# SER 334 A Session

## SI Session

**Monday, January 22nd 2024**

*7:00 pm - 8:00 pm MST*

# Agenda

- Run-Time Execution
- Review Memory
- OS Structures
- Module Samples

# 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

# What is a system call?

A call that needs a high level of privilege

Examples:

**SER 334**

**System Calls**

# What is a system call?

A call that needs a high level of privilege

## Main Categories:

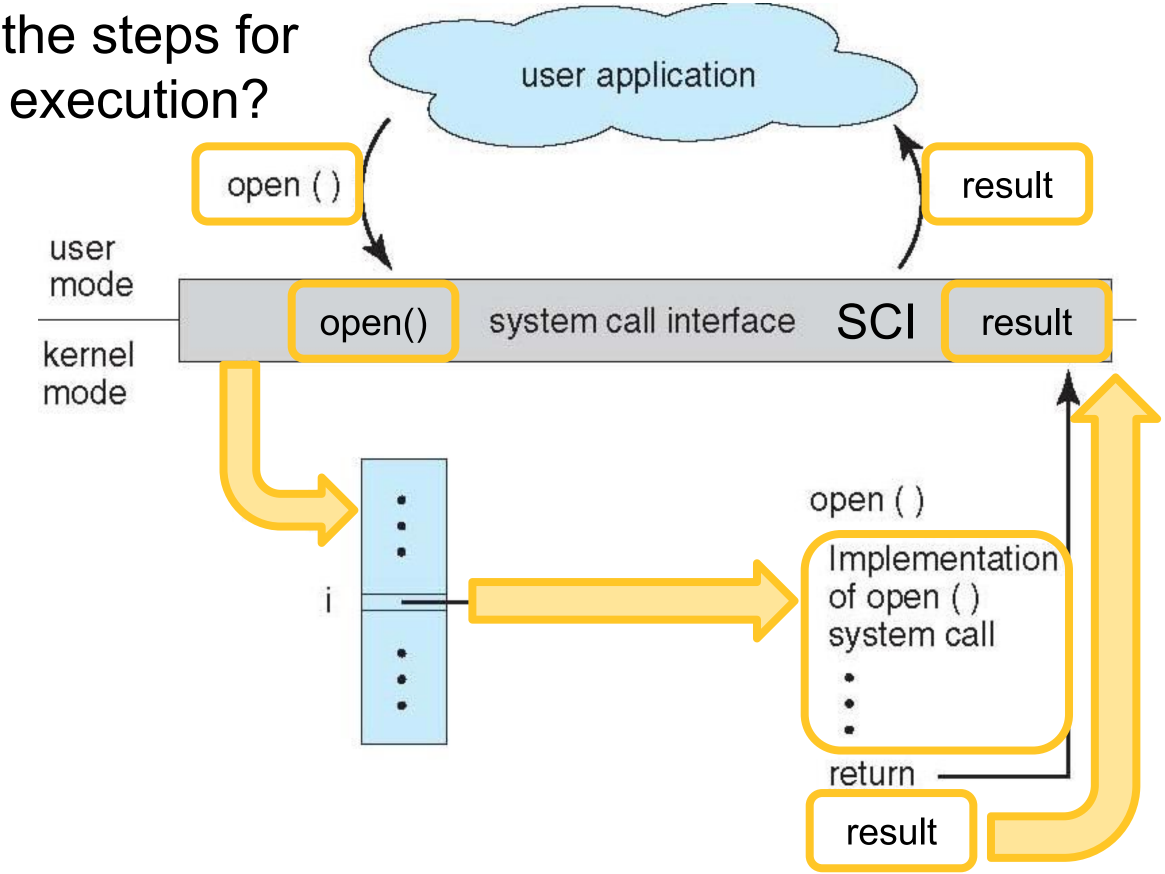
- Resource Allocation
- Interacting with Low-Level Hardware
- Potentially Dangerous Actions

# SER 334

## Run-Time Execution

## What are the steps for run-time execution?

- Make Call
- Go To: System Call Interface
- SCI looks up call in Table
- Find Implementation
- Run that implementation
- Obtain result and send to SCI
- Return result to user

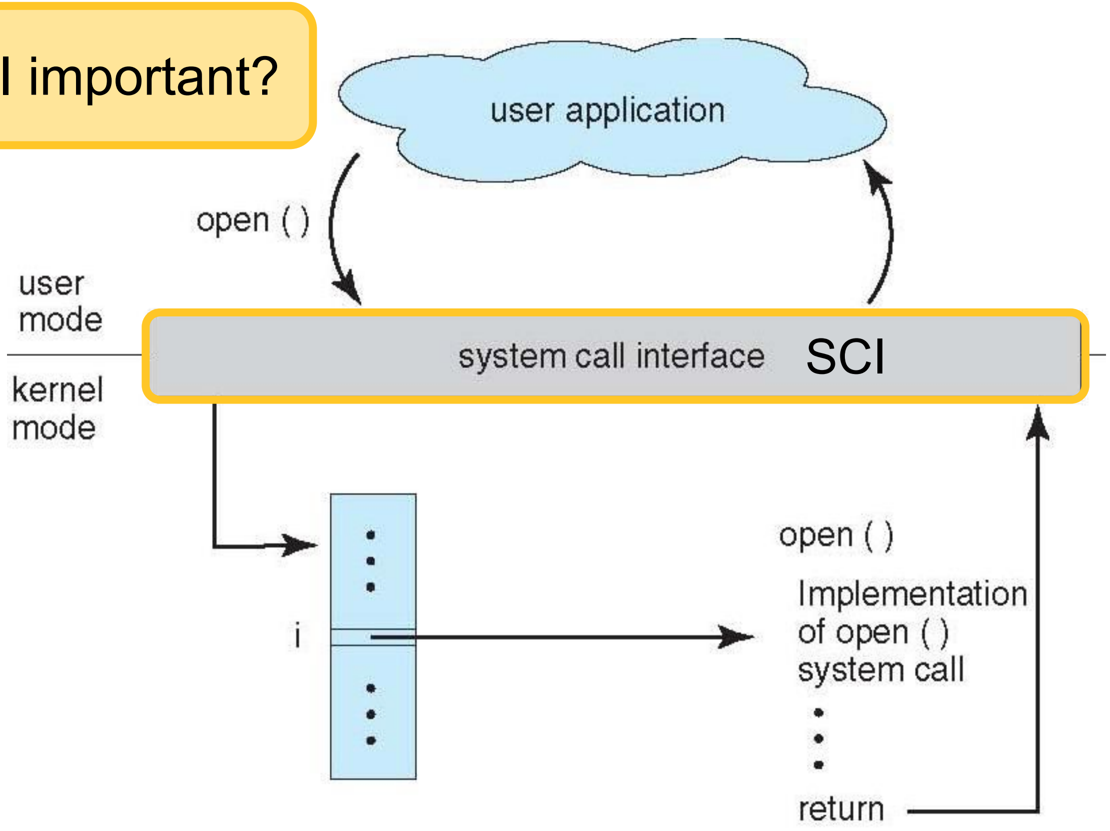


# SER 334

## Run-Time Execution

Why is the SCI important?

- Make Call
- Go To: System Call Interface
- SCI looks up call in Table
- Find Implementation
- Run that implementation
- Obtain result and send to SCI
- Return result to user

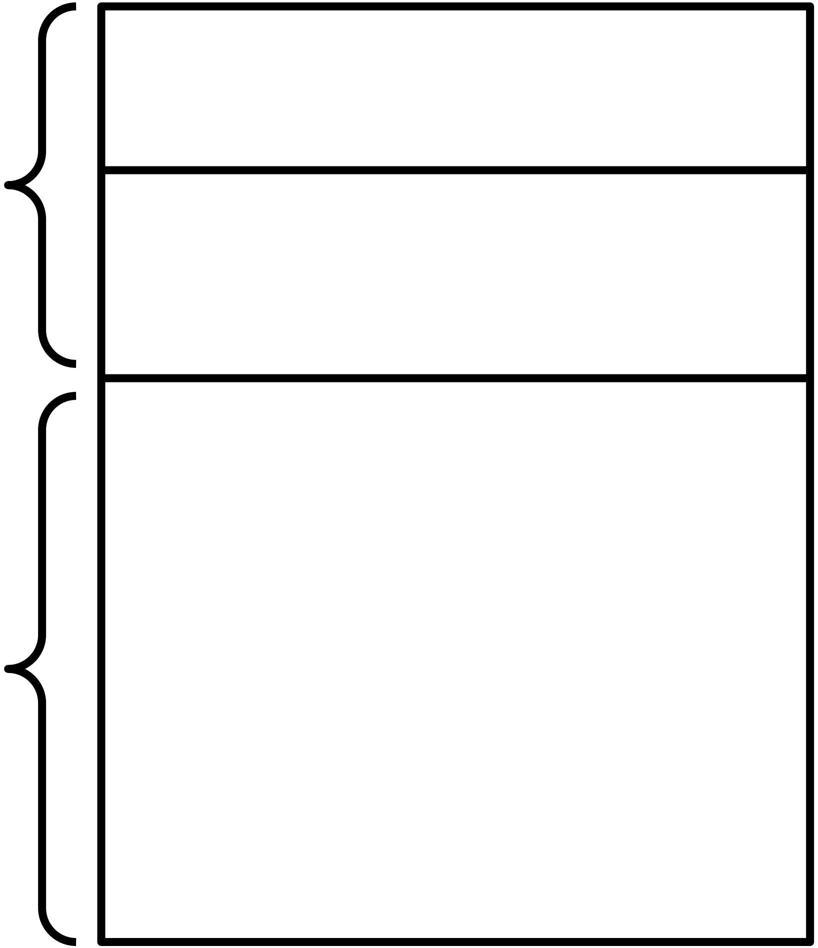


Abstraction - SCI will choose the correct implementation based on the hardware and the OS

**SER 334**

**Memory**

## Memory



## Contents:

Stack

Text/Code

Heap

Global & Static Variables/Objects

Static Memory

Dynamic Memory

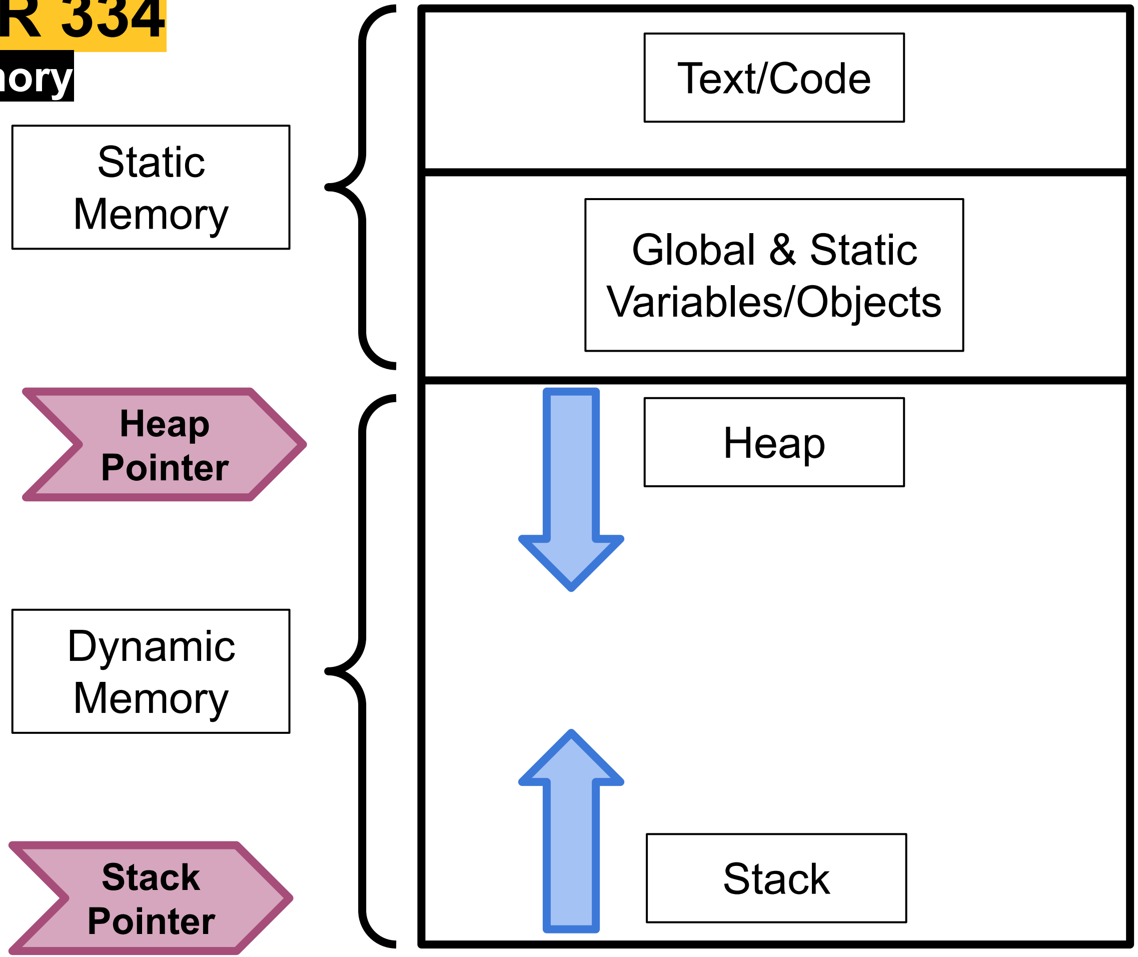
**Stack Pointer**

**Heap Pointer**

**SER 334**

**Memory**

# Memory



Text/Code

Global & Static Variables/Objects

Heap

Stack

Static Memory

Heap Pointer

Dynamic Memory

Stack Pointer

## SER 334

### Structure Types

Do we remember the 4 structure types?

Organic/Simple

Layered

Modules

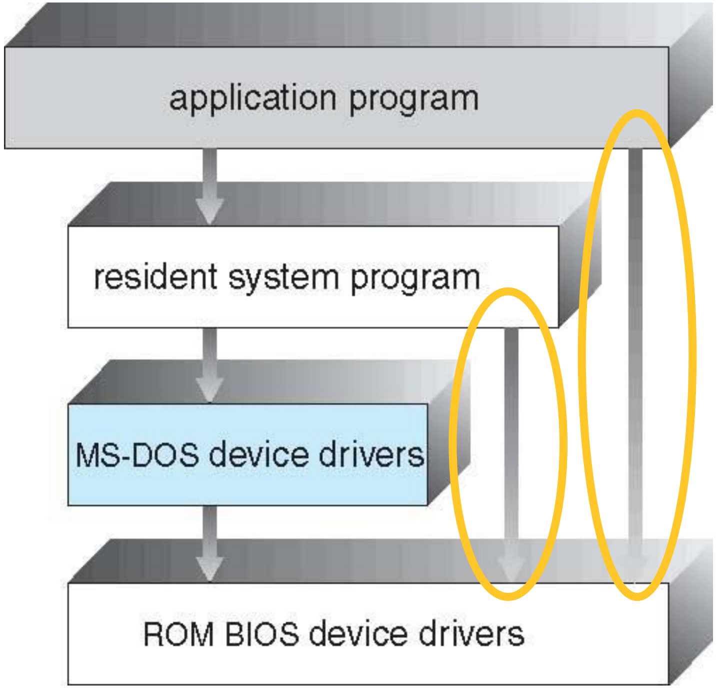
Microkernel

# SER 334

## Structure Types

## Organic/Simple

Goal:



### Pros:

- Fast Design
- Fast Implementation

### Cons:

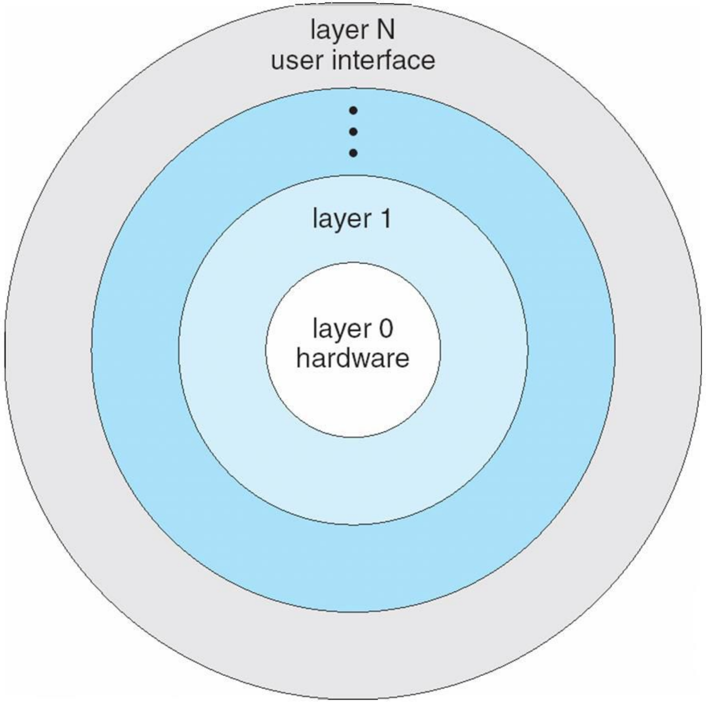
- Hard to maintain
- Unsafe
- Error-prone

**SER 334**

**Structure Types**

# Layered

Goal:



## Pros:

- Reliable
- Easy to maintain
- Portable

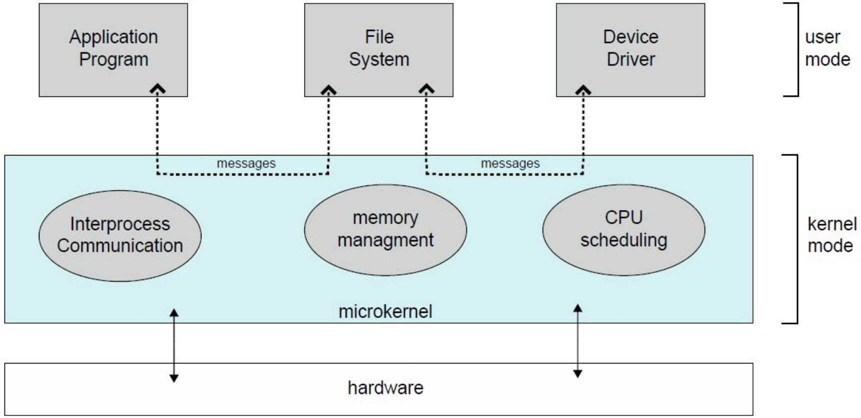
## Cons:

- SLOW Design
- SLOW speed

**SER 334**
**Structure Types**

# Microkernels

Goal:



Base Kernel only cares about:

- Message Passing
- Resource Allocation
- Plugins

## Pros:

- Kernel is small
- Fast
- Reliable
- Maintainable

## Cons:

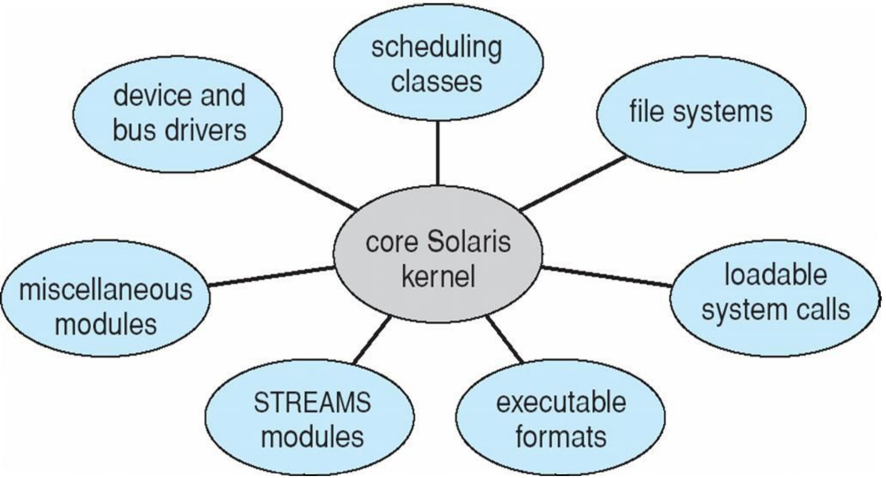
- Design Time

**SER 334**

**Structure Types**

# Modules

Goal:



What's the difference between Modules and Microkernels?

*Everyone can talk to everyone*

## Pros:

- Kernel is small
- Fastest
- Reliable
- Maintainable

## Cons:

- Unsafe

Unit 1, Question 2

# SER 334

## Sample Review

2. [Acuña] Consider the following program. It compiles without any compile-time errors in GCC, yet it contains a total of 4 issues (a combination of syntax, and logical problems). Study the program to identify all the issues. For each issue, list its type (syntactic, logical), what the problem is, and how to fix it.

```
1     #include <stdio.h>
2     int main() {
3           int input;
4           int result;
5           printf("Enter an integer number:\n");
6           scanf("d", input);
7
8           result = input % 2;
9
10          if (result = 0)
11                printf("\nNumber %d is even.", input);
12          else
13                printf("\nNumber %d is odd.", input);
14    }
```

1) Line:

Type:

2) Line:

Type:

3) Line:

Type:

4) Line:

Type:

Unit 1, Question 4

# SER 334

## Sample Review

4. [Karaliova] Consider the following declarations in C and Java. Answer the following for each declaration: 1) What data type is declared? 2) What value would we get if we attempt to access myExample[6]? 3) What value would we get if we attempt to access myExample[7]? [3 points]

```
a) C:
char myExample[] = {'s', 'e', 'r', '3','3','4'};
```

a) Data Type:

myExample[6]:

myExample[7]:

```
b) C:
char myExample[] = "ser334";
```

b) Data Type:

myExample[6]:

myExample[7]:

```
c)Java:
char[] myExample = {'s', 'e', 'r', '3','3','4'};
```

c) Data Type:

myExample[6]:

myExample[7]:

# SER 334

## Sample Review

Unit 1, Question 8

```
int a, *b, **c, d;
a = 5;
d = a + 1;
b = &a;
c = &b
// Point 1
*b = 8;
b = &d;
**c = 3;
// Point 2
 b+=7;
 // Point 3
```

| Type | int | int (deref) | int pointer | int pointer (deref) | int double pointer | int |
|---|---|---|---|---|---|---|
| Var Name | a | *b | b | *c | c | d |
| Point 1 | | | | | | |
| Point 2 | | | | | | |
| Point 3 | | | | | | |

Unit 2, Question 3

# SER 334

## Sample Review

3. [Acuña] Consider the problem of padding the following structure, and answer the three questions below. Assume that you are compiling on a system with a 32-bit architecture. [4 points total]

```
struct bmp_header {
     char creator_name[254];
     int width;
     int height;
     char signature_rgb[2];
     int offset_pixels;
};
```

(a) What is the size of this struct as defined?

(b) How much space would be wasted with word length padding?

(c) [Katie] Redefine the structure to reduce the wasted space.

Unit 2, Question 6

# SER 334

## Sample Review

6. [Acuña] Consider the following function which adds a new node to the front of a list passed as a parameter called param_list.

```
struct grade_node {
    int value;
    char assignment[255];
    struct grade_node* next;

};

void add_node(grade_node* param_list, grade_node* node) {
    if(node != NULL) {
      node->next = param_list;
      param_list = node;
   }
}
```

Is it possible for this function ever to work incorrectly? If so, under what conditions does it fail?

Unit 3, Question 4

# SER 334

## Sample Review

4. [Acuña] Consider the following file storage scenarios:

(a) A word processing file that stores text, formatting information, tables, and images.

(b) A configuration file that is meant to be edited by advance users.

(c) An executable program file that displays a class schedule.

Which of these should be implemented using a plain text file or a binary file format? Explain.

a) Data Type:

Reason:

a) Data Type:

Reason:

a) Data Type:

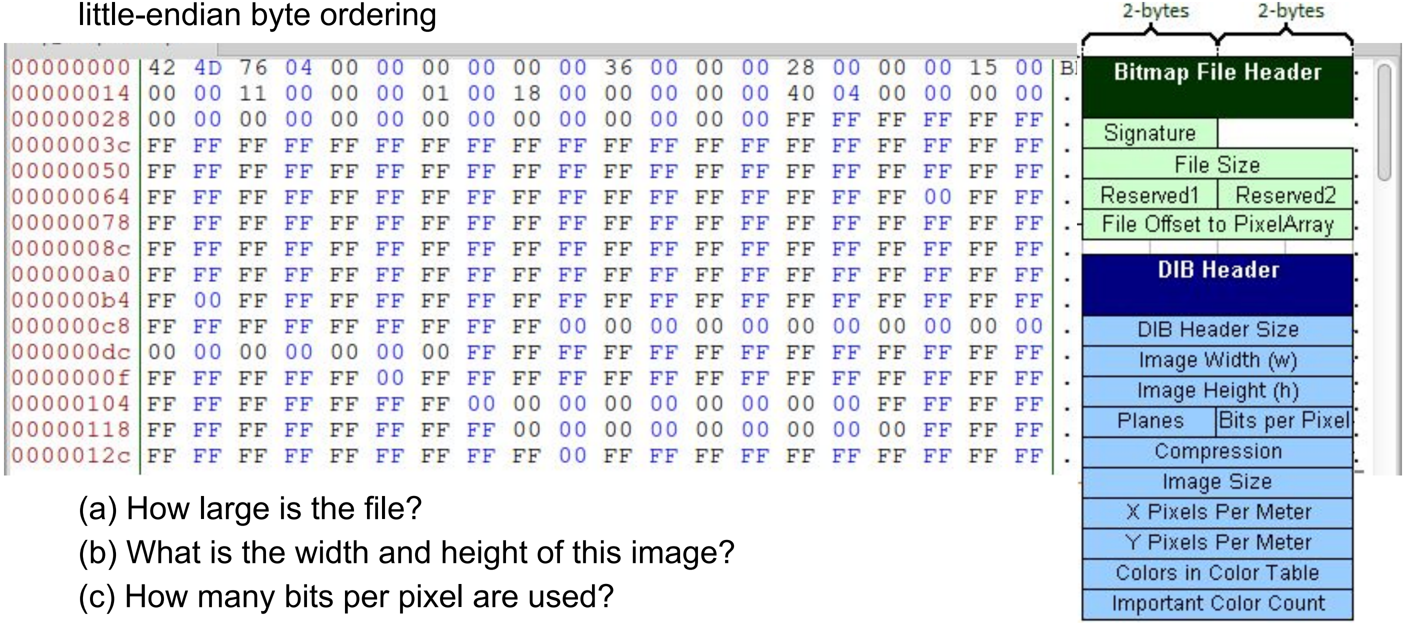
Reason:

Unit 3, Question 4

# SER 334

## Sample Review

5. [Acuña] Shown below is a valid BMP file that has been opened in a hex editor. Based on the data visible, answer the following questions. Indicate which number base you use for each question. (The complete specification is shown in the appendix.) Note that this screen shot comes from an Intel architecture system where numbers are stored with little-endian byte ordering



```
00000000 42 4D 76 04 00 00 00 00 00 00 36 00 00 00 28 00 00 00 15 00 BI
00000014 00 00 11 00 00 00 01 00 18 00 00 00 00 00 40 04 00 00 00 00 .
00000028 00 00 00 00 00 00 00 00 00 00 00 00 00 00 FF FF FF FF FF FF .
0000003c FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .
00000050 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .
00000064 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF 00 FF FF .
00000078 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .
0000008c FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .
000000a0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .
000000b4 FF 00 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .
000000c8 FF FF FF FF FF FF FF FF FF 00 00 00 00 00 00 00 00 00 00 00 .
000000dc 00 00 00 00 00 00 00 FF FF FF FF FF FF FF FF FF FF FF FF FF .
0000000f FF FF FF FF FF 00 FF FF FF FF FF FF FF FF FF FF FF FF FF FF .
00000104 FF FF FF FF FF FF FF 00 00 00 00 00 00 00 00 00 FF FF FF FF .
00000118 FF FF FF FF FF FF FF FF 00 00 00 00 00 00 00 00 00 FF FF FF .
0000012c FF FF FF FF FF FF FF FF FF 00 FF FF FF FF FF FF FF FF FF FF .
```

a)

b)

c)

d)

(a) How large is the file?

(b) What is the width and height of this image?

(c) How many bits per pixel are used?

(d) Using b and c, how many bytes are required for each row? How many are for padding?

Unit 4, Question 3

# SER 334

## Sample Review

3. [Karaliova] For most programming languages, the run-time support library provides a system call interface that translates function calls into corresponding system calls. Why do we need an intermediate between the two instead of invoking system calls from a program directly? Explain.

Unit 4, Question 7

# SER 334

## Sample Review

7. [Lisonbee] Compare and contrast the organic and microkernel operating system structures. Explain how the efficiency (speed), security, and maintainability of each compare, and why.

## Upcoming Events

# SI Sessions:

- Sunday, January 28th at 7:00 pm MST **Cancelled - good luck on Exam 1!**
- Monday, January 29th at 7:00 pm MST
- Thursday, February 1st at 7:00 pm MST

# Review Sessions:

- Exam 1 Review: Thursday, January 25th 7:00 pm - 9:00 pm MST

# Questions?

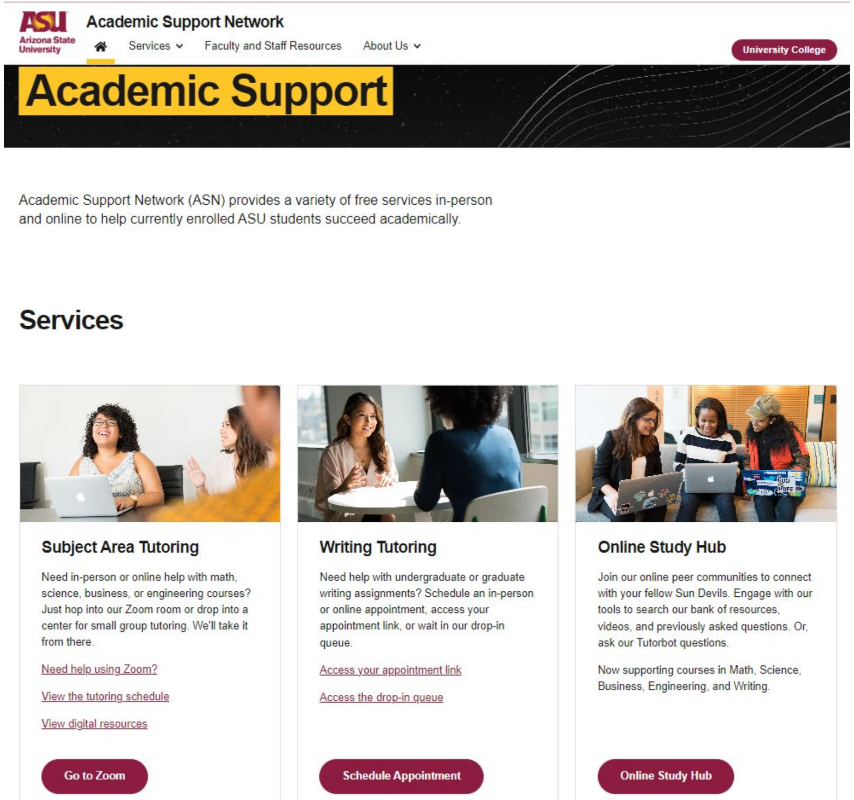
# Survey:

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

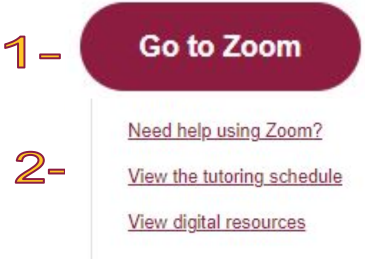
Schedule Appointment

### 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

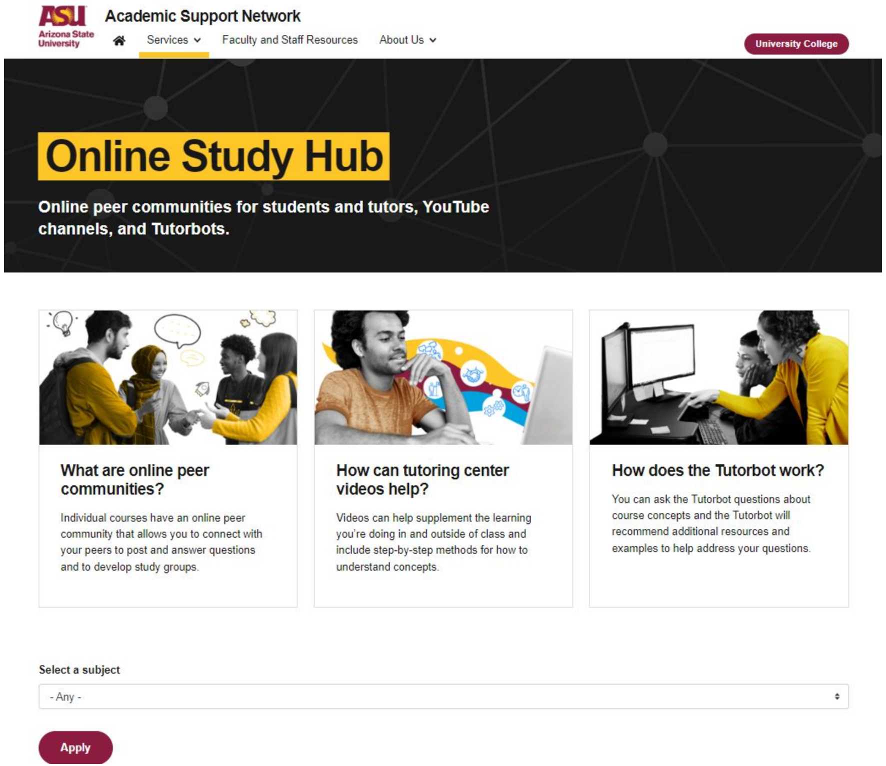


1. Click on ‘Go to Zoom’ to log onto our Online Tutoring Center.
2. Click on ‘View the tutoring schedule’ to see when tutors are available for specific courses.

# More Questions?

## Check out our other resources!

**tutoring.asu.edu/online-study-hub**



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?

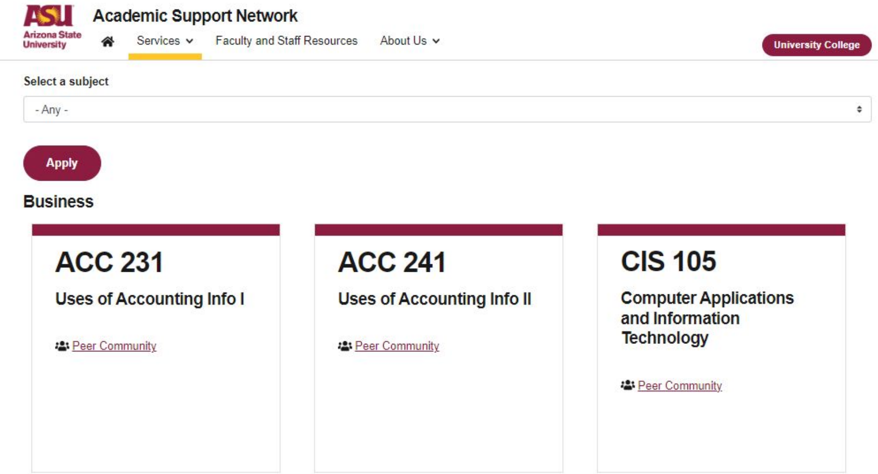
Videos can help supplement the learning you're doing in and outside of class and include step-by-step methods for how to understand concepts.

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 -



Select a subject

- Any -

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