

SER 321 B Session

SI Session

Thursday, November 14th 2024

7:00 pm - 8:00 pm MST

Agenda



Protobuf PSA

Threading Pitfalls

Concurrency Structures

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

Quick PSA for Protobufs!

Check out the recording for the discussion!

You must generate the Protobufs for use!

4-2 Starter Code

SER 321

Protobufs

Assignment4.2given

Project

- Assignment4.2given
 - .gradle
 - .idea
 - gradle
 - src
 - main
 - java
 - client
 - server
 - Game
 - SockBaseServer**
 - proto
 - request.proto
 - response.proto
 - test
 - .gitignore
 - build.gradle
 - gradlew
 - gradlew.bat
 - PROTOCOL.md
 - README.md
 - External Libraries
 - Scratches and Consoles

```
1 package server;
2
3 > import ...
4
12
13 class SockBaseServer {
14     static String logFilename = "logs.txt"; 3 usages
15
16     // Please use these as given so it works with our test cases
17     static String menuOptions = "\nWhat would you like to do? \n 1 - to see
18     static String gameOptions = "\nChoose an action: \n (1-9) - Enter an int
19
20
21     ServerSocket serv = null; no usages
22     InputStream in = null; 3 usages
23     OutputStream out = null; 4 usages
24     Socket clientSocket = null; 4 usages
25     private final int id; // client id 2 usages
26
27     Game game; // current game 3 usages
28
29     private boolean inGame = false; // a game was started (you can de
30     private String name; // player name 4 usages
31
32     private int currentState =1; // I used something like this to keep track
33
34     private static boolean grading = true; // if the grading board should be
35
36     public SockBaseServer(Socket sock, Game game, int id) { 1 usage
37         this.clientSocket = sock;
38         this.game = game;
39
40         try
41             in = clientSocket.getInputStream();
```

Gradle

- Assignment4.2given
 - Tasks
 - build
 - build setup
 - documentation
 - help
 - other
 - arguments
 - compileJava
 - compileTestJava
 - components
 - dependentComponents
 - extractIncludeProto
 - extractIncludeTestProto
 - extractProto
 - extractTestProto
 - generateProto**
 - generateTestProto
 - model
 - prepareKotlinBuildScriptModel
 - processResources
 - processTestResources
 - runClient
 - runServer
 - runServerGrading
 - verification
 - Dependencies

Check out the recording for the discussion!

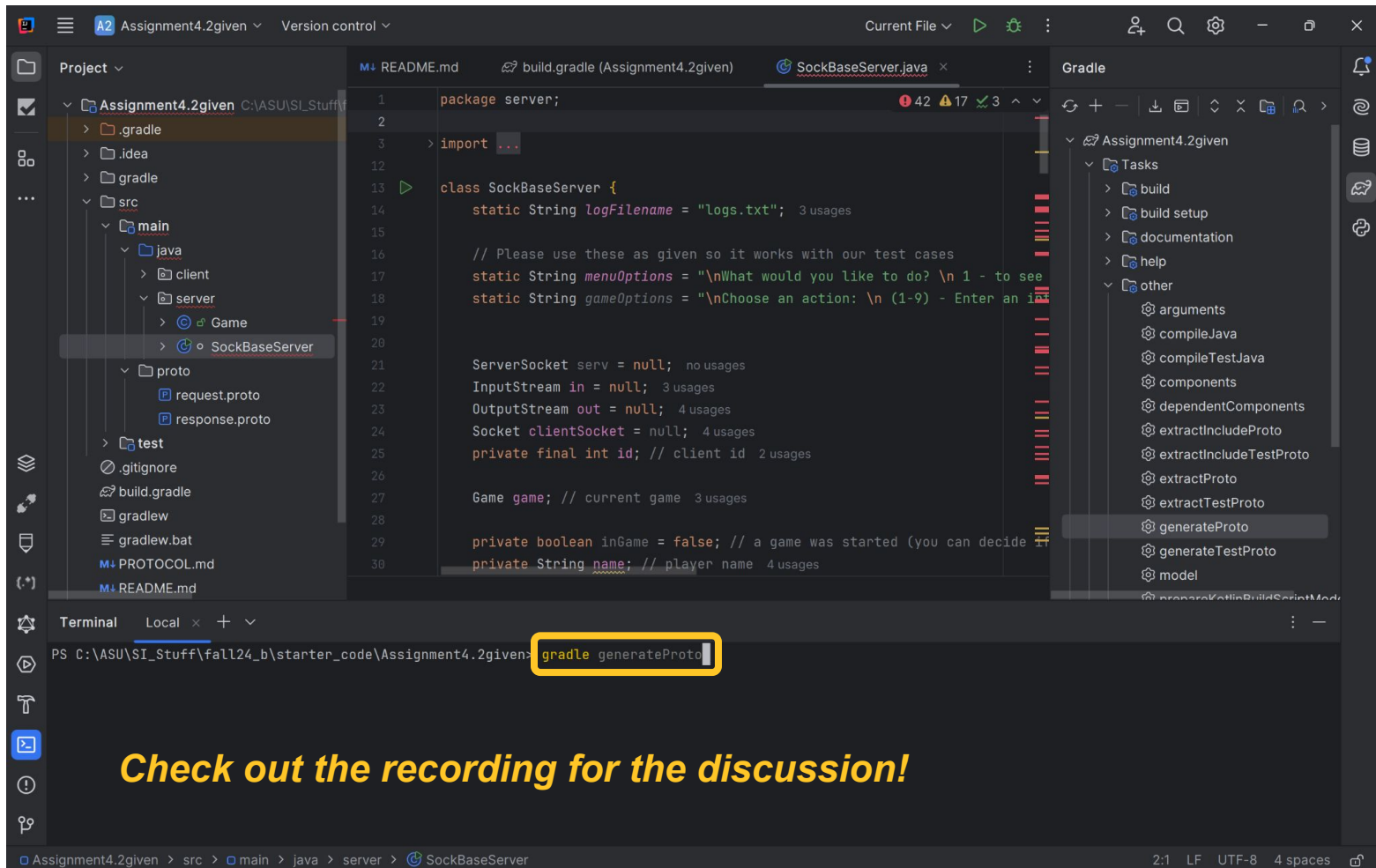
Assignment4.2given > src > main > java > server > SockBaseServer

2:1 LF UTF-8 4 spaces

4-2 Starter Code

SER 321

Protobufs



The screenshot shows an IDE with the following components:

- Project Explorer:** Shows the project structure for 'Assignment4.2given'. The 'src' directory contains 'main' (with 'java' and 'proto' subdirectories) and 'test'. The 'java' directory contains 'client', 'server', 'Game', and 'SockBaseServer'. The 'proto' directory contains 'request.proto' and 'response.proto'.
- Code Editor:** Displays the 'SockBaseServer.java' file. The code includes package declarations, imports, and a class definition for 'SockBaseServer' with various static and instance variables and methods.
- Gradle Panel:** Shows the 'Tasks' section for 'Assignment4.2given'. The 'other' category is expanded, showing tasks like 'arguments', 'compileJava', 'compileTestJava', 'components', 'dependentComponents', 'extractIncludeProto', 'extractIncludeTestProto', 'extractProto', 'extractTestProto', 'generateProto', 'generateTestProto', 'model', and 'prepareKotlinBuildScriptModel'. The 'generateProto' task is highlighted.
- Terminal:** Shows the command 'gradle generateProto' being executed in the terminal window.

Terminal Output:

```
PS C:\ASU\SI_Stuff\fall24_b\starter_code\Assignment4.2given> gradle generateProto
```

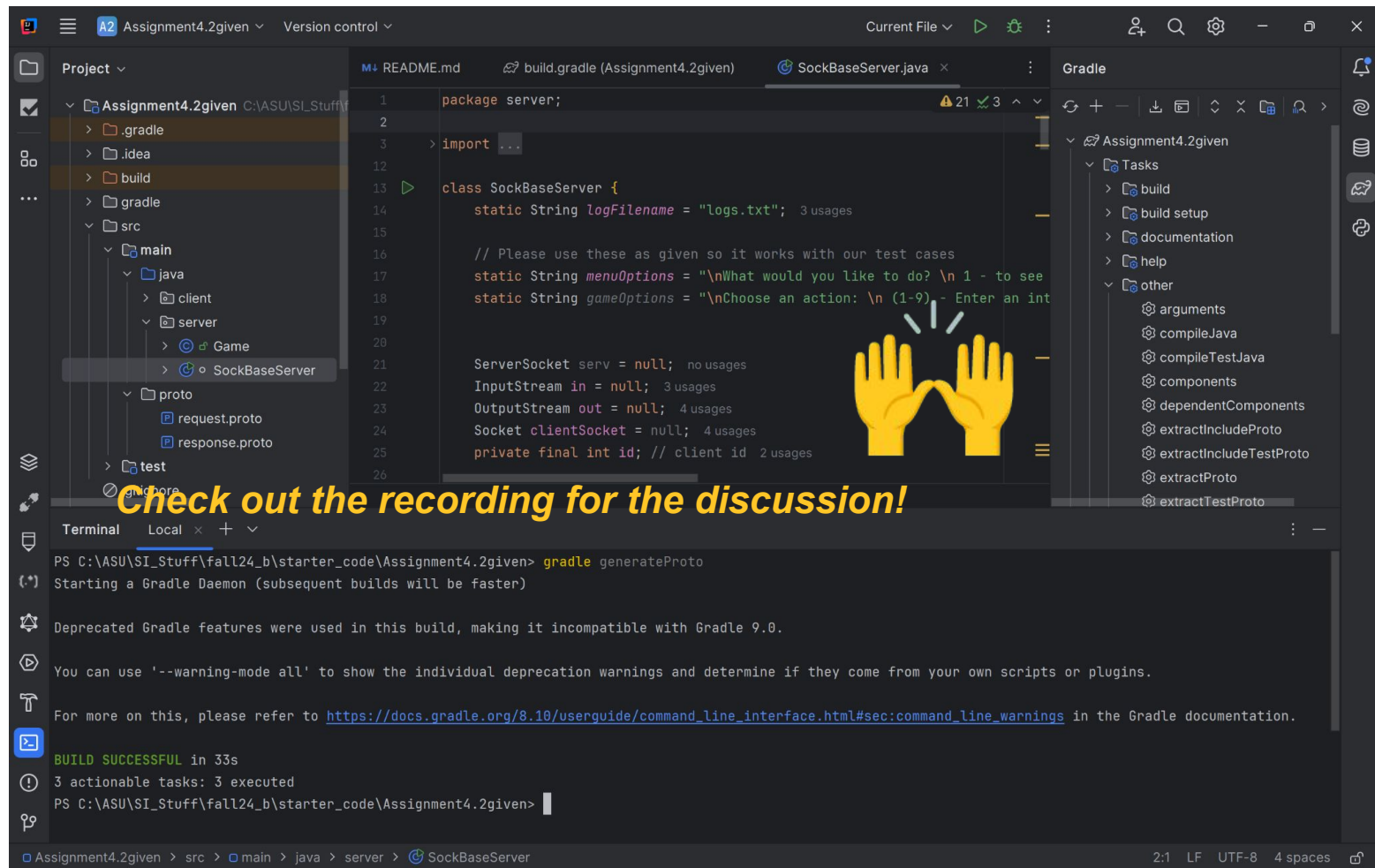
Bottom Bar: Shows the current file path: 'Assignment4.2given > src > main > java > server > SockBaseServer'.

Check out the recording for the discussion!

4-2 Starter Code

SER 321

Protobufs



The screenshot displays an IDE with the following components:

- Project Explorer:** Shows the project structure for 'Assignment4.2given'. The 'src/main/java/server' directory is expanded, highlighting the 'SockBaseServer' class.
- Code Editor:** Displays the content of 'SockBaseServer.java'. The code includes a package declaration, an import statement, and a class definition with static variables and methods.
- Gradle Panel:** Shows the 'Tasks' section for 'Assignment4.2given', listing various tasks like 'build', 'build setup', 'documentation', 'help', and 'other'.
- Terminal:** Shows the execution of the command 'gradle generateProto'. The output indicates that the build was successful, with 3 actionable tasks executed in 33 seconds.

Code Snippet (SockBaseServer.java):

```
1 package server;
2
3 import ...
12
13 class SockBaseServer {
14     static String logFilename = "logs.txt"; 3 usages
15
16     // Please use these as given so it works with our test cases
17     static String menuOptions = "\nWhat would you like to do? \n 1 - to see
18     static String gameOptions = "\nChoose an action: \n (1-9) - Enter an int
19
20     ServerSocket serv = null; no usages
21     InputStream in = null; 3 usages
22     OutputStream out = null; 4 usages
23     Socket clientSocket = null; 4 usages
24     private final int id; // client id 2 usages
25
26
```

Terminal Output:

```
PS C:\ASU\SI_Stuff\fall24_b\starter_code\Assignment4.2given> gradle generateProto
Starting a Gradle Daemon (subsequent builds will be faster)

Deprecated Gradle features were used in this build, making it incompatible with Gradle 9.0.

You can use '--warning-mode all' to show the individual deprecation warnings and determine if they come from your own scripts or plugins.

For more on this, please refer to https://docs.gradle.org/8.10/userguide/command\_line\_interface.html#sec:command\_line\_warnings in the Gradle documentation.

BUILD SUCCESSFUL in 33s
3 actionable tasks: 3 executed
PS C:\ASU\SI_Stuff\fall24_b\starter_code\Assignment4.2given>
```

Bottom Bar: Shows the current file path: 'Assignment4.2given > src > main > java > server > SockBaseServer'.

Check out the recording for the discussion!

SER 321

Protobufs

Options for Message Creation

```
Response.newBuilder()  
    .setResponseType(Response.ResponseType.GREETING)  
    .setMessage("Hello " + name + " and welcome to a simple game of Sudoku.")  
    .setMenuoptions(menuOptions)  
    .setNext(currentState)  
    .build();
```

SockBaseServer

Create the message in
a single statement

Create the message in
increments

```
Request.Builder req = Request.newBuilder();  
  
switch (response.getResponseTypes()) {  
    case GREETING:  
        System.out.println(response.getMessage());  
        req = chooseMenu(req, response);  
        break;
```

SockBaseClient - main

SER 321**Protobufs**

Options for Message Creation

Need one
more step...

```

static Request.Builder chooseMenu(Request.Builder req, Response response) throws IOException {
    while (true) {
        System.out.println(response.getMenuoptions());
        System.out.print("Enter a number 1-3: ");
        BufferedReader stdin = new BufferedReader(new InputStreamReader(System.in));
        String menu_select = stdin.readLine();
        System.out.println(menu_select);
        switch (menu_select) {
            // needs to include the other requests
            case "2":
                // this is not a complete START request!! Just as example
                req.setOperationType(Request.OperationType.START);
                return req;
            default:
                System.out.println("\nNot a valid choice, please try again.");
                break;
        }
    }
}

```

SockBaseClient - chooseMenu

```

Request.Builder req = Request.newBuilder();

switch (response.getResponseTypes()) {
    case GREETING:
        System.out.println(response.getMessage());
        req = chooseMenu(req, response);
        break;
}

```

SockBaseClient - main

```
req.build().writeDelimitedTo(out);
```

SER 321

Protobufs

Parsing Messages

GETTERS!

```
System.out.println("Got a response: " + response.toString());
```

```
switch (response.getResponse_type()) {  
    case GREETING:  
        System.out.println(response.getMessage());  
        req = chooseMenu(req, response);  
        break;
```

Fetch a single value

Fetch a *repeated* value

```
for (Entry lead: response3.getLeaderList()){  
    System.out.println(lead.getName() + ": " + lead.getPoints());  
}
```

SER 321

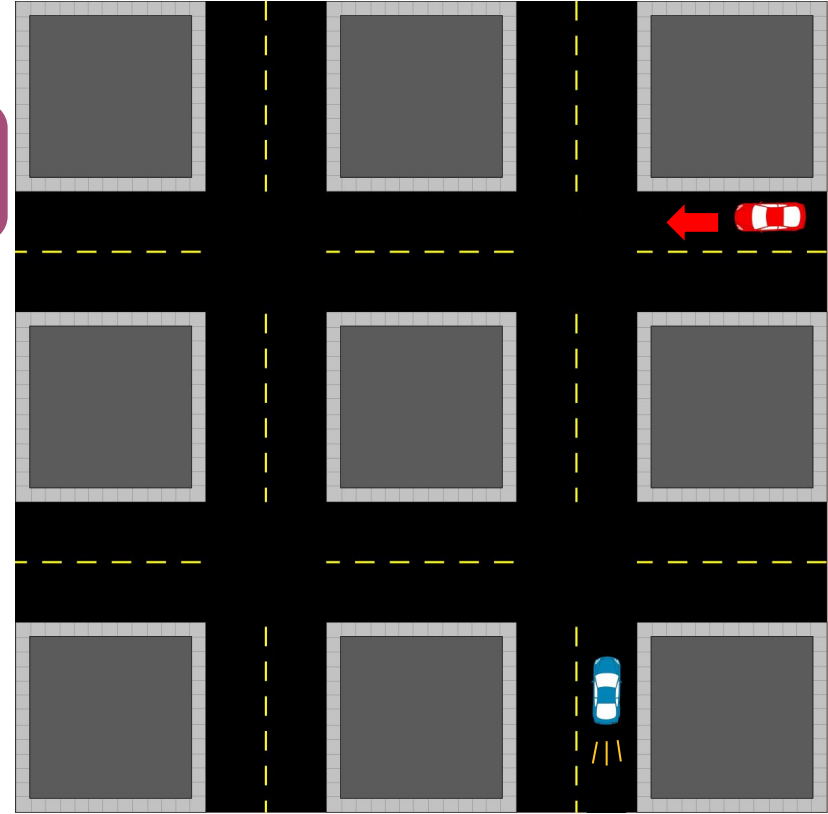
Threading Pitfalls

Check out the recording for the discussion!

Race Condition

Crash

More than one thread accesses a single resource at once



SER 321

Threading Pitfalls

Check out the recording for the discussion!

Race Condition

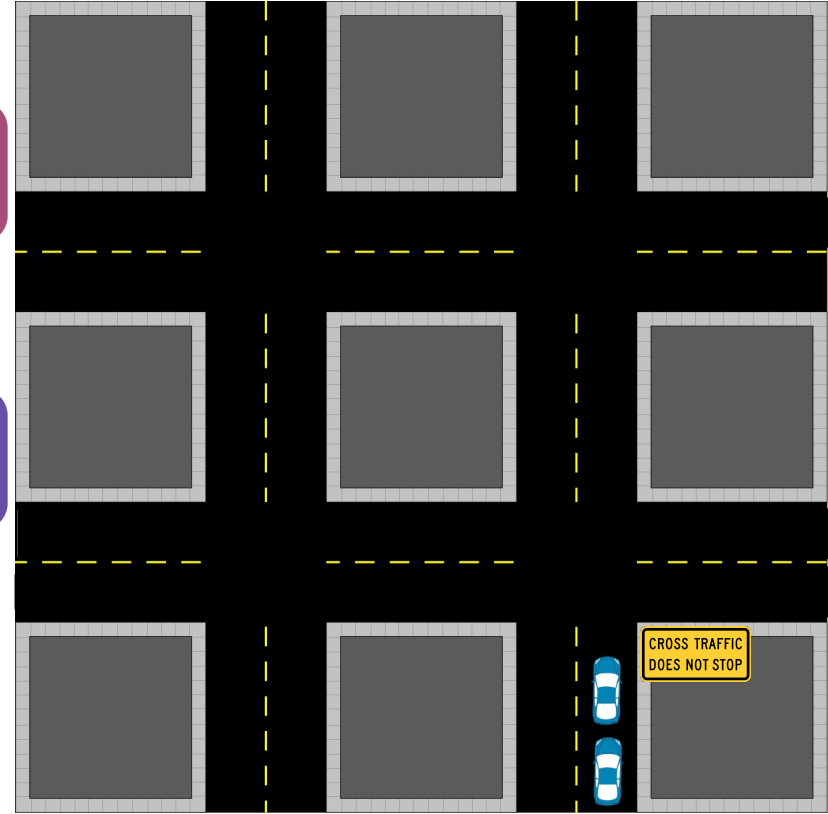
Crash

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

Check out the recording for the discussion!

Race Condition

Crash

More than one thread accesses a single resource at once

Starvation

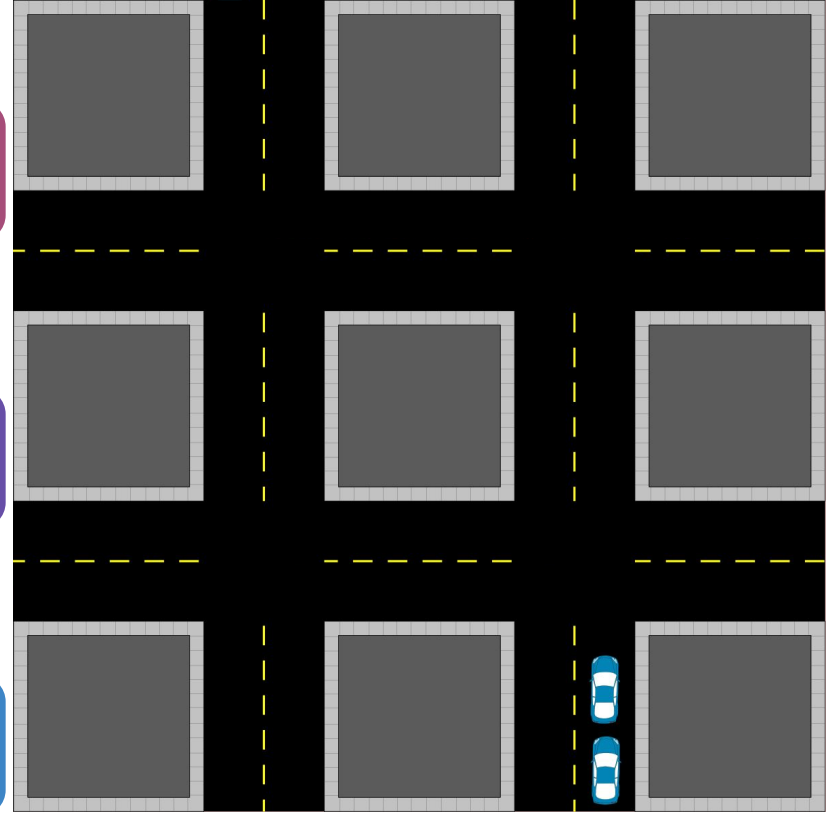
Cross Traffic

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

Check out the recording for the discussion!

Concurrency Structures

Can we name some concurrency structures?

Atomic Operations &
Variables

Locks

Semaphores

Monitors

SER 321

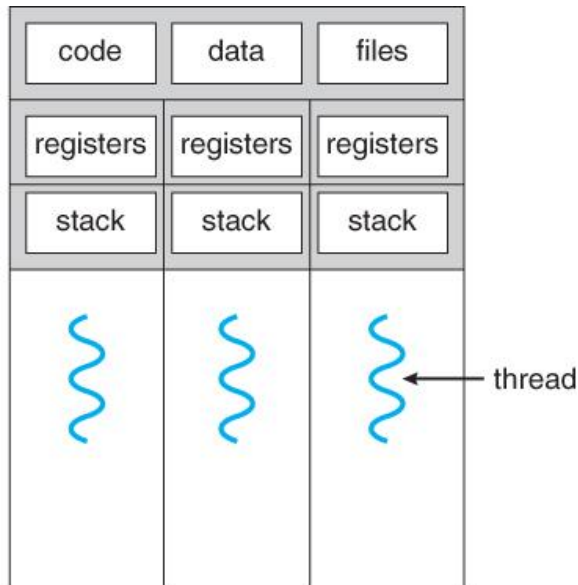
Concurrency Structures

Check out the recording for the discussion!

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
```


Check out the recording for the discussion!

SER 321

Concurrency Structures

Pros and Cons?



Locks

Acquire the Lock



Open & Enter

Close & Lock

Release the Lock



Unlock & Exit

Check out the recording for the discussion!

SER 321

Concurrency Structures

How am I
different from a
lock?

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?

Check out the recording for the discussion!

SER 321

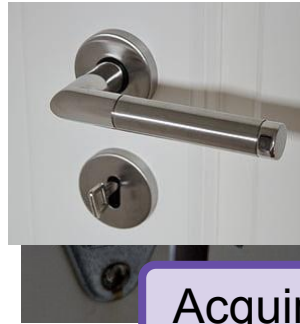
Concurrency Structures

Pros and Cons?

Monitors



You lock
the main
door
instead!



Acquire Lock



Open & Enter

Close & Lock

Covers the
entire object

Release Lock



Unlock & Exit

Check out the recording for the discussion!

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

Scratch Space

Upcoming Events

SI Sessions:

- Sunday, November 17th at 7:00 pm MST
- Tuesday, November 19th at 10:00 am MST
- Thursday, November 21st at 7:00 pm MST

Review Sessions:

- Sunday, December 1st at 7:00 pm MST - **2 hour Review Session**
- Tuesday, December 3rd at 10:00 am MST - **Q&A Session**

Questions?

Survey:

<https://asuasn.info/ASNSurvey>



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

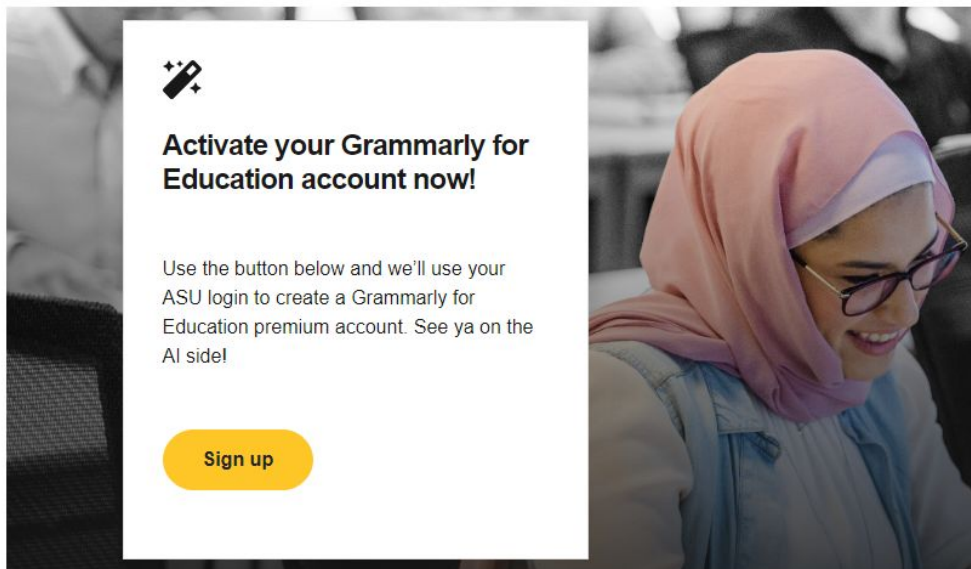
Computer Applications and Information Technology

 [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](#)