Deadlock I

Tom Kelliher, CS 311

Mar. 30, 2012

1 Administrivia

Announcements

Assignment

Finish reading Chapter 7.

From Last Time

Classic synchronization problems.

Outline

1. System model.
2. Deadlock examples.
3. Necessary conditions for deadlock.
4. Resource allocation graphs.
5. Dealing with deadlocks.
6. Deadlock prevention.

Coming Up

Deadlock II.

2 System Model

1. Processes competing for resources. Process model:

   ```
   while (1)
   {
      request some resources;
      compute;
      release some resources;
   }
   ```

2. Resource: *anything* a process may block on. Examples:

   - Semaphore.
   - Device (disk read/write, printer, etc.)
   - Memory.
   - CPU.
   - File.

   Two types of resources:

   (a) Serially reusable resources: printer, tape drive, CPU, etc.

   (b) Consumable resources: semaphores, messages, etc.

3 Deadlock Examples

3.1 Semaphores

```
semaphore A(0), B(0);

process1()
{
   A.wait();
   B.signal();
}

process2()
{
   B.wait();
   A.signal();
}
```

Process 1 is waiting for process 2 is waiting for process 1...

3.2 Tape Drives

System has five tape drives.

Process	**Allocated**	**Request**
P1	2	2
P2	2	3
P3	1	1

Resource allocation graph.

4 Necessary Conditions for Deadlock

1. Mutual exclusion — A resource is not sharable.

2. Hold and wait — A process is allowed to hold a resource while it's waiting for other resources.

3. No preemption — A process cannot be forced to give up a resource.

4. Circular wait — P1 is blocked because of P2; P2 is blocked because of P3; ...; Pn is blocked because of P1.

Why aren't these *sufficient* conditions?

How does deadlock differ from starvation?

5 Resource Allocation Graphs

Pretty simple:

1. Vertices:
   - Processes.
   - Collections of resources of same type.
2. Edges:
   - Allocation.
   - Request.

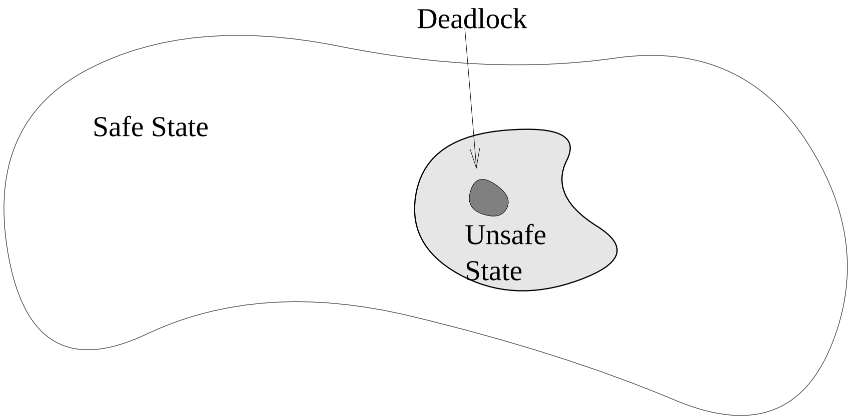
6 Dealing with Deadlocks

1. Ignorance. More common than you'd think.

2. Detection and recovery.

3. Prevention — Prevent one of the four necessary conditions.

4. Avoidance — Manage resources so that a deadlock never occurs:



7 Deadlock Prevention

Removal of one of the four necessary conditions.

7.1 Mutual Exclusion

1. Share the resource.
2. Example?
3. Limitations?

7.2 Hold and Wait

1. A process must release all resources before requesting more.
2. Example?
3. Limitations? (Request *everything* at once or wait a lot.)

7.3 No Preemption

1. Kernel forcibly reclaims resources.

2. Example?

3. Limitations?

7.4 Circular Waiting

1. Impose a *total order* on all resources.

2. Resource i may only be requested if process holds no resource j with $j > i$.

3. Example?

4. Limitations:

    - Determining the order.

    - How does an "out of order" process proceed?

7.5 Usefulness of Prevention

- Works for specific resources.

- No one mechanism is generally applicable.