Pomona College

Computer Science

Fall 2024: CSCI 181RT
Real-Time Systems in the Real World

Lecture 13

Tuesday, October 8, 2024
Edmunds Hall 105
2:45 PM - 4:00 PM

Professor Jennifer DesCombes

Pomona College

Agenda

- Go Backs
- Announcements
- Discussion on Reading
- Use of Semaphores (Part 3) +
- In Class Assignment
- Lab Overview
- Look Ahead
- Assignment
- Action Items

Go Backs

- General?
- Action Item Status
  - AI240910-2: Find recommended book on computer architecture.
  - AI240924-1: At what point as a development team grows does it make sense to have dedicated software and integration testers?
  - AI240924-2: Is there a limit on the size of an Agile development effort before it becomes less efficient than other development approaches?
    - Have information to Discuss - OK to Close after discussion?
  - AI240924-3: Are ‘C’ type Handles (pointer to a pointer) similar in concept to Java Script Handles

Announcements

- Fall Break!
  - No Lecture Tuesday, October 15th
  - Lab May Start Late on Wednesday, October 16th
- Reminder
  - No Lecture Tuesday, November 26th

Discussion on Reading

- The Mythical Man Month
  - Chapter 10, 11 & 12: The Documentary Hypothesis, Plan to Throw One Away, Sharp Tools

Use of Semaphores - Part 3

- Synchronization of Tasks ✓
- Protection of Shared Devices/Services ✓
- Controlled Processing of Input Data ✓
- Protection of Non-reentrant Code
- Guarantee Completion of Specific Operations

Pomona College

Use of Semaphores - Protection of Non-reentrant Code

```
sem_t codeScarrySem;
uInt32 scarryCode( *uInt32 );

.
.
.
  int myRet;
  myRet = sem_wait( &codeScarrySem );
  uInt32 scarryResult;
  scarryResult = scarryCode( &someGlobal);
  myRet = sem_post( &codeScarrySem );
.
.
.
```

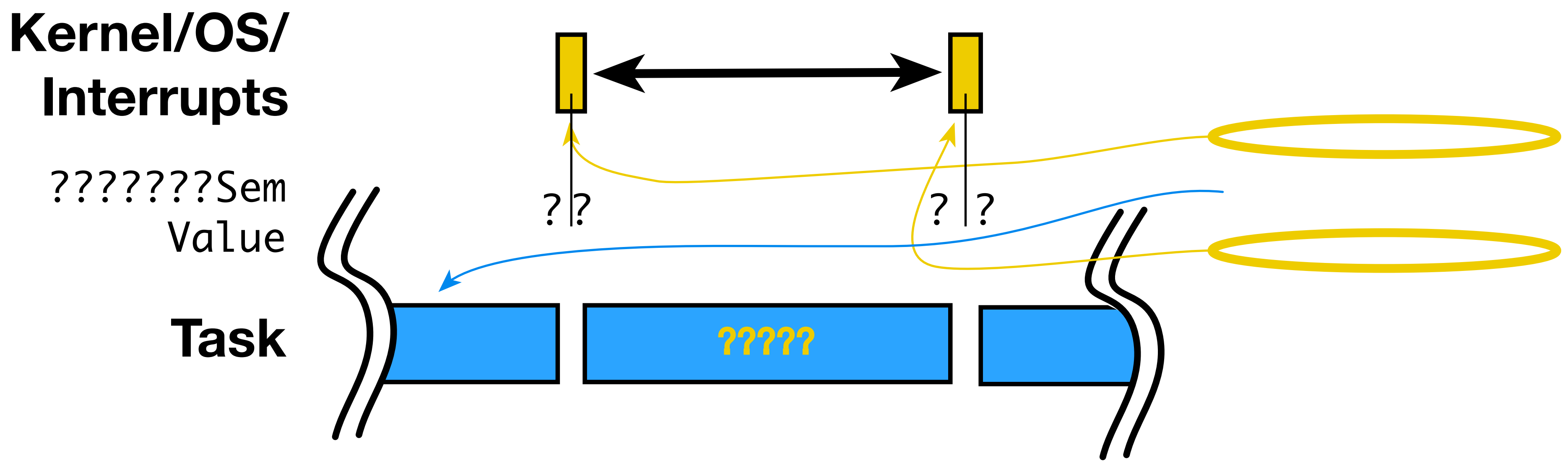
If no other task is currently executing the scarryCode method, execution will continue.

If the scarryCode is currently being executed by some other task, execution will be suspended (waiting) the other task is done with scarryCode. Code will resume execution when UART is no longer being used by other code/task.

Let the system/tasks know that the scarryCode is now available

Use of Semaphores - Protection of Non-reentrant Code

What is going on here?



Pomona College

Use of Semaphores - Completion of Operations

```
sem_t accessGVectorSem;
vectorFloat32Struct gfRateVector;
.
.
.
   int myRet;
   myRet = sem_wait( &accessGVectorSem );
   gfRateVector.x = 30.45;
   gfRateVector.y = 0.003;
   gfRateVector.z = 10.62;
  myRet = sem_post( &accessGVectorSem );
.
.
.
```

```
.
.
.
  // Set gfRateVector to zero - OK??
  gfRateVector = cfZeroRateVector;

.
  // Set gfRateVector to zero - OK??
  int myRet;
  myRet = sem_wait( &accessGVectorSem );
  gfRateVector = cfZeroRateVector;
  myRet = sem_post( &accessGVectorSem );
.
.
.
```

Are these both OK?

Pomona College

Use of Semaphores - Completion of Operations

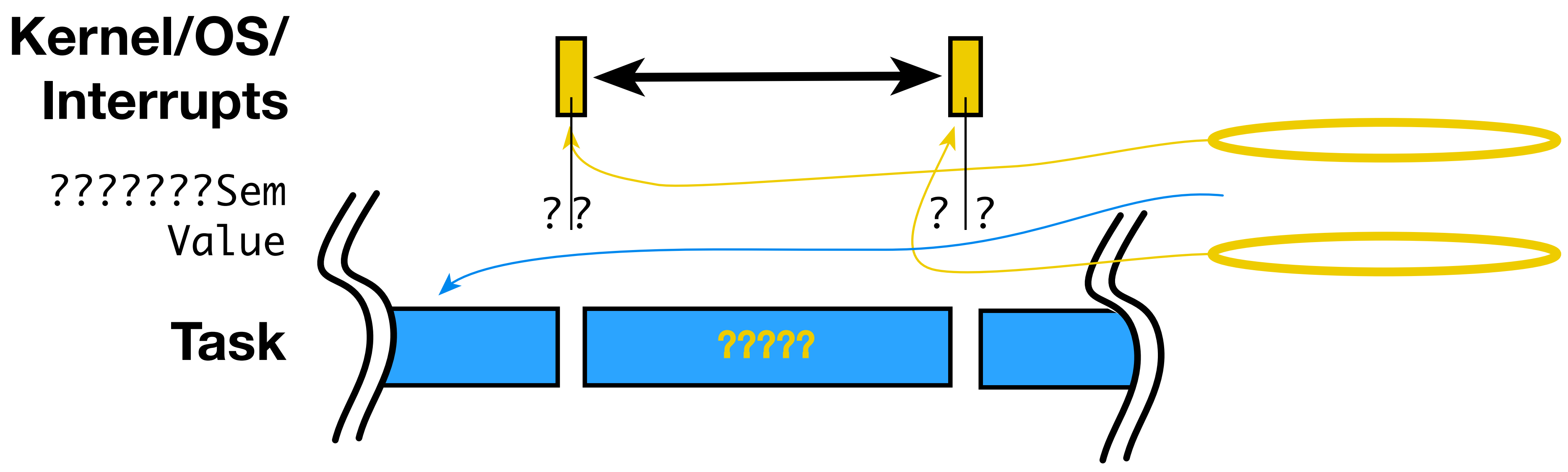
```
sem_t accessGVectorSem;
vectorFloat32Struct gfRateVector;
.
.
.
   // Set gfRateVector to zero - OK??
  gfRateVector = cfZeroRateVector;
        mov (cfZeroRateVector+0) -> (gfRateVector+0)
        mov (cfZeroRateVector+4) -> (gfRateVector+4)
        mov (cfZeroRateVector+8) -> (gfRateVector+8)
.
.
```

Can the Processor be Interrupted During These Pseudo-assembly Instructions?

If So, What Does This Do to the Data Integrity?

Use of Semaphores - Completion of Operations

What is going on here?



Pomona College

Use of Semaphores - Mutex and Locks

- Mutual Exclusion Semaphores - Mutex
  Mutual Exclusion semaphores are used to protect shared resources (data structure, file, etc..).
- A Mutex semaphore is "owned" by the task that takes it. If Task B attempts to semGive a mutex currently held by Task A, Task B's call will return an error and fail.
- Similar Functionality to Binary Semaphores
- Mutex May Not Provide Suspend/Resume Feature - Will be System Dependent

```
Thread A
   Take Mutex
      access data
      ...
      …
   Give Mutex
```

Use of Semaphores - Build Your Own!

- Three (3) Tasks
  - Task Priority 1, 10ms Interval, ~1ms Processing Time
  - Task Priority 2, 25ms Interval, ~ 8ms Processing Time
  - Task Priority 3, 3ms Interval , ~ 1ms Processing Time
  - No Dummy Task
- Draw Timeline For When Time = 50ms
  - Task 1 and Task 2 Timers Expire
- Draw Timeline For When Time = 75ms
  - Task 2 and Task 3 Timers Expire
- Draw Timeline For When Time = 150ms
  - Task 1, Task 2 and Task 3 Timers Expire

Lab 6 Overview

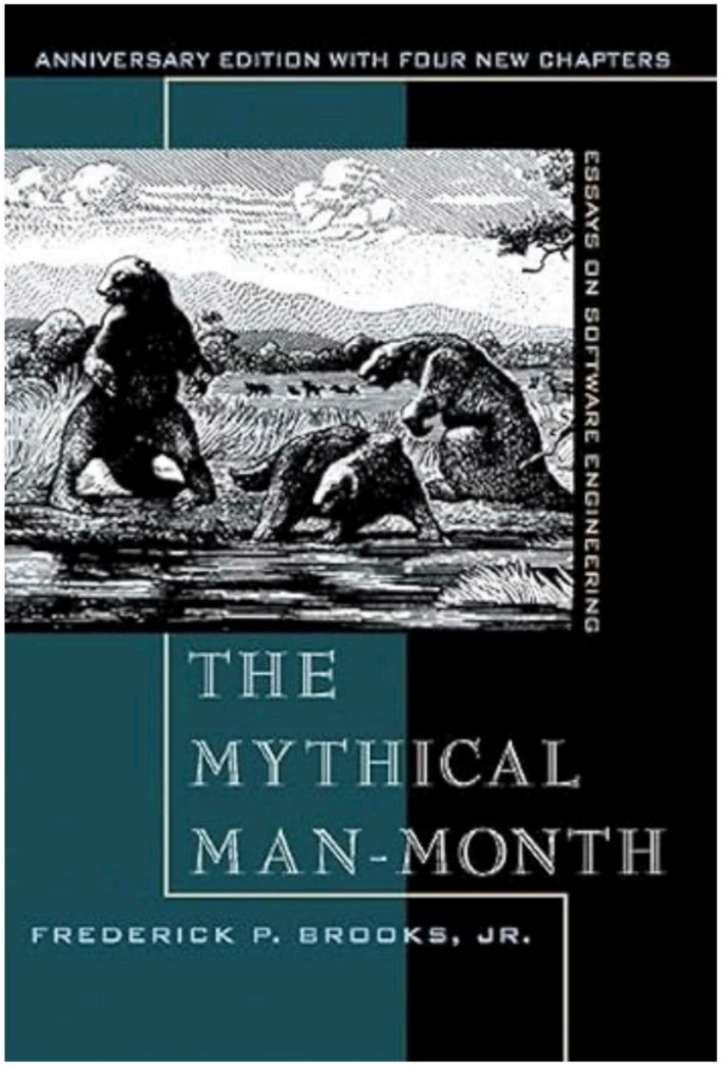
- Discussion of Sampling Intervals and Data Rate
- Code to Read an External Signal
- Evaluation with TE

Look Ahead

- Review of Reading
- Review of Lab 6
- Interrupts and OS Support

Assignment - Readings

- The Mythical Man Month
  - Chapter 13, 14 & 15: The Whole and the Parts, Hatching a Catastrophe, and The Other Face
  - Send Me Discussion Topics by 10:00 AM on Thursday, Oct. 10, 2024.


ANNIVERSARY EDITION WITH FOUR NEW CHAPTERS
ESSAYS ON SOFTWARE ENGINEERING
THE MYTHICAL MAN-MONTH
FREDERICK P. BROOKS, JR.

Pomona College

Action Items and Discussion

AI#:	Owner	Slide #	Document	Action