(a) The general *Resource Reservation* problem can be restated as follows. We have a set of m resources, and n processes, each of which requests a subset of the resources. The problem is to decide if there is a set of k processes whose requested resource sets are disjoint.

We first show the problem is in NP. To see this, notice that if we are given a set of k processes, we can check in polynomial time that no resource is requested by more than one of them.

To prove that the *Resource Reservation* problem in NP-complete we use the independent set problem, which is known to be NP-complete. We show that

Independent Set $\leq_P$ *Resource Reservation*

Given an instance of the independent set problem — specified by a graph G and a number k — we create an equivalent resource reservation problem. The resources are the edges, the processes correspond to the nodes of the graph, and the process corresponding to node v requests the resources corresponding to the edges incident on v. Note that this reduction takes polynomial time to compute. We need to show two things to see that the resource reservation problem we created is indeed equivalent.

First, if there are k processes whose requested resources are disjoint, then the k nodes corresponding to these processes form an independent set. This is true as any edge between these nodes would be a resource that they both request.

If there is an independent set of size k, then the k processes corresponding to these nodes form a set of k processes that request disjoint sets of resources.

(b) The case $k = 2$ can be solved by brute force: we just try all $O(n^2)$ pairs of processes, and we see whether any pair has disjoint resource requirements. This is a polynomial-time solution.

(c) This is just the Bipartite Matching Problem. We define a node for each person and each piece of equipment, and each process is an edge joining the person and the piece of equipment it needs. A set of processes with disjoint resource requirements is then a set of edges with disjoint ends — in other words, it is a matching in this bipartite graph. Hence we simply check whether there is a matching of size at least k.

(d) We observe that our reduction in (a) actually created an instance of *Resource Reservation* that had this special form. (Each edge/resource in the reduction was requested only by the two nodes/processes that formed its ends.) Thus, even this special case is NP-complete.

[1] ex588.290.312