CS 140 MP1: Iron Chef

Deadline: (Fri) Feb 14 2014 11:59pm

Problem Description

"Welcome to Kitchen Stadium…" goes the opening lines to Iron Chef, an extreme cooking contest where only the best in cooking multitasking are able to survive. You are to create an Iron Chef simulator. Your program will be given a list of recipes and the order the food is to be prepared, will output which dish should be on the stove during each minute of time of the contest.

Specifications

Your machine problem will accept as input a **recipe file** and a cooking **task list file**. The recipe file contains instructions on how to cook a particular dish while the task list file contains the order in which the dishes will be cooked.

A. The Cooking Task List file

The cooking task list file (stored as **tasklist.txt**) indicates the order at which dish to start cooking. The following is a sample tasklist file

```
tasklist.txt

adobo 1
tinola 5
karekare 8
adobo 14
```

Each line of tasklist file consists of a **single word** indicating what dish to start cooking at that particular instant of time. This task file says that at adobo will be cooked at time 1, tinola at time 5 etc. When a dish is scheduled to be cooked, your program loads the appropriate recipe file. Adobo.txt for adobo, tinola.txt for tinola, etc.

B. The Recipe file

A recipe file is a file that contains instructions on how to cook a particular dish. The following is a sample recipe file.

```
karekare.txt

karekare 3
prep 10
cook 30
prep 5
cook 10
```

The first line is the name of the dish followed by its **priority** (1-10, 10 highest)
Each line of the recipe file contains one word that describes and **action** followed by an **integer** indicating how long the action takes. In our example, prep 10 means karekare will spend 10 mins on the preparation followed by 30 mins of cooking, etc…

The steps will be either a preparation step or a cooking step. All the preparation steps are handled by your assistants. Preparation steps from different dishes can be done simultaneously by your assistants. There is no limit in the amount of dishes the assistants can handle.

The cooking step on the other hand can only be done in the stove. Only a single dish can be done by the stove at a given time.

C. The Scheduler

Your task is to design and implement a scheduler that will determine which dish should be on the stove at a specific time. Your scheduler should choose from the dishes that are in line for the stove and the dish currently at the stove.

Context Switching
When changing from one dish to another, leave the stove empty for 1 min to clean the pan (CS = 1min). If the stove is already empty, you can use the stove immediately. Assistants can work while you clean the pan (during CS)

Design Considerations
When designing your scheduler, take note of the following considerations.

- Total amount of time to finish all the dishes
- Stove utilization
- Waiting time of the dishes

D. Output File

Your program should produce an output file that will contain the process execution table in csv format.

Time	Stove	Ready	Assistants	Remarks
1	adobo(cook=5)	none	none	adobo arrives
2	adobo(cook=4)	none	none	none
3	adobo(cook=3)	soup(cook=5)	none	soup arrives
4	adobo(cook=2)	soup (cook=5)	karekare(prep=5)	karekare arrives

5	adobo(cook=1)	soup (cook=5)	karekare(prep=4)
			tinola(prep=3)
				tinola arrives

6	empty	soup (cook=5)	karekare(prep=3)
			tinola(prep=2)
			adobo(prep=6)
				adobo cook done
				Still cleaning the pan…
				Next is soup

7	soup(cook=5)	none	karekare(prep=2)
			tinola(prep=1)
			adobo(prep=5)
				none

8	soup(cook=4)	tinola(cook=9)	karekare(prep=1)
			adobo(prep=4)
				tinola mix done

You also include the performance statistics of the simulation

- Total amount of time to finish all the dishes
- Stove utilization
- Weighted average waiting time of all the dishes
  - (∑ waiting time x priority) / (total weight x number of dishes)

FAQs

- Design
  - The stove is initially empty
  - You can use multiple queues in your design
  - You are not limited to the scheduling schemes discussed in class

    - Simulation starts at time 1
- Tasklist file
    - A dish can appear only once
    - You can have at least 1 dish in the task list
    - You cannot know the dishes that will come in advance
        - You will not know that karekare will arrive at time=4 during time=1
    - The time is an integer from 1
- Recipes
    - Each recipe must contain at least 1 step
    - Each step has a non zero duration
    - The first step can be a cook step or a preparation step
    - Each recipe alternates from preparation step and cooking step
    - Once the recipe is loaded, you can examine the duration of all the steps
-

Deliverables

1. Source code
2. Documentation
    a. Group Member Names
    b. Introduction
        i. Goals and objectives(the goal of the design)
        ii. Major constraints (assumptions and limitations)
    c. Algorithm Design
        i. Data structures used
        ii. Pseudocode on how the selection/scheduling is made
    d. Algorithm Performance
        i. Best-case performance
        ii. Worst-case performance
3. Team Evaluation
    a. Your major contributions
    b. Division of work
    c. Evaluation of every member

Grading

- Originality of algorithm design 30%
- Documentation (design, code) 15%
- Correctness of implementation 45%
    – (no bugs from your part of the code =))
- Teamwork 10%
    – maximum of 4 members only

Submission

- Deadline: Feb 14, 2014 (11:59 pm)
- Email:ISubmitMyCode@gmail.com
- Subject: CS 140 MP
- Contents:
    – Source code
    – Documentation
    – Team evaluation (in a separate email)

Final Notes

This is meant as a group exercise. You may discuss the MP with other groups, but you are expected to come up with your own **original solution** to the problem. Source codes will be compared.
A **demo** of your implementation will be scheduled. During this demo all members of the group should be present and be ready to answer questions regarding the design of the algorithm and its implementation. Points will be deducted for late submissions (5% for every working day late, i.e., not including Saturdays, Sundays, holidays, and class suspensions). Submission of the MP (even if it will get a grade of zero due to deductions) is strictly required for the course.

Do not assume anything for this MP. Maraming namamatay sa maling akala. If you aren’t sure about something, feel free to ask the teacher. If you have any questions, please write an

- email directly to me with subject *CS 140 MP1 Question* to jbdelmundo@gmail.com
- post a question in facebook group
- send a message in facebook