CSCI3180 – Principles of Programming Languages – Spring 2019

Assignment 1 — Human Resources Management System in FORTRAN and COBOL

Deadline: Feb 03, 2019 (Sunday) 23:59

1   Introduction

In this assignment, you have to implement a system for processing employee attendance records. You are required to implement it once using FORTRAN and once using COBOL.

2   Assignment Details

2.1   Company Attendance Record Processing

Jimmy is designing a simple Human Resources Management System (the "System" hereafter) for the PPL corporation. He is asking you to implement the attendance tracking module of this system, which reads the employees' daily attendance records, updates the monthly attendance records, and generates a daily attendance summary for the human resources manager.

The PPL corporation's employees have to work from 10:00am to 5:00pm. Employees have to tap their smart staff card at the attendance reader when they arrive at work and depart after work respectively. They will be fined for being late for each 15-minute period in whole. For example, an employee will be fined for being 15 minutes late if he or she comes at 10:27am. On the other hand, the PPL corporation provides overtime work compensation. Every month, employees can claim at most 30 overtime working hours with proven attendance records. Only complete hours of overtime work are counted. For example, if an employee leaves work at 6:47pm, he or she can claim one hour of overtime work for that day.

The System keeps the employee data in a file: **employees.txt**. This file contains personal information of all employees. Each line of the file is an employee record, containing the staff number, name, gender, date of birth, hiring date, department and monthly salary of the employee. Also, the System keeps an employee monthly attendance summary: **monthly-attendance.txt**. This file contains the monthly attendance status of all employees. The first line of this file contains the month information. In the rest of the file, each line contains the staff number, number of days absent, number of complete 15-minute periods being late, and the number of complete overtime work hours of an employee. Both files are sorted by the staff number in increasing order. The following is an example of the **employees.txt** and **monthly-attendance.txt** files.

employees.txt

```
1009CHAN        TAI MAN           M1992-01-012007-02-04ITD024320
1077WONG        ALICE             F1990-10-102007-02-04ITD020000
1823WONG        SIU MING          M1991-08-082007-02-04HRD015000
```

monthly-attendance.txt

```
2019-01
1009001002001
1077000000000
1823000002000
```

At every midnight, the System will generate one file: **attendance.txt**. This file contains all employee arrival/departure information recorded by the attendance reader on the previous day. The first line of the file is the date information. If it is the first day of the month, we need to reset the monthly attendance summary for all employees and also change the month information on the first line of the file **monthly-attendance.txt**. The rest of the file contains one tapping record per line, which includes the staff number, status and tapping time of an employee. There are two possible statuses: *ARRIVE* and *LEAVE*. Records with *ARRIVE* statuses show the time of arrival for the corresponding employee. Records with *LEAVE* statuses show the time of departure. The attendance reader will mark all records before 5:00pm as *ARRIVE* and all records at or after 5:00pm as *LEAVE*. Records in this file will be sorted by the tapping time in chronological order.

Every employee should have one *ARRIVE* and one *LEAVE* record each day. If there is more than one *ARRIVE* record for the same employee, we handle only the first *ARRIVE* record. Similarly, if there is more than one *LEAVE* record for the same employee, we handle only the first *LEAVE* record. If an employee has either the *ARRIVE* or *LEAVE* records (but not both) missing, we should report the employee's name to the human resources manager with the *SUSPICIOUS* status and should not update the monthly attendance summary for that employee. If an employee has no attendance records in a particular day, the employee is considered absent.

The following is an example of the **attendance.txt** file:

```
2019-01-04
1077ARRIVE2019-01-04-09:54
1823ARRIVE2019-01-04-09:55
1009ARRIVE2019-01-04-10:43
1077LEAVE 2019-01-04-18:45
1009LEAVE 2019-01-04-19:23
```

From the above example, we can see only staff 1009 was late for work, and would be penalized for being late for two counts of complete 15-minute periods. We report staff 1009 to the human resources manager with the *LATE* status. Also, we can see staff 1077 and 1009 worked overtime. They would be rewarded with one and two overtime working hours respectively. Moreover, we can see staff 1823 had only one *ARRIVE* record, we should report to the manager with the *SUSPICIOUS* status and not update the monthly attendance summary for staff 1823.

The System will also generate a daily summary for the human resources manager: **summary.txt**. The file consists of three parts. The first part of the file gives the headers of the summary. The second part of the file contains the attendance summary for all employees. Each line contains the staff number, name, department and attendance status of an employee. The status can be *PRESENT*, *LATE*, *ABSENT*, or *SUSPICIOUS*. If an employee has no attendance records, the status should be *ABSENT*. If an employee has an *ARRIVE* record and one *LEAVE* record, the status should be *PRESENT* for those arriving on time or *LATE* for those arriving late. Otherwise, the status should be *SUSPICIOUS*. The last part of the file reports the occurrence of each status.

The following is an example of the summary.txt file based on the above example **attendance.txt** and **employees.txt** files,

```
Daily Attendance Summary
Date: January 4, 2019
Staff-ID Name                          Department Status
----------------------------------------------------------
1009     CHAN      TAI MAN             ITD        LATE
1077     WONG      ALICE               ITD        PRESENT
1823     WONG      SIU MING            HRD        SUSPICIOUS
----------------------------------------------------------
Number of Presences:    1
```

```
Number of Absences:        0
Number of Late Arrivals:       1
Number of Suspicious Records:      1
```

To sum up, you are required to implement a program to <mark>update the monthly attendance summary</mark> and <mark>generate a daily attendance summary</mark> based on the employee information and the daily attendance records. Your calculations should be based on the policy described above. Here is a diagram summarizing the relationship of all files.
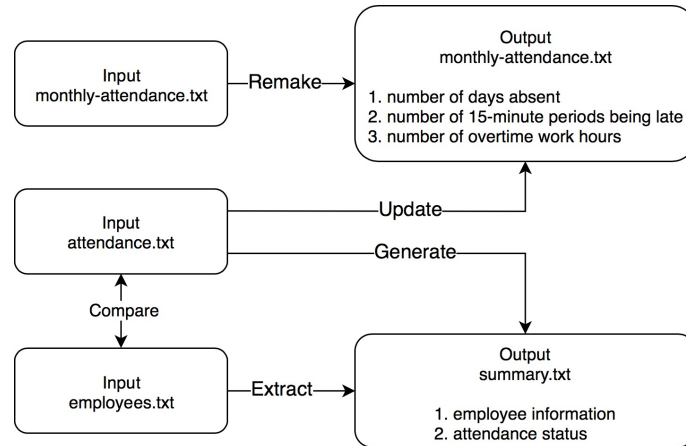

Figure 1: Relationship of input and output files

2.2 General Specification

You are required to write two programs, one in FORTRAN and the other one in COBOL, for the attendance tracking module of the System. You should name your FORTRAN source as "atd.for" and your COBOL source as "atd.cob".

1. **Input and Output Specification**
   Your programs should <mark>read three input files</mark>: **employees.txt**, **attendance.txt**, and **monthly-attendance.txt**, which contain employees' information, daily and monthly attendance records. Your program should <mark>compute</mark> the number of complete 15-minute periods being late, number of complete overtime work hours, number of days absent, and the status of every employee. These calculations should be strictly based on the policy described above. Your program should output two files for updating the monthly attendance information and generating a summary for the human resources manager. The output files should follow the description in Section 2.4.

   For FORTRAN, the name of the input files should be passed to your program as parameters in the command line:

   ```
   ./atd employees.txt attendance.txt monthly-attendance.txt
   ```

   For COBOL, you should "hardcode" the name of the input files in your program EXACTLY as "employees.txt", "attendance.txt" and "monthly-attendance.txt".

   The naming of output files should be as follows:
   Output ASCII filenames for FORTRAN: **monthly-attendancefor.txt**, **summaryfor.txt**
   Output ASCII filenames for COBOL: **monthly-attendancecob.txt**, **summarycob.txt**

2. **Restrictions on using FORTRAN and COBOL**
   In order to force you to program as in the old days, <mark>ONLY</mark> 2 keywords are allowed in

selection and loop statements: "IF" and "GOTO". You are not allowed to use modern control constructs, such as if-then-else or while loop. Using any other keywords will receive marks deduction.

3. **Error Handling**
   The programs should also handle possible errors gracefully by printing meaningful error messages to the standard output. For example, failure to open a non-existing file. However, you **CAN** assume that the input files are free of errors.

4. **Good Programming Style**
   A good programming style not only improves your grade but also helps you a lot in debugging. Poor programming style will receive marks deduction. Construct your program with good readability and modularity. Provide sufficient documentation by commenting your codes properly but never redundantly. Divide up your programs into subroutines instead of clogging the main program. The main section of your program should only handle the basic file manipulation such as file opening and closing, and subprogram calling. The main purpose of programming is not just to make the program right but also make it good.

5. **Other Notes**
   You are **NOT** allowed to implement your program in another language (e.g. Assembly/C/C++) and then initiate system calls or external library calls in FORTRAN and COBOL. Your source codes will be compiled and PERUSED, and the object code tested!

   Do not implement your programs in multiple source files. Although FORTRAN and COBOL do allow you to build a project with subroutines scattered among multiple source files, you should only submit one source file for each language.

   NO PLAGIARISM!!!! You are free to design your own algorithm and code your own implementation, but you should not "borrow" codes from your classmates. If you use an algorithm or code snippet that is publicly available or use codes from your classmates or friends, be sure to DECLARE it in the comments of your program. Failure to comply will be considered as plagiarism.

   A crash introduction to FORTRAN and COBOL will be given in the upcoming tutorials. Please DO attend the tutorials to get a brief idea on these two languages, and then learn the languages by yourselves. For a more in-depth study, we encourage students to search relevant resources on the Internet (just Google it!).

2.3 Input File Format Specification

There are three input files: **employees.txt**, **attendance.txt**, and **monthly-attendance.txt**. All input files are in plain ASCII texts. Each line is ended with the characters "\r\n". You should strictly follow the format as stated in the following.

- The dates used in all files are specified by a string of the form: YYYY-MM-DD, where YYYY, MM, and DD represent a 4-digit year, a 2-digit month, and a 2-digit day respectively. If a month of a day has only one digit, start the number with zero. For example, the month of May is specified as "05".

- Each line of **employees.txt** contains eight fields of fixed lengths of an employee.

  1. *Staff number*: a 4-digit string. If it is less than 1000, the left side of the number will be padded with zero(es).

  2. *Last name*: a string of length 10. If there are less than 10 alphanumeric characters in the name, spaces would be padded at the right side of the name.

  3. *First name*: a string of length 20. If there are less than 20 alphanumeric characters in the name, spaces would be padded at the right side of the name.

4. *Gender*: a single character. It is always 'M' or 'F'.

5. *Date of birth*: a string of length 10. It is in date format.

6. *Hiring date*: a string of length 10. It is in date format.

7. *Department*: a string of length 3. It is an abbreviation of the department. It always consists of 3 capital letters.

8. *Monthly salary*: a 6-digit string. It is between 0 to 999999. If it is less than 100000, the left side of the number will be padded with zero(es).

- File **attendance.txt** is as follows. The first line is the date in date format. The rest of the file contains three fields of fixed lengths of a daily attendance record of an employee on each line.

1. *Staff number*: a 4-digit string. If it is less than 1000, the left side of the number will be padded with zero(es).

2. *Status*: a string of length 6. It is always "ARRIVE", or "LEAVE". If length of status is less than 6, spaces would be padded on the right side of the status.

3. *Time*: a 16-character string of the form YYYY-MM-DD-HH:NN, where YYYY, MM, DD, HH, and NN represent a 4-digit year, a 2-digit month, a 2-digit day, a 2-digit hour, and a 2-digit minute. Time is written in 24-hour clock convention. If a 2-digit number is less than 10, there is a leading zero.

- File **monthly-attendance.txt** is as follows. The first line is month information, in YYYY-MM format, representing a 4-digit year and a 2-digit month. The rest of the file contains four fields of fixed lengths of a monthly attendance record of an employee on each line. The number of lines in this file should be the same as that in the **employees.txt** file.

1. *Staff number*: a 4-digit string. If it is less than 1000, the left side of the number will be padded with zero(es).

2. *Number of days of absent*: a 3-digit string. If it is less than 100, the left side of the number will be padded with zero(es).

3. *Number of complete 15-minute periods being late*: a 3-digit string. If it is less than 100, the left side of the number will be padded with zero(es).

4. *Number of overtime work hours*: a 3-digit string. If it is less than 100, the left side of the number will be padded with zero(es).

You may make the following assumptions on the files:

- All dates and times are valid.

- Maximum number of employees is 9999.

- File **employees.txt** is sorted by staff number in ascending order.

- File **attendance.txt** is sorted by time in chronological order from the second line onward.

- File **monthly-attendance.txt** is sorted by staff number in ascending order from the second line onward.

- The timestamp of the first attendance record in file **attendance.txt** is on or after YYYY-MM-DD-00:00, where YYYY-MM-DD is the date on the first line of the same file.

- The timestamp of the last attendance record in file **attendance.txt** is on or before YYYY-MM-DD-23:59, where YYYY-MM-DD is the date on the first line of the same file.

- Each line respects the description in this specification.

The following are examples of the input files.

employees.txt

```
1023CHAN      TAI MAN             M1992-01-012007-02-04ITD024320
1024WONG      SIU MING            M1993-11-112007-02-04ITD024320
1025A123456789B1234567890123456789F1993-11-112007-02-04ITD024320
1026PAN       PETER               M1993-11-112007-02-04HRD024320
```

attendance.txt

```
2018-09-29
1026ARRIVE2018-09-29-10:10
1023ARRIVE2018-09-29-10:28
1025LEAVE 2018-09-29-20:01
1026LEAVE 2018-09-29-20:01
1023LEAVE 2018-09-29-20:01
```

monthly-attendance.txt

```
2018-09
1023001000000
1024000001000
1025001011002
1026003000001
```

2.4 Output File Format Specification

There should be two output files, **monthly-attendanceXXX.txt**, **summaryXXX.txt** (**XXX = for** or **cob**), containing the updated monthly attendance summary and the daily attendance summary. Each line is ended with the characters "\r\n". You should strictly follow the format as stated in the following.

- File **monthly-attendanceXXX.txt** should have the same format as **monthly-attendance.txt** as described in Section 2.3. It should be sorted by staff number in ascending order from second line onward.

- File **summaryXXX.txt** is as follows. The file contains three parts.

    - The first part is the header of summary. It consists of four lines.
        1. The first line is fixed as "Daily Attendance Summary".
        2. The second line is the date. It begins with "Date: ", followed by the date information, which is the same as the date information in **attendance.txt**. However, the date is written with month, day, and year in order with a comma before the year. Month is given in english. There are no leading zeroes. The date has length of 18. Space(s) is padded on the right of the date if needed. For example, "2018-01-02" in **attendance.txt** becomes "January 2, 2018   " in **summary.txt**.
        3. The third line is the field header. It begins with "Staff-ID Name", followed by 28 spaces, and followed by "Department Status".
        4. The fourth line contains 62 "-".

    - The second part is the attendance summary for all employees. Each line contains four fields of fixed lengths of attendance summary. The number of lines in this file is the same as that in **employees.txt**. It should be sorted by staff number in ascending order.
        1. *Staff number*, *Last name*, *First name*, *Department*: same information and format from **employees.txt**, i.e. strings of length 4, 10, 20, 3, respectively, with zero(es)

or spaces padded if needed. *Staff number*, *last name*, and *department* are aligned with the field headers `Staff-ID`, `Name`, and `Department` respectively. There is a space between *last name* and *first name*.

2. *Status*: a string of length 10. It is always "PRESENT", "LATE", "ABSENT", or "SUSPICIOUS". If the length of status is less than 10, spaces should be padded on the right of the status. It is aligned with the field header `Status`.

– The last part summerizes the occurrence of each status. It consists of five lines.

1. The first line contains 62 "-".
2. The second line begins with "`Number of Presences: `", followed by the number of staff with the "PRESENT" status.
3. The third line begins with "`Number of Absences: `", followed by the number of staff with the "ABSENT" status.
4. The fourth line begins with "`Number of Late Arrivals: `", followed by the number of staff with the "LATE" status.
5. The fifth line begins with "`Number of Suspicious Records: `", followed by the number of staff with the "SUSPICOUS" status.
6. All numbers are of length 4. There are no leading zeroes. If a number is less than 1000, the left side of the number is padded with space(s).

Please make sure that each employee in **employees.txt** has a corresponding line in both the **monthly-attendanceXXX.txt** and **summaryXXX.txt** files.

The following is the content of the output files based on the sample input files given.
monthly-attendanceXXX.txt

```
2018-09
1023001001003
1024001001000
1025001011002
1026003000004
```

summaryXXX.txt

```
Daily Attendance Summary
Date: September 29, 2018
Staff-ID Name                            Department Status
--------------------------------------------------------------
1023    CHAN     TAI MAN                 ITD        LATE
1024    WONG     SIU MING                ITD        ABSENT
1025    A123456789 B1234567890123456789 ITD        SUSPICIOUS
1026    PAN      PETER                   HRD        PRESENT
--------------------------------------------------------------
Number of Presences:       1
Number of Absences:        1
Number of Late Arrivals:      1
Number of Suspicious Records:      1
```

2.5 Report

You should give a simple report to answer the following questions within one A4 page:

1. Compare the conveniences and difficulties in implementing the attendance tracking module of the Human Resources Management System in FORTRAN and COBOL. You can divide

the implementation into specific tasks such as "reading file in certain format", "simulating loops", "case control" and so on. Give code segments in your programs to support your explanation.

2. Compare FORTRAN and COBOL with modern programming languages (e.g. Java/C++/...) from different aspects (e.g. paradigm, data type, parameter parsing, ...). You are free to pick your favorite modern programming language.

3. In your program design, how do you separate the tasks into submodules? Tell us briefly the functionality of each submodule and the main flow of your program in terms of these submodules.

3   Submission Guidelines

Please read the guidelines CAREFULLY. If you fail to meet the deadline because of submission problem on your side, marks will still be deducted. So please start your work early!

1. In the following, **SUPPOSE**

   > your name is *Chan Tai Man*,
   > your student ID is *1155234567*,
   > your username is *tmchan*, and
   > your email address is *tmchan@cse.cuhk.edu.hk*.

2. In your source files, insert the following header. REMEMBER to insert the header according to the comment rule of FORTRAN and COBOL.

```
/*
 * CSCI3180 Principles of Programming Languages
 *
 * --- Declaration ---
 *
 * I declare that the assignment here submitted is original except for source
 * material explicitly acknowledged.  I also acknowledge that I am aware of
 * University policy and regulations on honesty in academic work, and of the
 * disciplinary guidelines and procedures applicable to breaches of such policy
 * and regulations, as contained in the website
 * http://www.cuhk.edu.hk/policy/academichonesty/
 *
 * Assignment 1
 * Name : Chan Tai Man
 * Student ID : 1155234567
 * Email Addr : tmchan@cse.cuhk.edu.hk
 */
```

   The sample file header is available at

   > `http://course.cse.cuhk.edu.hk/~csci3180/resource/header.txt`

3. Make sure you compile and run the FORTRAN program without any problem with f77 on Solaris (sparc machine).

4. Make sure you compile and run the COBOL program without any problem with Open-COBOL 1.1 on Windows computers in SHB924/904. We will grade your works based on those machines.

5. The report should be submitted to VeriGuide, which will generate a submission receipt. The report and receipt should be submitted together with your FORTRAN and COBOL codes in the same ZIP archive.

6. The FORTRAN source should have the filename "atd.for". The COBOL source should have the filename "atd.cob". The report should have the filename "report.pdf". The VeriGuide receipt of report should have the filename "receipt.pdf". All file naming should be followed strictly and without the quotes.

7. Tar your source files to `username.tar` by

   ```
   tar cvf tmchan.tar atd.for atd.cob report.pdf receipt.pdf
   ```

8. Gzip the `tarred` file to `username.tar.gz` by

   ```
   gzip tmchan.tar
   ```

9. Uuencode the `gzipped` file and send it to the course account with the email title "HW1 *studentID yourName*" by

   ```
   uuencode tmchan.tar.gz tmchan.tar.gz \
   | mailx -s "HW1 1155234567 Chan Tai Man" csci3180@cse.cuhk.edu.hk
   ```

10. Please submit your assignment using your Unix accounts.

11. An acknowledgement email will be sent to you if your assignment is received. **DO NOT** delete or modify the acknowledgement email. You should contact your TAs for help if you do not receive the acknowledgement email within 5 minutes after your submission. **DO NOT** re-submit just because you do not receive the acknowledgement email.

12. You can check your submission status at

    http://course.cse.cuhk.edu.hk/~csci3180/submit/hw1.html.

13. You can re-submit your assignment, but we will only grade the latest submission.

14. Enjoy your work :>