CS19003 : Programming and Data Structures Laboratory

Assignment 9 : File Handling Operations in C

Date: 29-Jun-2021

Problem Statement:

While the court proceedings of the murder suspects are going at a full swing, Detective Bakshi also got hold of two eye-witnesses who were present at the spot during the time of these murders. After he and Ajit interrogated them, they confessed many important details related to the murder mystery. So, he asked these eye-witnesses to write down all the details as they saw and told. Taking the documents prepared by them, Mr. Bakshi asked Ajit to find out the commonalities in these two text documents that they have written. However, Ajit cannot find any way to check the commonalities between these two narrations as they wrote a very long text.

Now, can you help Ajit by automating this process through a generic C-program which can check the union, intersection, differences of the words present in the two text file documents that are written by these eye-witnesses? In particular, your program will do the following:

- Take from user as input a dictionary file, say `wordlist.txt`, containing – (a) in the first line, the number of words and the maximum length of an word in the dictionary; (b) from the second line, all the words in a newline till the end of file (sorted in ascending lexicographic order). By reading this file and with the help of its parameters, dynamically allocate the memory for the dictionary of words as a two-dimensional array (i.e. array of strings), defined as `char **dictionary` where all these words are being stored runtime.
- Now, create two files, say `File1` and `File2`, by **randomly** picking N_1 and N_2 words from the dictionary array, so that the words appear inside each of these files are also in ascending lexicographic order and are space-separated (however, words may repeat while being picked up randomly from dictionary). Here, the filenames and N_1, N_2 values will be provided as input by the user.
  Hint: A random number generation code snippet is provided below. Please modify this concept as required.

```
#include <stdlib.h>     // for srand(), rand()
#include <sys/types.h>  // for getpid()
#include <unistd.h>     // for getpid()
int main()
{
  srand( getpid() );    // declare srand() in the beginning of main (once)
  x = rand() % 10;      // generate any random number [0-9]
  return 0;
}
```

- Now, create an output file (named "`output.txt`") and there first write down the contents of `File1` and `File2` (as it is) one after another.
- Then, perform the following four set-theoretic operations on the set of words present in these two files:
  1. Take a *complete union* of all the words present in both these files (include all duplicate words also) and append it in the output file created.
  2. Take an *intersection* between all the words that are present in both of these files (include all duplicate/common intersecting words too) and append it in the output file created.
  3. Take the *difference* of these files, that is, all those words which are present in `File1`, but not in `File2` and append it in the output file created.
  4. Take the *symmetric-difference* of these files, that is, all the words which are present either `File1` or in `File2` (exclusively and not in both) and then append those also in the output file created.

  A sample of the output file produced containing all these outcomes (as enlisted above) are also presented in the execution details part (under Sample-1).
- Moreover, while opening of the input and output files, keep suitable checks over the file pointers in order to ensure the existence of the file and prompt warning messages (refer to Sample-2 of execution details).

Note: You are **not allowed** to create separate arrays from the words taken out from the file and act upon the arrays, rather you need to operate over files directly through file handling operations only.

Example Execution Details:

Sample-1:

```
++ Enter Dictionary File Name: wordlist.txt
-- Dictionary of Words Loaded --
```

```
++ Enter Number of Total Words for First File: 50
++ Enter First File Name: a.txt

-- File-1 (Name: a.txt) Generated --

++ Enter Number of Total Words for Second File: 50
++ Enter Second File Name: b.txt

-- File-2 (Name: b.txt) Generated --

== Set-Theoretic File Data Operation Statistics ==
        Generated in "output.txt" File!

+-----------------------------+
| CONTENT OF "output.txt" FILE |
+-----------------------------+
== Set-Theoretic File Data Operation Statistics ==

** File-1 Data **
acidophilic anoectochilus antigenic apogee benison bum colpocele conspire cricketground denuded denying
diametral eparchy exteriority fordable garroter guevina half-holiday harvester hedonist infuse ink
kwajalein lifegiving minimized minivan moreover notorious nullity paradise peneidae platylobium poplar
rascal reservatory rhinoptera sagittariidae scaramouch semiliterate sensualist serine spodoptera squelch
synonymist syntax todo two-lane undermined vehicles wintry

** File-2 Data **
acharnement argonauta aspidistra berg colloquialism combinative comminute deceiver decimal disencumber
dustman dysphoria elephas epanodos ferociousness font forethoughtful granulate herbarian improvisatory
inglorious lithops loadstar mage medias methyltestosterone middleman militat mutton nonkosher phi
placability poplar pout programma psychically ratables recessed render roller saturnalia sergeant spake
superscript tenements thousand uninebriated uniquely visayan washout

** Complete Union of Two File Data (F1 U F2) **
acharnement acidophilic anoectochilus antigenic apogee argonauta aspidistra benison berg bum colloquialism
colpocele combinative comminute conspire cricketground deceiver decimal denuded denying diametral
disencumber dustman dysphoria elephas epanodos eparchy exteriority ferociousness font fordable
forethoughtful garroter granulate guevina half-holiday harvester hedonist herbarian improvisatory infuse
inglorious ink kwajalein lifegiving lithops loadstar mage medias methyltestosterone middleman militat
minimized minivan moreover mutton nonkosher notorious nullity paradise peneidae phi placability
platylobium poplar poplar pout programma psychically rascal ratables recessed render reservatory
rhinoptera roller sagittariidae saturnalia scaramouch semiliterate sensualist sergeant serine spake
spodoptera squelch superscript synonymist syntax tenements thousand todo two-lane undermined uninebriated
uniquely vehicles visayan washout wintry

** Intersection of Two File Data (F1 ^ F2) **
poplar

** Difference of Two File Data (F1 - F2) **
acidophilic anoectochilus antigenic apogee benison bum colpocele conspire cricketground denuded denying
diametral eparchy exteriority fordable garroter guevina half-holiday harvester hedonist infuse ink
kwajalein lifegiving minimized minivan moreover notorious nullity paradise peneidae platylobium rascal
reservatory rhinoptera sagittariidae scaramouch semiliterate sensualist serine spodoptera squelch
synonymist syntax todo two-lane undermined vehicles wintry

** Symmetric Difference of Two File Data (F1 ~ F2) **
acharnement acidophilic anoectochilus antigenic apogee argonauta aspidistra benison berg bum colloquialism
colpocele combinative comminute conspire cricketground deceiver decimal denuded denying diametral
disencumber dustman dysphoria elephas epanodos eparchy exteriority ferociousness font fordable
forethoughtful garroter granulate guevina half-holiday harvester hedonist herbarian improvisatory infuse
inglorious ink kwajalein lifegiving lithops loadstar mage medias methyltestosterone middleman militat
minimized minivan moreover mutton nonkosher notorious nullity paradise peneidae phi placability
platylobium pout programma psychically rascal ratables recessed render reservatory rhinoptera roller
sagittariidae saturnalia scaramouch semiliterate sensualist sergeant serine spake spodoptera squelch
superscript synonymist syntax tenements thousand todo two-lane undermined uninebriated uniquely vehicles
visayan washout wintry
```

Sample-2:

```
++ Enter Dictionary File Name: abc.txt
** Error: Dictionary of Words Not Found!
```

Submit a single C source file. Do not use global/static variables.