CS 6360.004

ASSIGNMENT 5

Subject: Database Design

Topic: Single Level Index

Names of student: Pritom Das Radheshyam

Net Id: PXR180033

Tool Used: Visual Studio

I have attached with a “.cpp file” but the code can also be found here.

#define STDC\_WANT\_LIB\_EXT1 1

#include <stdio.h>

#include <io.h>

#include<fstream>

#include <iostream>

#include <string.h>

#include <cstring>

#include <stdlib.h>

#include <ctype.h>

#include <vector>

#include <set>

#include <algorithm>

#include <memory.h>

using namespace std;

union charLong

  {

  long long lpart;

  char cpart[8];

  };

charLong cl;

struct keyval{

  char key[1024];

  long long pos;

   bool operator<(const keyval& t) const

    {

        try{

        if(strcmp(this->key,t.key)==-1)

          return true;

        else

          return false;

        } catch(exception e){

           cout<<"Exception occured : "<<e.what();

           return false;

        }

    }

}

k;

int main(int argc, char \*\*argv)

{

   if(argc<5){

   } else {

     set<struct keyval> s;

     vector<struct keyval> v;

     string mode = argv[1];

     string inputfile =  argv[2];

     string outputfile =  argv[3];

     int keylen =  atoi(argv[4]);

     char line[1024];

     char keyp[1024];

     char buffer[1024];

     int count = 0;

     int bufptr = 0;

     int i;

     string create = "-c";

     string list = "-l";

     cout<<"Entered parameters"<<endl;

     cout<<mode<<" "<<inputfile<<" "<<outputfile<<" "<<keylen<<endl;

     if(!strcmp(mode.c\_str(),create.c\_str())){

      ifstream is;

      ofstream os;

      is.open(inputfile, ifstream::binary);

      if (is.fail())

        {

        cout << "error opening file" << endl;

        is.open(inputfile, ios::out);

        if (is.fail())

          {

          cout << "Real error opening file" << endl;

          system("pause");

          return 0;

          }

        is.close();

        is.open(inputfile, ios::in | ios::out | ios::binary);

        }

      try{

      while(!is.eof()){

          k.pos = is.tellg();

          is.getline(line,500);

          cout<<"Line "<<count+1<<" "<<line<<endl;

          if(strlen(line)>0){

          for(i=0;i<keylen;i++){

            k.key[i] = line[i];

          }

          k.key[i]='\0';

          cout<<"K key "<<k.key<<" K pos "<<k.pos<<endl;

          s.insert(k);

          v.push\_back(k);

          count++;

          }

      }

      is.close();

      set<struct keyval>::iterator it;

      os.open(outputfile,ios::binary);

      cout<<endl;

      cout<<" printing keys in set :-"<<endl;

      // print in ascending order as required.

      for (it = s.begin(); it != s.end(); it++)

      {

          bufptr = 0;

          cout << (\*it).key << endl;

          cl.lpart = (\*it).pos;

          memset(&buffer[bufptr], 0, 1024);

          memcpy\_s(&buffer[bufptr], keylen, (\*it).key, keylen);

          bufptr+=keylen;

          memcpy\_s(&buffer[bufptr], 8, cl.cpart, 8);

          os.write(buffer,keylen+8);

      }

      os.close();

      } catch(exception e){

        cout<<"Exception occured : "<<e.what();

      }

   } else if(!strcmp(mode.c\_str(),list.c\_str())){

     int filesize,records;

     ifstream is,isd;

     ofstream os;

     long long pos;

     char \*record = new char[keylen+1025];

     is.open(outputfile, ios::binary | ios::ate);

     isd.open(inputfile, ifstream::binary);

     os.open("outputfile.txt", ofstream::binary );

     //is.seekg(0, ios::end);

     filesize = is.tellg();

     //cout<<"filesize : "<<filesize<<endl;

     is.seekg(0, ios::beg);

     records = filesize/(keylen+8);

     //bufptr = 0;

     cout<<"The contents of the Data File in sorted order are :-"<<endl;

     for(int i=0;i<records;i++){

         is.read(buffer, keylen+8);

        // cout<<"Buffer : "<<buffer<<endl;

         memcpy\_s(cl.cpart, 8, (buffer+keylen), 8);

         pos = cl.lpart;

         isd.seekg(pos,ios::beg);

         isd.getline(record, keylen+1024);

         cout<<record<<endl;

         os<<record<<endl;

     }

    delete[]  record;

    os.close();

    isd.close();

    is.close();

   }

   }

}