國立中央大學九十二學年度碩士班考試入學招生試題卷

系所別: 資訊工程學系 網路學習科技研究所 科目: 系統程式

1. Consider a system consisting of n resources of the same type, being shared by m processes. Resources can be requested and released by processes only one at a time. Whether the system is dead free or not if given the following conditions:
   (a)The maximum need of each process is between 1 and n resources. (b) The sum of all maximum needs is less than n+m.
   Please prove it otherwise you will get zero score. (20%)
2. Suppose that we have already written a two-pass assembler that allows literals. In general, the current version of this assembler always places the literal pool at the end of the assembly. In some cases, however, it is desirable to place literals into a pool at some other location in the object program. To allow this, we add the assembler directive LTBEGIN. This literal pool is placed in the object program at the location where the LTBEGIN directive was encountered. Describe the data structures needed and implement LTBEGIN in a two-pass assembler. (20%)
3. A file is to be shared among different processes, each of which has a unique number. The file can be accessed simultaneously by several processes, subject to the following constraint: The sum of all unique numbers associated with all the processes currently accessing the file must be less than n. Write a monitor coordinate access to the file. The monitor consists of the "acquire" and "leave" procedures. (10%)
4. Term discrimination: (a) literal and immediate value (b) control section and program block (c) dynamic linking and linkage editor. (10%)
5. Could a one-pass assembler produce a relocatable object program and handle external references? Describe the processing logic that would be involved and identify any potential difficulties. (10%)
6. Explain the functions of a loader and how it works. (10%)
7. Discuss the reasons why the 1-pass macro processor described in the textbook cannot handle recursive macro expansion. How would you modify the algorithm to solve this problem? (10%)
8. Describe how recursive decent parsing works. What's the limitation? (10%)