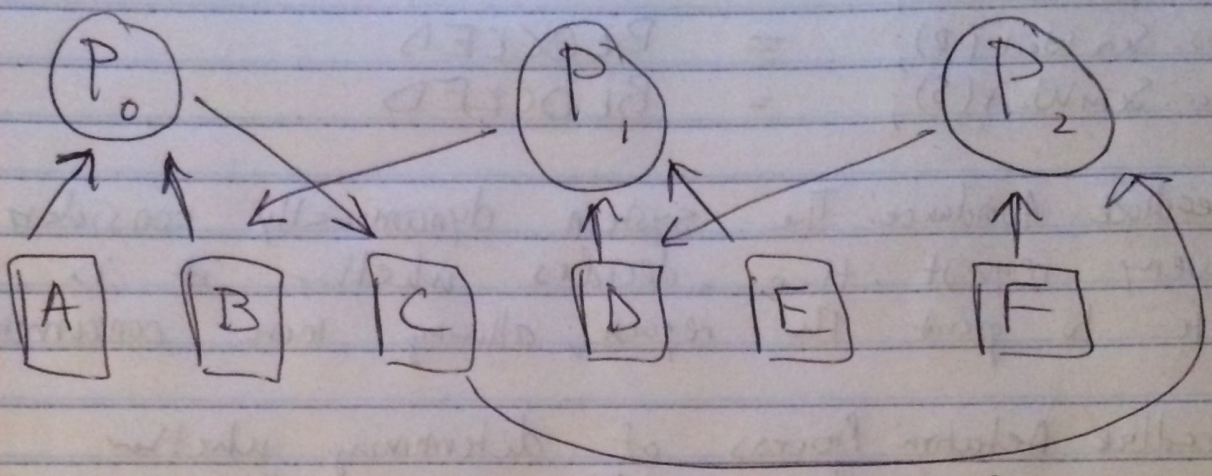
RayShawn Ware
COMP 3500 HW 2
10/12/15

①

a)



P_0 is waiting for P_2 which is waiting for P_1 which is waiting for P_0. Cycle waiting creates a deadlock.

b)

```
Void P0()
{
 while(true){
   get(B);
   get(C);
   get(A);
 //
 //
   release(A);
   release(B);
   release(C);
 }
}
```

```
Void P1()
{
 //
   get(D);
   get(E);
   get(B);
 //
 //
   release(D);
   release(E);
   release(B);
 }
}
```

```
Void P2()
{
 //
   get(F);
   get(D);
   get(C);
 //
 //
   release(C);
   release(F);
   release(D);
 }
}
```

(2) Yes

Foo: SemWait(S);
Bar: SemWait(R);
Foo: SemWait(R); = BLOCKED
Bar: SemWait(S); = BLOCKED

(3) Deadlock Avoidance: The system dynamically considers every request, then decides whether it is safe to grant the request; allowing more concurrency.

Deadlock Detection: Process of determining whether or not a deadlock exists. Must check allocation against the availability of the resource for all possible allocation sequences to determine if the system is in a deadlocked state.

Deadlock Prevention: Prevent deadlocks from occuring by constraining how requests can be made and how they are handled.

(4) Only time the system is deadlocked is when a process can not access the maximum number of resources it needs. One process must be able to obtain two resources. No more resources are required for this process, thus it will return its resources when its complete.