CS 558: Computer Systems Lab Assignment
Deadline: 11:59 pm, 11 March 2022

Instructions:

- This assignment is to be done in groups. Continue with the same group partner you had for the last assignment.
- The programs can be written in C/C++/Java.
- Your code should have a readme file, a makefile, and it should be well commented. These will carry separate marks for each question.
- No extensions in submission are allowed. Delay in submission will lead to penalty in marks.
- Assignments submitted before the deadline will only be considered for evaluation.
- Please do not email your assignments separately to the TAs, it will not be considered for evaluation.
- Your code will be checked for plagiarism. Any kind of academic dishonesty, plagiarism, etc. will lead to penalties.
- No sharing of code between students, submission of downloaded code is allowed.
- The first instance of code copying will result in ZERO marks for the assignment. The second instance of code copying will result in a `F' grade. Students may also be reported to the Students Disciplinary Committee, which can impose additional penalties.
- TOTAL MARKS = 30X3 (Includes 10 marks for viva for each question, 5 for readme, makefile, and, proper comments in the code)

1. Hotel TAJ has three blocks named A, B and C with equal facilities. Each of the blocks has N rooms with different capacities. There are n_1 rooms with single occupancy, n_2 rooms with double occupancy and n_3 rooms with triple occupancy. Rooms are made in such a way that triple occupancy rooms are nearer to the office room where manager allocates rooms. Double occupancy rooms are far from the office. Only physical room booking is available in the current scenario. The room rent of the different types of rooms are given in Table 1.

Sl No.	Room Type	Price/Day(Rs)
1	Single Occupancy	5000
2	Double Occupancy	9000
3	Triple Occupancy	12500

Table 1: Room Rents

The hotel's management decides to allocate three managers Aman, Raj and Alok to blocks A, B, and C respectively. Aman tries to make the customer happy by allocating room as fast as possible without bothering about the company's profit.

Raj is not loyal to the management and assigns Triple occupancy rooms first followed by Double occupancy rooms and finally single occupancy rooms.

Alok tries to maximize the company's revenue by allocating the rooms as per the need of the customers and utilizes the available rooms intelligently so that more number of bookings can be made.

Let us suppose that a total of M bookings has been made on 26th January 2022 to each of the blocks with different occupancy requirements. The management tries to analyze the following things:

1. Who is the best manager among the three who maximize the company profit?
2. What is the individual performance of each manager?
3. How much money was wasted by the individual manager by wrong allotment of rooms?
4. How many booking requests were rejected because of the nature of each manager?

Write a program in C/C++/Java that will simulate the above scenario. All the inputs should be read from the file and output should be written to the console and file.

Assumption:

A request which is not confirmed at the time of booking is rejected.

2. The Department of CSE has a plan to appoint a librarian who will maintain book records in the departmental library. The library maintains a sufficient amount of books for B.Tech, M.Tech, and Ph.D. courses in the library depo (a place where all books are kept). There are a number of subjects in each course as mentioned in Table 2. Subjects are identified by unique alphanumeric subject_ids. The first letter of the subject_id is the same as the first letter of the course. Each subject has one type of book with multiple copies. Multiple copies of a book are maintained by unique book_ids. To minimize the time for searching of books from a large set of books in the library depo and to avoid overcrowding, the librarian maintains three trays (T_1, T_2 and T_3) that can hold a fixed number of books. T_1 is the tray that keeps the books of B.Tech students, T_2, is the tray that keeps the books of M.Tech students and T_3 is the tray that keeps the books of Ph.D students**.** Maximum number of books that can be placed in each tray is mentioned in Table 2.

At the time of issuing a book, the librarian searches in trays T_1, T_2 and T_3 first depending on the course. If a book is found in the corresponding tray, he issues it to the student and replaces another copy of the book at the same place at a later point of time. Initially, the trays are empty. Librarian is not sure about the approach which should be followed to keep the books in the trays.

Tray Number	Course	Number of subjects	Maximum number of books per tray
T_1	B.Tech	n_b	t_b
T_2	M.Tech	n_m	t_m
T_3	Ph.D	n_p	t_p

Table 1: Book Count

Following are the three techniques used by the librarian for three different courses:

1. **B.Tech:** In case a requested book is not found in T_1, he brings two copies of the same book from the library depo. One is issued to the students and the other is kept in the tray. If the capacity of the tray is full then the book which was kept in the tray first is removed and a book of requested subject is placed at that place in First In First Out (FIFO) manner.
2. **M.Tech:** In case a requested book is not found in T_2, he brings two copies of the same book from the library depo. One copy is issued to the student and another copy replaces the book of a subject which has been requested least frequently.
3. **Ph.D:** In case a requested book is not found in T_3, he brings two copies of the same book from the library depo. One copy is issued to the student and another copy replaces the book of the subject which was least recently issued.

Conditions to be followed:

- All the books in a tray are from different subjects.
- Students have no idea about the book_id. They only request to issue a book by subject_id.
- The number of books in the tray must be less than or equal to the number of subjects per course (eg. $t_b <= n_b$).

Department plans to automate this task in future so that one robot will do the task of librarian. Write a program in C/C++/Java that will simulate the above scenario.

Input: All the inputs should be read from a file. First line of the input file should be the number of subjects for each course separated by blank spaces. Second line of the file should be the number of books per tray. Third line of the file should be the subject ids of the books that students have issued.
Output: The program should give the following output which should be written both in the console as well as in an output file:

a) Number of times the librarian has to go to search the entire book depo during issue of a book.
b) The ids of the subjects for which maximum number of books are issued from each course.
c) The ids of books and their corresponding subject_ids present at the end of simulation in each tray.

3. Implement a simple file system which makes the following assumptions:

- There is only a single level root directory.
- The root directory can have any number of files of any size.
- Each file has a unique name.
- Each disk block which stores the file is of size 4 Bytes.
- A file can be created, deleted, or renamed at any time.

When the contents of a file are given as input in the command line, the file system should be capable of creating as many disk blocks as necessary with the contents of the file in the form of .txt files with disk numbers as their names. Every file should have an inode pointing to all the disk blocks that store the file. When a file is renamed, it should still have the same inode number. When a file is deleted, all the corresponding disk blocks should be deleted from the file system and its meta-data (name and inode number) should be deleted from the directory. When a file is read, all the contents of the input file are supposed to be retrieved correctly.

Implement a file system that takes the following commands as inputs :

1. **mf file-name "filecontents"**: Should create the file with the given file-name and file contents.
   Example : mf File1 "This is file 1"

2. **df file-name** : Should delete the file from the directory.
   Example : df File1

3. **rf file-name1 file-name2** : Should rename the file from file1 to file2.

4. **pf file-name** : Should display all the contents of the file correctly.
   Example : "This is file 1".

5. **ls** : Should display all the file-names and their inode number correctly.