

SER 334 A Session

SI Session

Thursday, February 1st 2024

7:00 pm - 8:00 pm MST

Agenda



Process Tracing

Parallelism

Threading Issues

Dining Philosophers

SI Session Expectations

Thanks for coming to the **SER 334** 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 334
Module 5 Sample

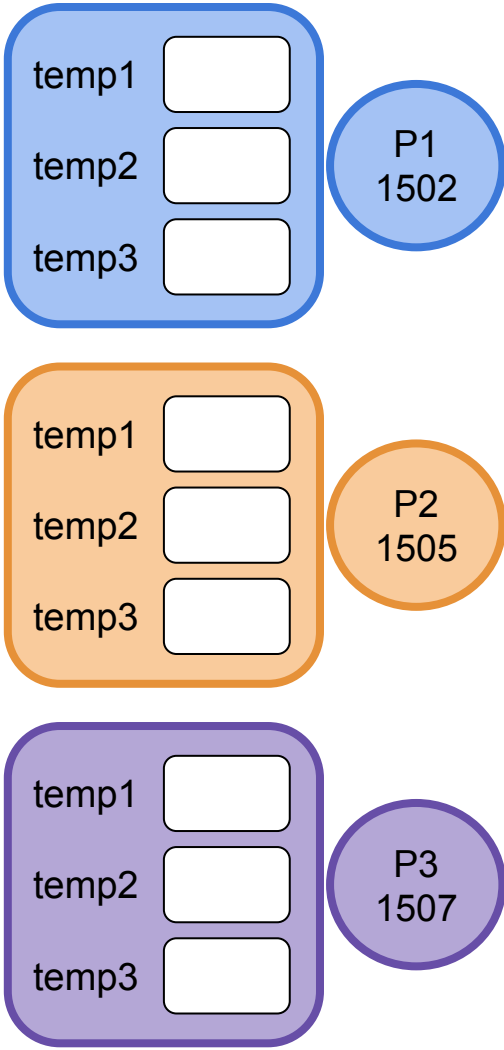
4. [Lisonbee] Trace the program below, identify the values of the pids at lines A, B, C, D, E, and F. (Assume that the actual pid of Process 1 is 1502, Process 2 is 1505, and Process 3 is 1507. Also assume that fork will always succeed.) [4 points]

| | |
|---|--|
| A | |
| B | |
| C | |
| D | |
| E | |
| F | |

```
int main() {
    pid_t temp1, temp2, temp3;
    temp1 = fork();

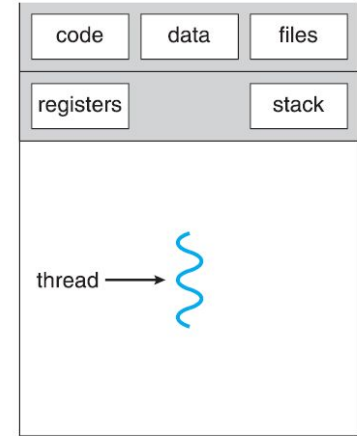
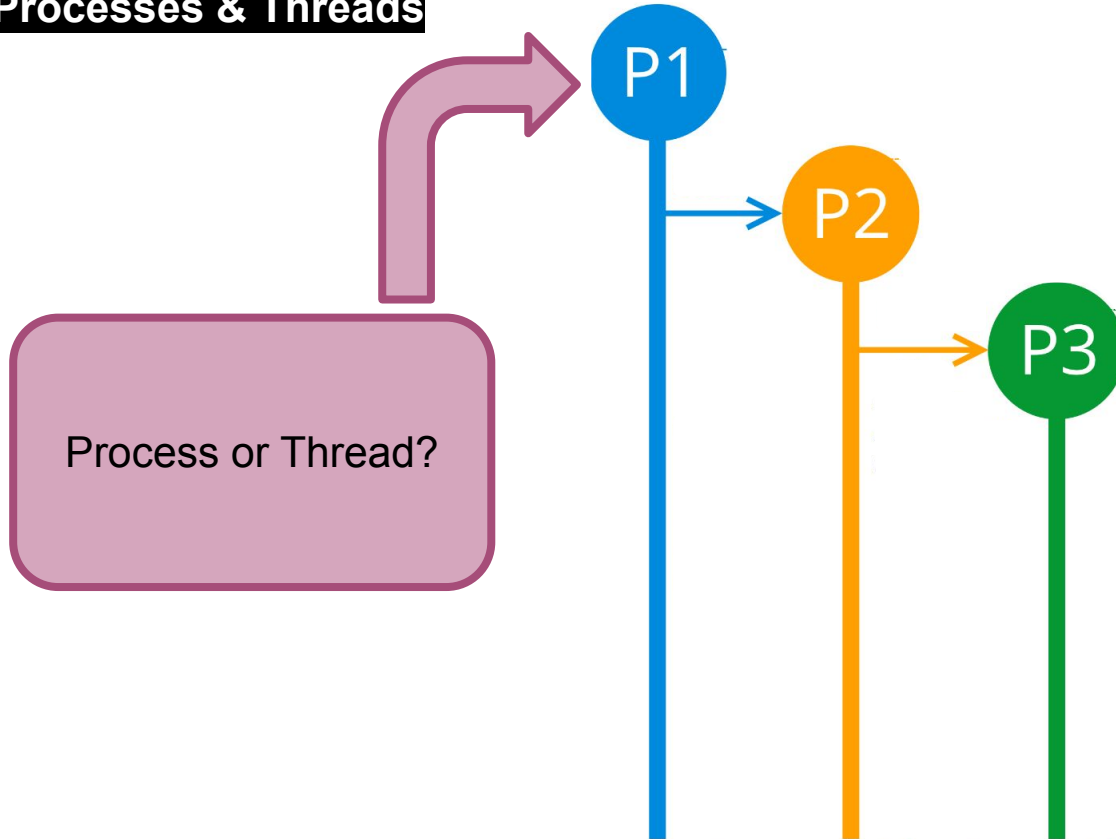
    if (temp1 < 0) { /* Error occurred */
        fprintf(stderr, "Fork Failed");
        return 1;
    }
    else if (temp1 == 0) { /* Process 2 */
        temp2 = fork();

        if (temp2 < 0) { /* Error occurred */
            fprintf(stderr, "Fork Failed");
            return 1;
        }
        else if (temp2 == 0) { /* Process 3 */
            temp3 = getpid();
            printf("temp2 = %d", temp2); /* A */
            printf("temp3 = %d", temp3); /* B */
        }
        else { /* Process 2 */
            temp3 = getpid();
            printf("temp2 = %d", temp2); /* C */
            printf("temp3 = %d", temp3); /* D */
            wait(NULL);
        }
    }
    else { /* Process 1 */
        temp2 = getpid();
        printf("temp1 = %d", temp1); /* E */
        printf("temp2 = %d", temp2); /* F */
        wait(NULL);
    }
    return 0;
}
```



SER 334

Processes & Threads



single-threaded process

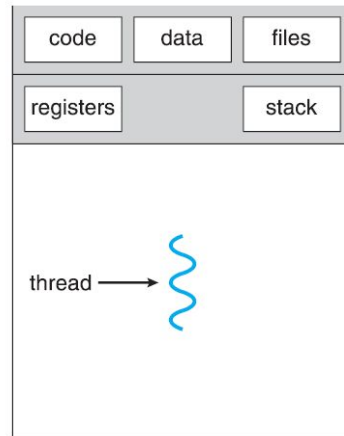
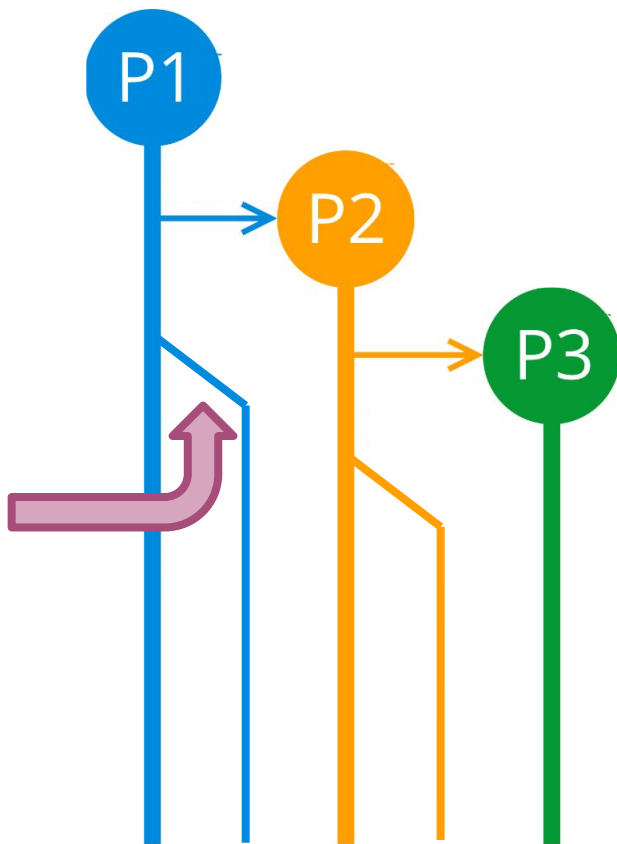
SER 334

Depicting Threads

You get a stack!

You get a stack!

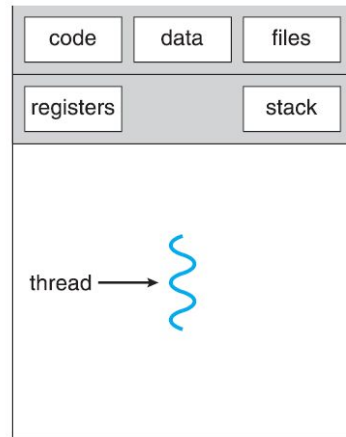
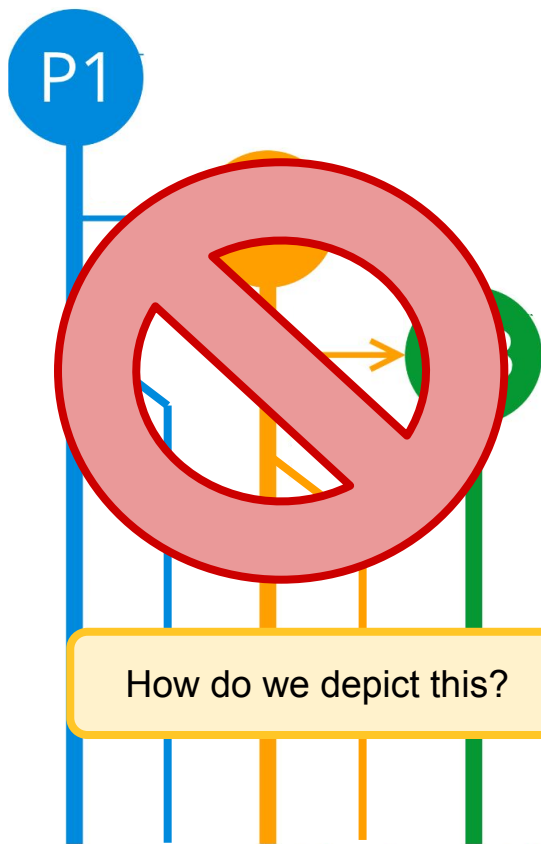
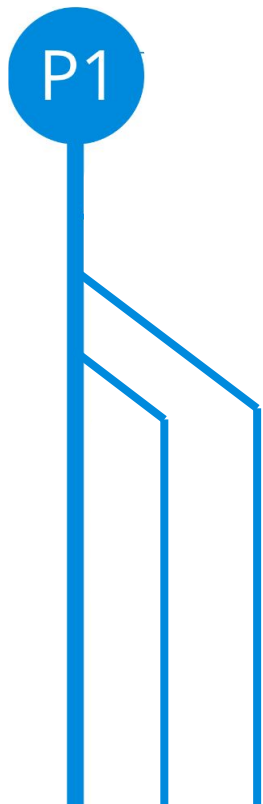
pthread_create



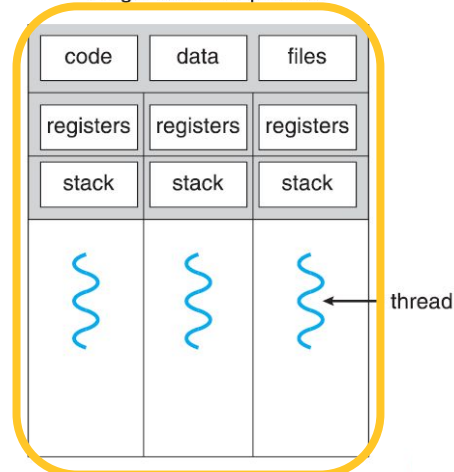
single-threaded process

SER 334

Depicting Threads



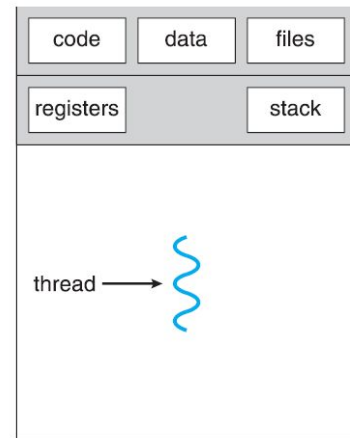
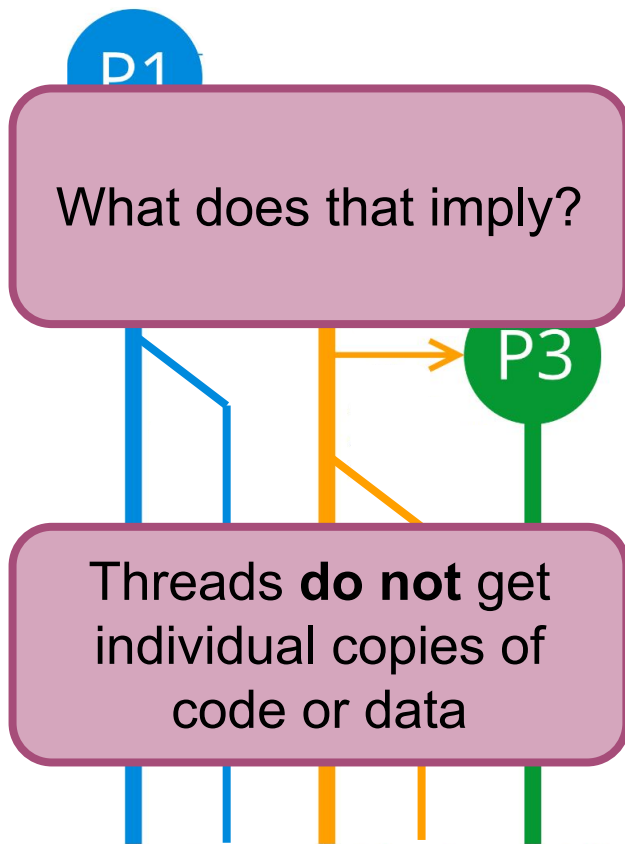
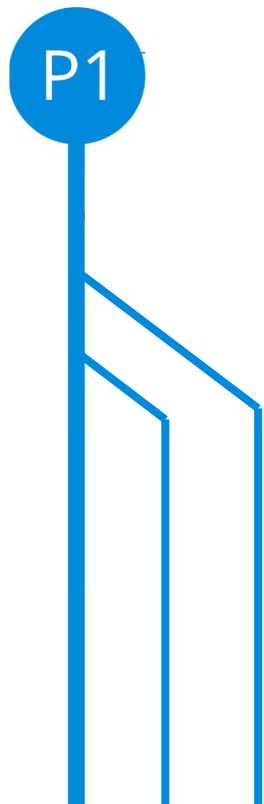
single-threaded process



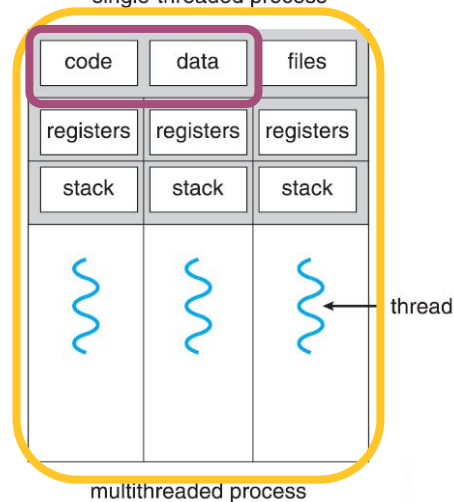
multithreaded process

SER 334

Depicting Threads



single-threaded process



multithreaded process

SER 334

Module 6 Samples

4. [Silberschatz 4.15 edited] Consider the following code fragment and answer the following questions:

- (a) Using “lifeline notation”, draw the creation of processes and threads during execution.
- (b) How many unique processes are created? (Do not include the initial process.)
- (c) How many unique threads are created? (Hint: processes don't count!)

```
pid_t pid;  
temp = fork();  
  
if (temp == 0) {  
    fork();  
}  
  
thread_create(...);
```

SER 334

Multi-Threading

What's the difference??

Foreshadowing for
CPU scheduling!



Parallelism

vs.

Concurrency

Simultaneous Execution

Near-Time Execution

Run at the same time

Exist at the same time

SER 334

Parallelism Types

Data Parallelism

vs.

Task Parallelism

**Same Task
Different Data**

**Same Data
Different Task**

SER 334

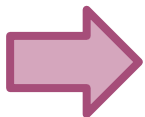
Parallelism Types

Data Parallelism

vs.

Task Parallelism

Many data sets



Same Task
Different Data

Same Data
Different Task



One data set

SER 334

Threading Issues

Check out the recording for the solution!

- Test and Debugging
 - A. Identifying independent functionality
- Data Splitting
 - B. Ensuring comparable amounts of work
- Identifying Tasks
 - C. Partitioning and minimizing memory use
- Data Dependency
 - D. Some tasks must be performed sequentially
- Balance
 - E. Multiple “executors” for each line makes it harder to identify the culprit

SER 334

Enter topic here

6. [Lisonbee] Consider a system where two processes (a producer and a consumer) use a message-passing system to communicate, and each process does work at different rates. The producer can produce (perform work) at any rate, but the consumer has to wait for the producer to complete its task before moving on. Based on this system's needs, explain whether a synchronous or asynchronous communication system would be a better choice and why. [2 points]

Synchronous

Asynchronous

SER 334

Scratch Space

Upcoming Events

SI Sessions:

- Sunday, February 4th at 7:00 pm MST
- Monday, February 5th at 7:00 pm MST
- ~~Sunday, February 11th at 7:00 pm MST~~ **Cancelled - Good luck on Exam 2!**

Review Sessions:

- Exam 2 Review: Thursday, February 8th 7:00 pm - 9:00 pm MST
- Exam 3 Review: TBD

Questions?

Survey:

<http://bit.ly/ASN2324>



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!

Additional Resources

- [Course Repo](#)
- [Course Discord](#)
- [BMP File Format \(Wiki\)](#)
- [Linux Kernel API](#)
- [Bootlin - Linux Cross Referencer](#)
- [Dining Philosophers Interactive](#)