1. Answer:

a)

Mutual exclusion: No two jobs can simultaneously write data to the same location in the spooling file.
Hold and Wait: Partially spooled jobs remain in the spooling file until more space is available.
No preemption: Jobs cannot remove other jobs from the spooling file.
Circular Wait: When the spooling file is full, each job waits for all the other jobs to free up space.

For example, if the spooling file is, say, 10 MB and the first half of ten 2-MB jobs arrive, the disk will be full. Assuming that jobs cannot start printing until they are fully spooled, deadlock is possible.

b)

No-preemption. The spooling file is preemptable if a user is allowed to delete an incomplete file and send it later.

2. Answer:

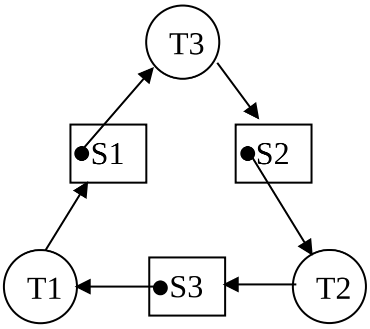
a) The system is deadlock free. Suppose that each process has one resource. There is one resource free. Either process can ask for it and get it, in which case it can finish and release both resources. Consequently, deadlock is impossible.

b) If a process has *m* resources, it can finish and cannot be involved in a deadlock. Therefore, the worst case is where every process has $m - 1$ resources and needs another one. If there is one resource left over, one process can finish and release all its resources, letting the rest finish too. Therefore the condition for avoiding deadlock is $r \geq p(m - 1) + 1$.

3. Answer:

a) Deadlock occurs when all processes progress to their second `semwait()`.

b) This resource allocation graph shows the deadlock state (i.e., a circular wait).



c) No deadlock will occur if T1 is changed.

Self-test

1. D
2. B
3. C
4. D
5. B & C
6. B