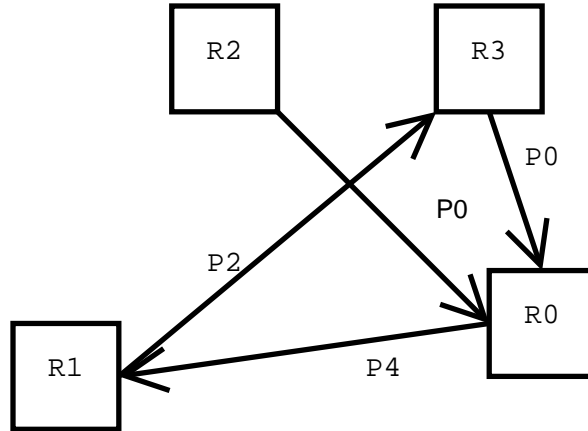


4. (15 pts.) Deadlock

- (a) What does the following wait-for graph tell you about whether there is deadlock and what processes are involved?



- (b) Consider programs which request resources A and B . How would you avoid deadlock by allowing preemption?

- (c) Consider programs which request resources one at a time always in the order A, C, B . Can this system deadlock? Explain.

- (d) How would you use the Banker's algorithm to detect deadlock?

8. (10 pts.) Processes.

(a) What is finite progress, and how is it ensured in an operating system?

(b) What mechanisms are used to prevent one process from attacking another process?

(c) What tasks must be done by the operating system kernel for the process abstraction?

9. (8pts.) Two-phase locking.

(a) List the steps (including conditions) necessary to obtain a read lock.

(b) List the steps (including conditions) necessary to obtain a write lock on a location which is not currently locked.

(c) List the steps (including conditions) necessary to upgrade a lock on a location for which the upgrade can be performed.

10. (7pts.) Scheduling.

(a) When and from where does a process get moved to the ready state?

(b) When and from where does a process get moved to the run state?
