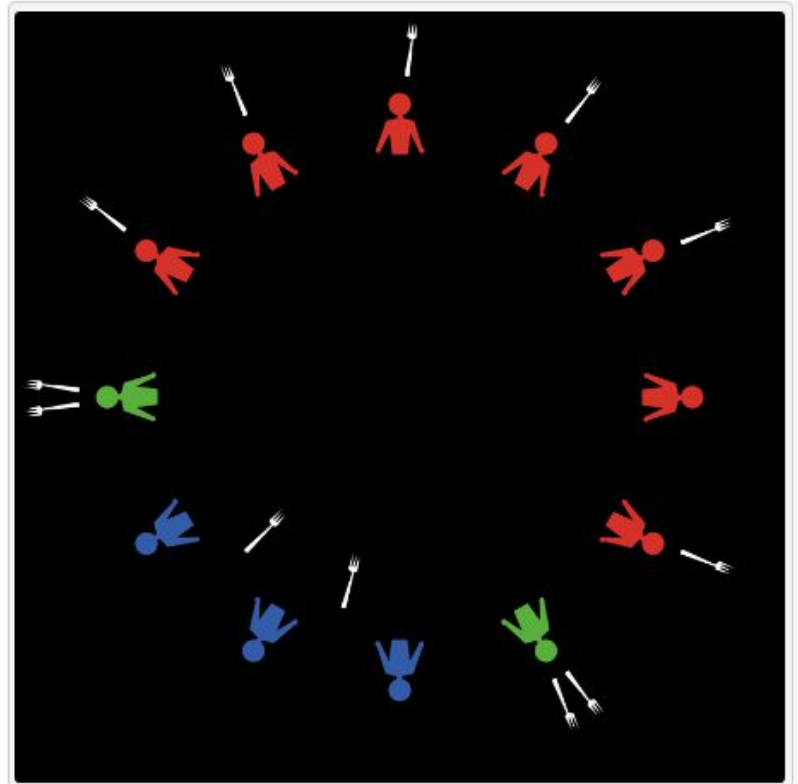
☐ cooperation?



SER 321

Threading Pitfalls

Can we take a guess at what happened here?



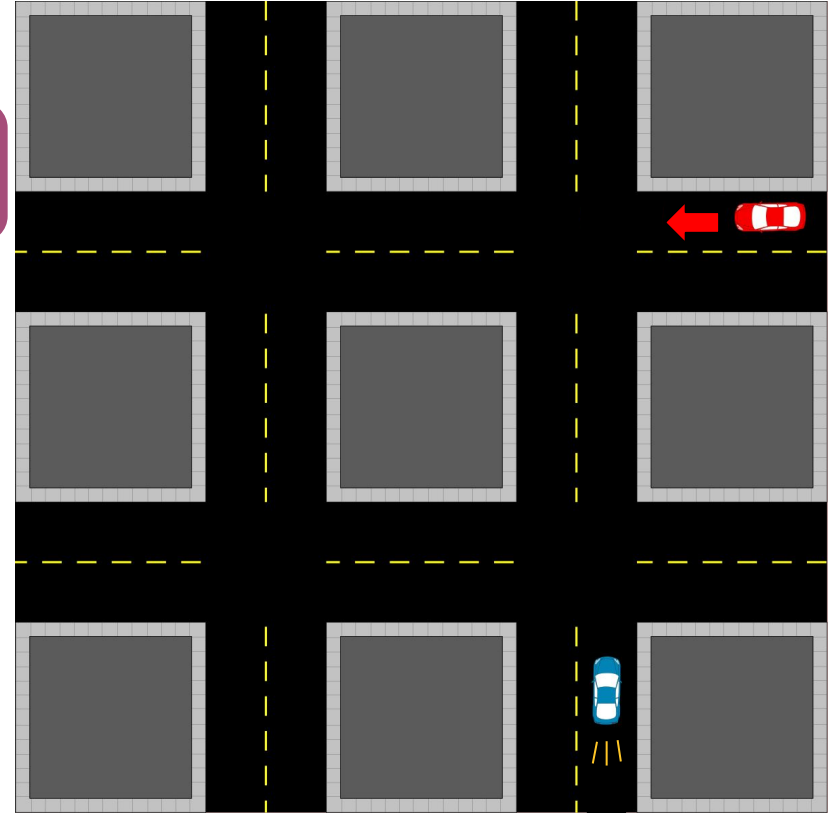
SER 321

Threading Pitfalls

Race Condition

Crash

More than one thread accesses a single resource at once



SER 321

Threading Pitfalls

Race Condition

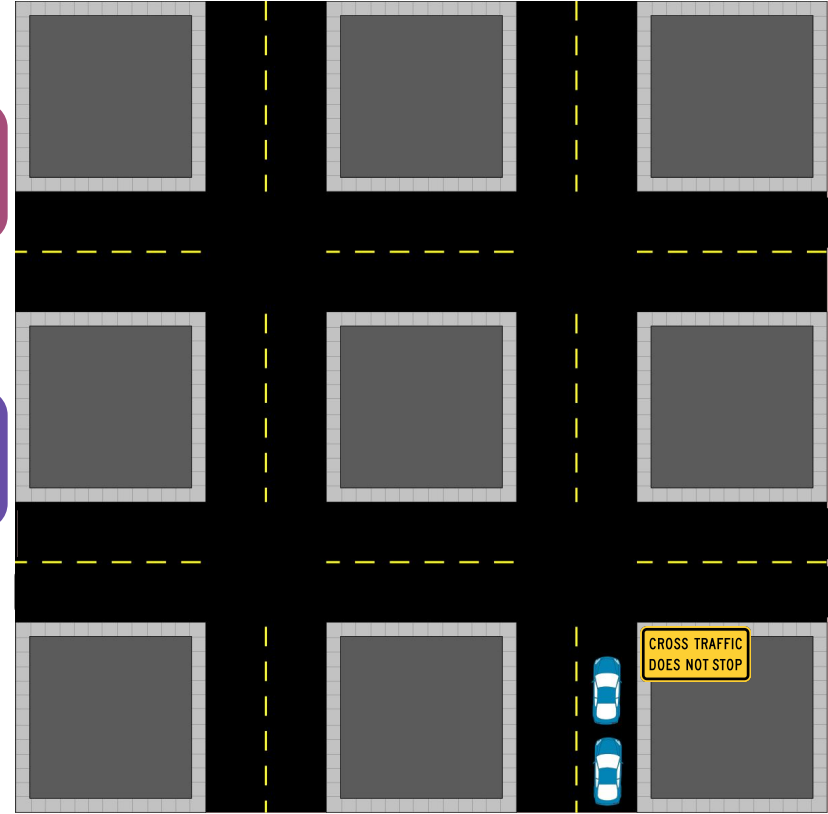
Crash

More than one thread accesses a single resource at once

Starvation

Cross Traffic

A thread never gains access to the resource it needs



SER 321

Threading Pitfalls

Race Condition

Crash

More than one thread accesses a single resource at once

Starvation

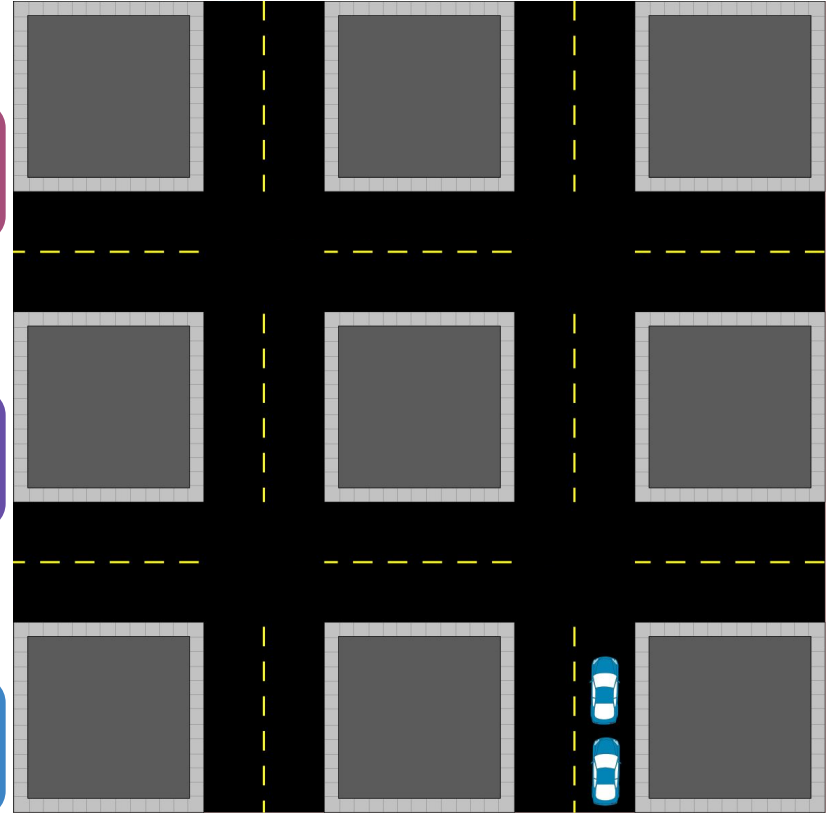
Cross Traffic

A thread never gains access to the resource it needs

Deadlock

Gridlock

A thread is only able to acquire some of the needed resources



SER 321

Concurrency Structures

Can we name some concurrency structures?

Atomic Operations &
Variables

Locks

Semaphores

Monitors

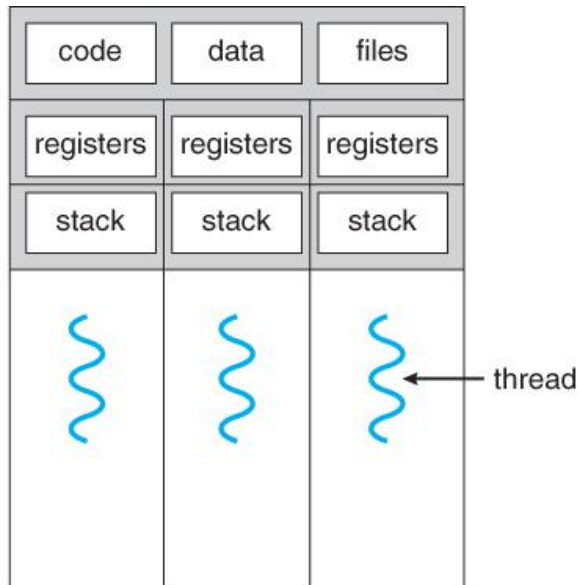
SER 321

Concurrency Structures

Atomic Operations & Variables

Recall *registers*...

Ensures updates are immediately visible for the local copy in *each thread*



main:

```
pushq    %rbp
movq     %rsp, %rbp
subq     $48, %rsp
call     __main
movl     $5, -4(%rbp)
movl     $12, -8(%rbp)
movl     -4(%rbp), %eax
addl     $7, %eax
movl     %eax, -12(%rbp)
movl     -8(%rbp), %edx
movl     -12(%rbp), %eax
addl     %edx, %eax
movl     %eax, -16(%rbp)
movl     -16(%rbp), %eax
movl     %eax, %edx
leaq     .LC0(%rip), %rax
movq     %rax, %rcx
call     printf
movl     $0, %eax
addq     $48, %rsp
popq     %rbp
ret
```


SER 321

Concurrency Structures

Pros and Cons?

Locks

Acquire the Lock



Open & Enter

Close & Lock

Release the Lock

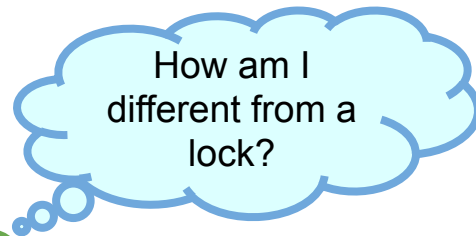


Unlock & Exit



SER 321

Concurrency Structures



Semaphores



More
than one
stall!

Acquire Lock



Open & Enter

Close & Lock

Release Lock



Unlock & Exit

Semaphores support
more than one acquirer

When would that be beneficial?

SER 321

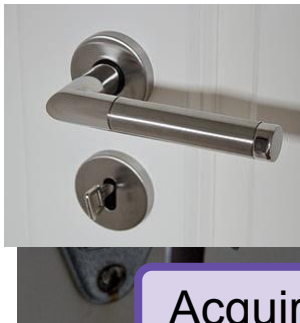
Concurrency Structures

Pros and Cons?

Monitors



You lock
the main
door
instead!



Acquire Lock



Open & Enter

Close & Lock

Release Lock



Unlock & Exit

Covers the
entire object

SER 321

Concurrency Structures

RECAP

Atomic Operations &
Variables

YOU control the
locks directly

Locks

YOU control the
locks directly

Semaphores

YOU control the
locks directly

Monitors

Locks managed
for you

SER 321

Threaded Server

Given the standard server socket steps...

Ideas on how we could introduce threads?

1. Define Params

2. Create Socket

3-5. Mark Socket to Listen

6. Wait for Connection

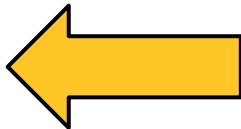
7. Handle Client Connection

8. Close Client Connection

9. Continue Listening

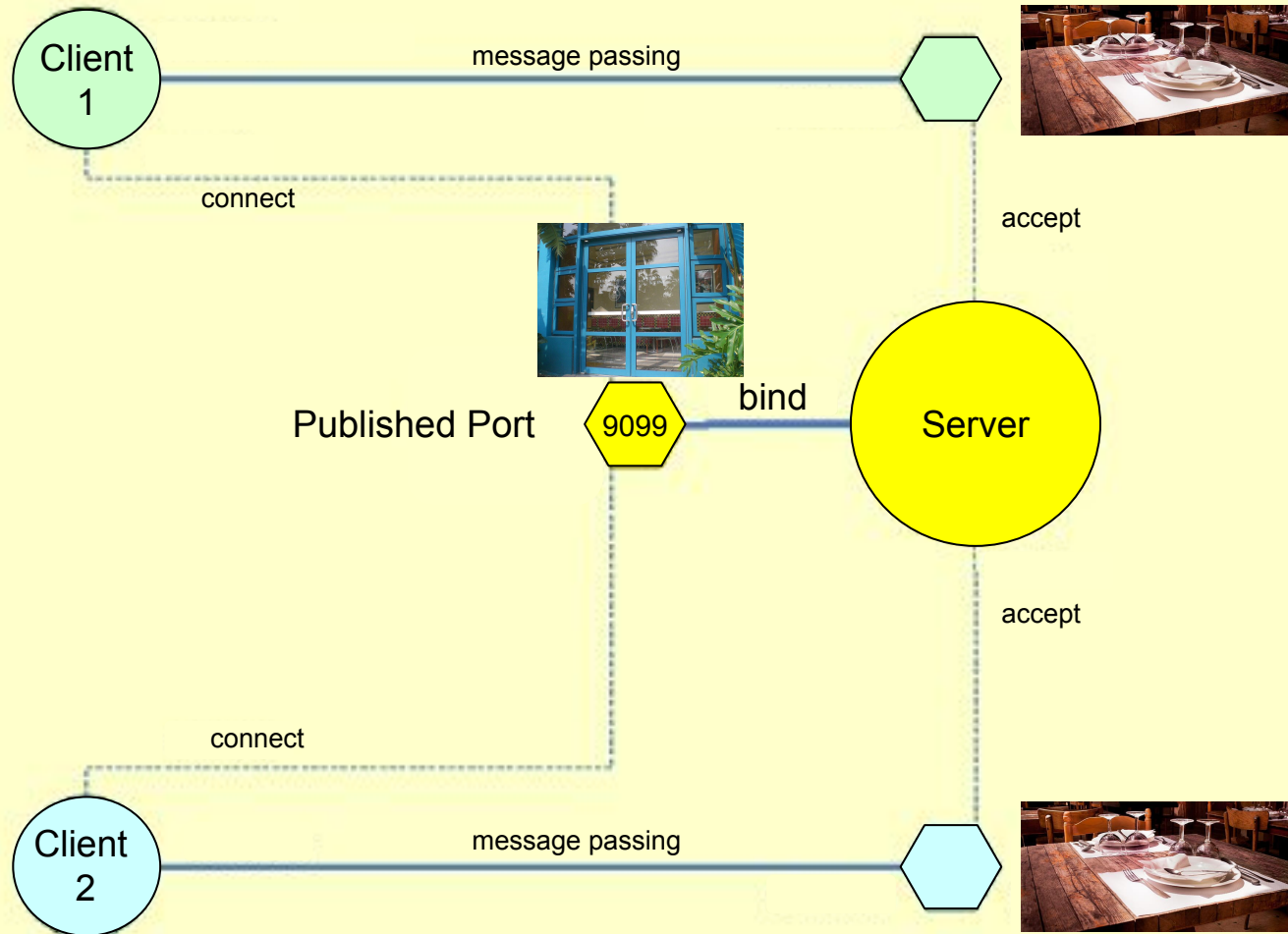
Why do we send the *client socket* to the thread?

7. Send Client Socket to thread



SER 321

Sockets!



SER 321

Scratch Space

Questions?



Survey:

<http://bit.ly/ASN2324>



Upcoming Events

SI Sessions:

- Thursday, June 13th at 6:00 pm MST
- Sunday, June 16th at 6:00 pm MST
- Monday, June 17th at 6:00 pm MST

Review Sessions:

- Review Session - **Wednesday**, July 3rd at 6:00 pm MST (2 hr Session)
- Q&A Session - Sunday, July 7th at 6:00 pm MST (Final Session)

More Questions?

Check out our other resources!

tutoring.asu.edu



Academic Support

Academic Support Network (ASN) provides a variety of free services in-person and online to help currently enrolled ASU students succeed academically.

Services



Subject Area Tutoring

Need in-person or online help with math, science, business, or engineering courses? Just hop into our Zoom room or drop into a center for small group tutoring. We'll take it from there.

[Need help using Zoom?](#)

[View the tutoring schedule](#)

[View digital resources](#)

Go to Zoom



Writing Tutoring

Need help with undergraduate or graduate writing assignments? Schedule an in-person or online appointment, access your appointment link, or wait in our drop-in queue.

[Access your appointment link](#)

[Access the drop-in queue](#)

Schedule Appointment



Online Study Hub

Join our online peer communities to connect with your fellow Sun Devils. Engage with our tools to search our bank of resources, videos, and previously asked questions. Or, ask our Tutorbot questions.

Now supporting courses in Math, Science, Business, Engineering, and Writing.

Online Study Hub

1-

Go to Zoom

2-

[Need help using Zoom?](#)

[View the tutoring schedule](#)

[View digital resources](#)



1. Click on 'Go to Zoom' to log onto our Online Tutoring Center.
2. Click on 'View the tutoring schedule' to see when tutors are available for specific courses.

More Questions?

Check out our other resources!

tutoring.asu.edu/online-study-hub

 **Academic Support Network**

 [Services](#)  [Faculty and Staff Resources](#) [About Us](#) 

[University College](#)

Online Study Hub

Online peer communities for students and tutors, YouTube channels, and Tutorbots.



What are online peer communities?

Individual courses have an online peer community that allows you to connect with your peers to post and answer questions and to develop study groups.



How can tutoring center videos help?

Videos can help supplement the learning you're doing in and outside of class and include step-by-step methods for how to understand concepts.



How does the Tutorbot work?

You can ask the Tutorbot questions about course concepts and the Tutorbot will recommend additional resources and examples to help address your questions.

Select a subject

- Any -

[Apply](#)



Academic Support Network



[Services](#) 

[Faculty and Staff Resources](#)

[About Us](#) 

[University College](#)

Select a subject

- Any -

[Apply](#)

Business

ACC 231

Uses of Accounting Info I

 [Peer Community](#)

ACC 241

Uses of Accounting Info II

 [Peer Community](#)

CIS 105

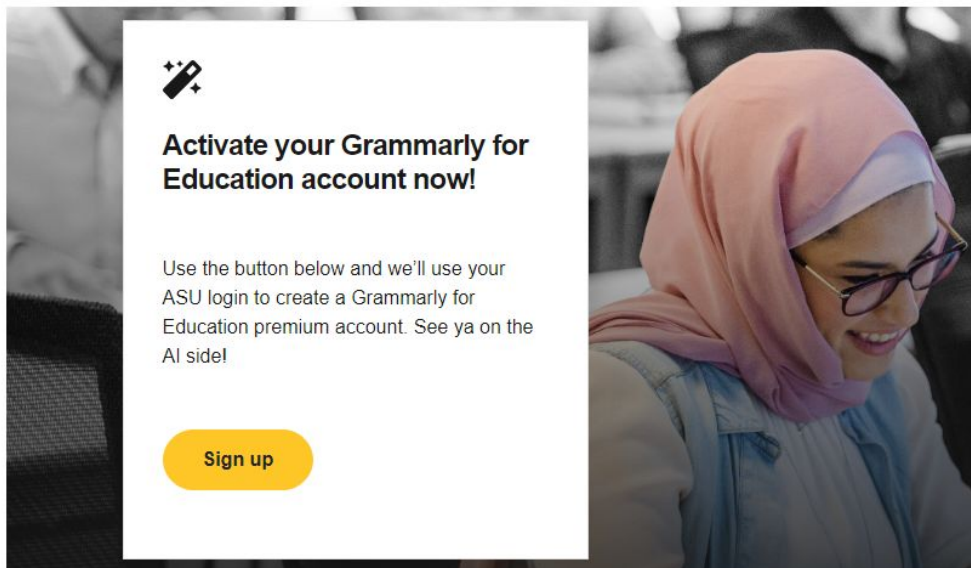
Computer Applications and Information Technology

 [Peer Community](#)

Don't forget to check out the Online Study Hub for additional resources!

Expanded Writing Support Available

Including Grammarly for Education, at no cost!



tutoring.asu.edu/expanded-writing-support

*Available slots for this pilot are limited

Additional Resources

- [Course Repo](#)
- [Gradle Documentation](#)
- [GitHub SSH Help](#)
- [Linux Man Pages](#)
- [OSI Interactive](#)
- [MDN HTTP Docs](#)
 - [Requests](#)
 - [Responses](#)
- [JSON Guide](#)
- [org.json Docs](#)
- [javax.swing package API](#)
- [Swing Tutorials](#)
- [Dining Philosophers Interactive](#)
- [Austin G Walters Traffic Comparison](#)