SCOMP

Sprint C

US 2001b As Product Owner, I want the system to, continuously, process the files produced by the Applications Email Bot, so that they can be imported into the system by initiative of the Operator
– Priority: 1
– References: See NFR13(SCOMP).

US 4000 As a Customer Manager, when displaying the candidate data, I expect the system to present a top 20 list of the most frequently referenced words from files uploaded by a candidate. Additionally, I require a comprehensive list of the files in which these words appear
– Priority: 1
– References: See NFR14(SCOMP)

NFR13

The “Applications File Bot” must be developed in C and utilize **processes, shared memory and semaphore** primitives. It's crucial to eliminate busy waiting wherever possible.

A child process should be created to periodically monitor an input directory for new files related to the 'Application' phase of the recruitment process. If new files are detected, a notification (using a semaphore) should be sent to the parent process.

Please refer to Section 2.2.3 of the “System Specification” document for a description of the input directory, input files, output directory, and their expected subdirectories.

Upon receiving the notification, the parent process should distribute the new files among a fixed number of worker child processes. Each child process will be responsible for copying all files related to a specific candidate to its designated subdirectory in the output directory.

Once a child has finished copying all files for a candidate, it should inform its parent that it is ready to perform additional work. Child workers do not terminate.

Once all files for all candidates have been copied, the parent process should generate a report file in the output directory. This report should list, for each candidate, the name of the output subdirectory and the names of all files that were copied.

The names of the input and output directories, the number of worker children, the time interval for periodic checking of new files, etc., should be configurable. This configuration can be achieved either through input parameters provided when running the application or by reading from a configuration file.

Unit and integration tests are highly valued.

NFR14

This US should be developed in Java and utilize threads along with the synchronization mechanisms outlined in SCOMP.

Utilization of any Java concurrency mechanisms beyond those covered in class, including thread pools, streams, etc., is strictly prohibited. Similarly, avoid employing data types that inherently support concurrent access.

The objective is to compile a list of the top 20 most frequently referenced words from files submitted by a candidate. This list should include the number of occurrences of each word and the corresponding files in which they appear.

Multiple approaches can be taken for processing files: either assigning a thread per file or allowing multiple threads to process a single file. Nevertheless, synchronization among threads is imperative to guarantee the accuracy of the obtained results.

Unit and integration tests are highly valued.