15 POINTS

2. Consider the following resource-allocation policy for a fixed inventory of **serially reusable** resources of three different types (such as tape drives, printers, shared memory, etc.):

- Requests and releases of resources are allowed at any time.
- If a request for resources cannot be satisfied because resources are not available, then we check any processes that are blocked, waiting for additional resources:
- If they have the desired resources, then these resources **are taken away** from those blocked processes and are **given** to the requesting process.
- The inventory of resources that the requesting process is collecting is **increased** to include the resources that were taken away from the blocked process.
- If the requesting process is still not able to collect the resources it needs to continue to run, it will be **forced to block and wait for what it needs**. Once it blocks, its current resource inventory will become **vulnerable to other running processes looking for additional resources.**
- Whenever a resource is freed, blocked process needing such resource are made ready, and when they are dispatched to the run state they will have a chance to again try and secure their necessary inventory of resources so that they can continue to run.

For example, consider a system with **three resource types** and a **total inventory** of (4,2,2). If process **A** asks for (2,2,1) it gets them. If process **B** asks for (1,0,1), it also gets them. Then, if process **A** asks for a further allocation of (0,0,1), it is blocked (resource not available). If process **C** now asks for (2,0,0), it gets the available one from the systems (1,0,0) inventory and the one which was allocated to process **A** (since process **A** is blocked). Process **A's** allocation goes down to (1,2,1) and its need (outstanding request) goes up to (1,0,1).

Problem 2 is continued on next page