List Processing Handout

Here is how to do the List Processing Algorithm.

[1] Before you start, you must first have a Priority List and an order-requirement digraph. In some problems you will be given the priority list, in others you will be told to use a rule like the decreasing time rule (i.e., longer tasks have higher priority) to get your priority list.

[2] Keep a Ready List. Tasks go on the ready list before you put them onto the schedule. As soon as a processor becomes free, update your ready list by checking to see what tasks are now ready to be worked on. When a task becomes ready, you put it on your ready list. Cross a task off from the ready list and from the priority list when you put it onto the schedule.

[3] To make your schedule, check your ready list. If there’s only one task there, cross it off and put it on your schedule. If there’s more than one there, check your priority list to see which one to put on your schedule first. If your ready list has nothing on it, schedule dead time until a task can be put on your ready list.

[4] When a task is done, cross it off on your digraph.

Priority List: T_2 T_1 T_3 T_4 T_5 T_6

Ready List:

Processor 1

Processor 2

Priority List: (use decreasing time)

Ready List:

Processor 1

Processor 2

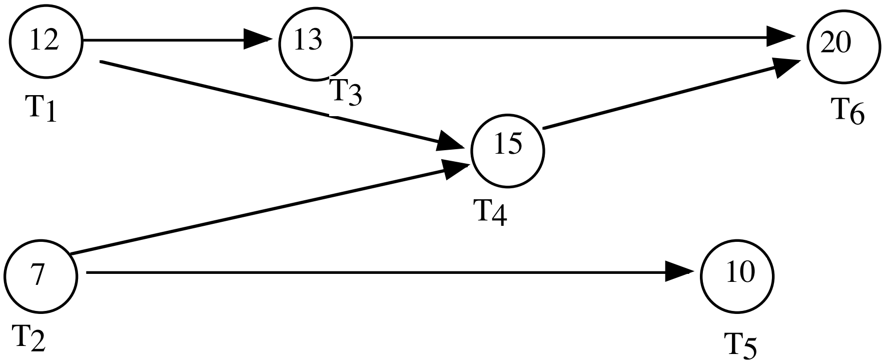
Priority List:

Ready List:

Processor 1

Processor 2

Processor 3


12
T1
13
T3
20
T6
15
T4
7
T2
10
T5