



**HACETTEPE UNIVERSITY
ENGINEERING DEPARTMENT
COMPUTER ENGINEERING**

LESSON
BBM 203 PROGRAMING LABORATORY

ADVISOR
Dr. Sevil ŞEN, Dr. Burcu CAN, Dr. Adnan ÖZSOY Dr. Ahmet Selman
BOZKIR

EXPERIMENT 3
Car Production Line Simulation

NAME : Taha BAŞKAK - 21228104

This homework is also required to make a factory simulation. Generally speaking, the parts of the factory related to each other, the vehicles to be machined at the factory and the instantaneous status of the cars. This simulation works in real time.

Factory departments, cars and the reporting system have their own characteristics. There can be more than one from the same department in the factory departments, and each section has a time slot in which the running car must remain. Apart from these, the chapter number contains features that are free. The car's information includes model, special number, system entry time characteristics. There are 3 types of reporting in the reporting system. These are reporting with the car's special number, reporting all vehicles, reporting sections. In the reporting content, the special number of the car and the time, duration, and percentage of completion of the car in the factory section, which indicates the duration of processing of each car in each section and where the car has been completed, is specified. In the factory department reporting system, the content includes the writing of each part with its special number and model if the car is carved in it, and the writing of the special numbers and models of other vehicles previously completed in that part.

In the algorithm I created while creating this simulation, I created four different single linked lists. These are the "carReport" list, which holds the "carList" list that holds the car properties, the "factoryStage" list that holds the properties of the factory sections, the "departList" list that holds only the names of the factory sections and section time, and the general car information that you will use in the report.

When I started the program, I connected each section to the linked list, keeping the data of the department names, number of partitions and runtime that came with the "AddDept" command from the input file, as well as "doneCarCounter, isEmpty, carlist currentProcCar, doneCarList" I also kept the "departList" information that will be used in the reporting system in this section. Then, after the department creation process was performed, it was printed on the screen.

```
Department Frame has been created.  
Department Paint has been created.  
Department Engine has been created.  
Department Electronics has been created.  
Department Indoor has been created.  
Department Test has been created.
```

Then, with the "PrintFactory" command, the generated linked list is printed on the screen. A tab was left in the printing process to visually show the different parts.

```

-Frame 1 Frame 2
  -Paint 1 Paint 2
    -Engine 1 Engine 2 Engine 3 Engine 4
      -Electronics 1 Electronics 2
        -Indoor 1 Indoor 2 Indoor 3
          -Test 1 Test 2

```

I then list the car model, the car special number, the start time, and the "queueInsertTime, startProc, orgStartTime" information I will use later in the car list structure that comes with the "Produce" command.

Then the "Report" command was used to execute the program in the form of the desired reporting system. For the "report" function written for the reporting system, I sent the factory section, the starting element of the car and the "departList" list, the factory "reportTime", the report type, the special number of the car to be reported.

Within this function, the system turns around the time it will be processed and finds the first car to be added to the factory sections. This insertion was determined based on the car's starting time priority. Then, if the car was finished in the department, the completed car information was added to the contents of this section. If the duration of the car is full in that section, the next division is transferred. In this way, all of the cars on the list of cars arriving in order are processed one by one and their processes are indicated. Lastly, the moment the factory is full, the screen is printed according to the desired reporting system.

To mention the general algorithm I use, it is placed on the "factoryStage" linked with the linked list, in the order of "carList" where the cars are kept, and then in the empty factory space first. If the factory section is not empty and a new car arrives, the new car waits for the factory section to empty. If the two vehicles arrive at the same time, they are placed in the factory area, which is empty according to the priority. If the factory section is not empty and more than one car section is waiting to be emptied, the emptied section is divided by the priority of the next vehicle start time. The car which has finished its operation in the factory section, the next division process is carried out by the above checks. The information of the vehicles processed in each section and processed in the department is also kept separately for each department. The vehicle processed in the factory parts enters the next division, not the division with the same name. The order in the factory sections is determined by the order from the "AddDept" command. Processing logic works with linked list push. This is the type of screen printout. These actions apply to all reporting options. For "Report Deepman" all the sections and the list of cars processed in the section are printed on the screen. In the "Report Cars" report, the cars are processed in the sections described above. In the content of this reporting system, the duration of the transactions in which the cars are passed and the time in the section in which they are not completed are indicated, the start times and the percentage of completion of the car are printed on the screen. In the "Report Car" report, the information is printed on the screen as if the specified car were in the "Report Cars" system.

NOTE:

In the "Report Department" section, the printout of the departmental properties is printed in the input file in the order in which the suppression algorithm works. In addition, I printed out the section of the factory section to be printed after the "PrintFactory" function, adding tabs according to sorting while printing different sections. Because when I print the length of the death names, the display on the screen is distorted.