Jorge A. Barrios, 1155128883
CSCI3180, Assignment 1
Feb 3, 2019

Human Resources Management System in FORTRAN and COBOL Report

Question 1.

The conveniences and difficulties in implementing the attendance tracking module of the Human Resources Management System in FORTRAN and COBOL can be divided into "reading file in certain format", "simulating loops", "subprograms".

Reading a file requires similar instructions like "Open", "Read" and "Close", but COBOL is better suited for the HRMS formats because it's naturally designed to process records, which are common in business processes, whereas FORTRAN's scientific nature provides more options for numerical computation precision instead of display. For example, the File Section the Data Division in COBOL is very useful to directly read all variables from files into a predefined structure, this took some hacky iterative character array manipulations in FORTRAN, such as " character arr(10000)*100".

Loop simulation uses a combination of IF and GO TO statements with the limitations specified for this assignment and was thus similar for COBOL and FORTRAN. COBOL had the advantage of specifying meaningful label names with alphanumeric characters, while FORTRAN used cumbersome integers. For example "GO TO WRITE-SUMMARY-FOOTER" is more readable than "go to 13". COBOL also was limited to paragraph names for labels and this resulted in confusions between modularity and branching.

Subprograms. FORTRAN provided more intuitive options with subroutines and parameters, such as "subroutine swap(arr,a,b)" whereas fortran used a more ambiguous and limited paragraph convention, such as "PROCESS-EMPLOYEE.".

Question 2.

A comparison of COBOL and FORTRAN to Python reveals that the latter is of a much higher level of abstraction, it provides OOP features and is interpreted. Python is more abstract because it can do the same task without needing to specify how to do it with much precision, for example its use of dynamic type binding, whereas in COBOL and FORTRAN types are statically binded. Python has OOP constructs such as classes and other ADTs which help with reusability of code, which COBOL and FORTRAN lack. Python's interpretation makes it more flexible again than COBOL and FORTRAN's compilation, but also slower, which is why FORTRAN is better suited for low-latency computing. FORTRAN and COBOL's static nature is comprehensive.

Question 3.

1. Preprocess files -> Open, Read, sort files and write headers
2. Process Employee -> Read info for an employee until EOF and go to # 5
3. Process Attendance -> Calculate attendance status for employee
4. Process Monthly Attendance -> Update monthly attendance for employee and go to #2
5. Post Process files -> Write footers and close files