Отчет по курсовому проекту N 9 по курсу

"Фундаментальная информатика"

Студент группы: М80-107Б-21, Павлов Иван Дмитриевич
Контакты: pavlov.id.2003@gmail.com
Работа выполнена: 11.05.2022
Преподаватель: Найденов Иван Евгеньевич

1 Тема

Сортировка и поиск

2 Цель работы

Составить программу на языке Си с использованием процедур и функций для сортировки таблицы заданным методом и двоичного поиска по ключу в таблице.

3 Задание

Метод сортировки: двоичными вставками
Тип ключа: строковый
Длина ключа: 6 байт
Хранение данных и ключей: вместе

4 Оборудование

Процессор: AMD Ryzen 5 4600H with Radeon Graphics
ОП: 7851 Мб
НМД: 256 Гб
Монитор: 1920x1080

5 Программное обеспечение

Операционная система семейства: linux (ubuntu), версия 20.04.3 LTS
Интерпретатор команд: bash, версия 5.0.17(1)-release.
Система программирования: gcc*, версия 17
Редактор текстов: emacs, версия 25.2.2
Утилиты операционной системы: make
Прикладные системы и программы: vscode
Местонахождение и имена файлов программ и данных на домашнем компьютере: /home/ggame/newlabs/

6 Идея, метод, алгоритм решения задачи

- Считаем исходные данные в виде структуры (ключ и значение) на динамический массив (*vector*);
- Отсортируем полученный массив структур методом двоичных вставок функция (*bin_ insertion_ sort*);
- Выведем отсортированный массив на экран;
- Считаем ключ и проведем бинарный поиск по тексту, выведем найденное значение на экран;

6.1 Принцип работы сортировки двоичными вставками

- Итеративно пройдем по всем элементам массива;
- Для двоичного поиска заведем переменные *left*, *right* и *mid*, которые будут равны соответственно началу, середине и концу отсортированной части массива;

- Запускаем двоичный поиск для отсортированной части массива, внутри него сравниваем ключи последнего и искомого элементов;
- После этого заводим переменные целого и структурного типа, и присваиваем им индекс найденной позиции массива и элемент, который нужно вставить;
- Сдвигаем все элементы отсортированной части массива вправо на 1 элемент, чтобы вставить нужный на найденную позицию, затем вставляем.

6.2 Принцип работы бинарного поиска

- Определяем переменные начала, конца и серидины массива;
- Запускаем цикл до того момента, когда переменные начала и конца совпадут; При выводе вернем элемент под этим индексом;
- Если элемент, лежащий в серидине меньше искомого, то присваиваем серидине значение (начало + 1); иначе - (конец - 1).

Так как в моем варианте строковые ключи, то я написал компаратор, урезав функцию *strcmp*. Можно проверять и обычным *strcmp*. Ключи сортируются в прямом лексеографическом порядке.

7 Сценарий выполнения работы

Тест программы:
Изначально есть 3 тестовых файла: отсортированный, реверсивный и в рандомном порядке по элементам. В результате работы должен олучиться так называемый blindtext.

Листинг 1: test1

```
ggame@ggame:~/newlabs/kp9/massiv_struktur$ ./main blindtext_sorted
Lorem ipsum dolor sit amet consectetuer adipiscing elit Aenean commodo ligula eget
    dolor Aenean massa Cum sociis natoque penatibus et magnis dis parturient montes
     nascetur ridiculus mus Donec quam felis ultricies nec pellentesque eu pretium
    quis sem Nulla consequat massa quis enim Donec pede justo fringilla vel aliquet
     nec vulputate eget arcu In enim justo rhoncus ut imperdiet a venenatis vitae
    justo Nullam dictum felis eu pede mollis pretium Integer tincidunt Cras dapibus
     Vivamus elementum semper nisi Aenean vulputate eleifend tellus Aenean leo
    ligula porttitor eu consequat vitae eleifend ac enim Aliquam lorem ante dapibus
     in viverra quis feugiat a tellus Phasellus viverra nulla ut metus varius
    laoreet Quisque rutrum Aenean imperdiet Etiam ultricies nisi vel augue
    Curabitur ullamcorper ultricies nisi Nam eget dui Etiam rhoncus Maecenas tempus
     tellus eget condimentum rhoncus sem quam semper libero sit amet adipiscing sem
     neque sed ipsum Nam quam nunc blandit vel luctus pulvinar hendrerit id lorem
    Maecenas nec odio et ante tincidunt tempus Donec vitae sapien ut libero
    venenatis faucibus Nullam quis ante Etiam sit amet orci eget eros faucibus
    tincidunt Duis leo Sed fringilla mauris sit amet nibh Donec sodales sagittis
    magna Sed consequat leo eget bibendum sodales augue velit cursus nunc

Search in text:
Enter key: aaaaaa
Your value: Lorem
Enter key: aaaabc
Your value: quam
Enter key:
ggame@ggame:~/newlabs/kp9/massiv_struktur$
```

Листинг 2: test2

```
ggame@ggame:~/newlabs/kp9/massiv_struktur$ ./main blindtext_reverse
Lorem ipsum dolor sit amet consectetuer adipiscing elit Aenean commodo ligula eget
    dolor Aenean massa Cum sociis natoque penatibus et magnis dis parturient montes
     nascetur ridiculus mus Donec quam felis ultricies nec pellentesque eu pretium
    quis sem Nulla consequat massa quis enim Donec pede justo fringilla vel aliquet
     nec vulputate eget arcu In enim justo rhoncus ut imperdiet a venenatis vitae
    justo Nullam dictum felis eu pede mollis pretium Integer tincidunt Cras dapibus
```

```
        Vivamus elementum semper nisi Aenean vulputate eleifend tellus Aenean leo
        ligula porttitor eu consequat vitae eleifend ac enim Aliquam lorem ante dapibus
        in viverra quis feugiat a tellus Phasellus viverra nulla ut metus varius
        laoreet Quisque rutrum Aenean imperdiet Etiam ultricies nisi vel augue
        Curabitur ullamcorper ultricies nisi Nam eget dui Etiam rhoncus Maecenas tempus
        tellus eget condimentum rhoncus sem quam semper libero sit amet adipiscing sem
        neque sed ipsum Nam quam nunc blandit vel luctus pulvinar hendrerit id lorem
        Maecenas nec odio et ante tincidunt tempus Donec vitae sapien ut libero
        venenatis faucibus Nullam quis ante Etiam sit amet orci eget eros faucibus
        tincidunt Duis leo Sed fringilla mauris sit amet nibh Donec sodales sagittis
        magna Sed consequat leo eget bibendum sodales augue velit cursus nunc
3
4   Search in text:
5   Enter key: aaaaaa
6   Your value: Lorem
7   Enter key: aaaabc
8   Your value: quam
9   Enter key:
10  ggame@ggame:~/newlabs/kp9/massiv_struktur$
```

Листинг 3: test3

```
1   ggame@ggame:~/newlabs/kp9/massiv_struktur$ ./main blindtext_random
2   Lorem ipsum dolor sit amet consectetuer adipiscing elit Aenean commodo ligula eget
        dolor Aenean massa Cum sociis natoque penatibus et magnis dis parturient montes
        nascetur ridiculus mus Donec quam felis ultricies nec pellentesque eu pretium
        quis sem Nulla consequat massa quis enim Donec pede justo fringilla vel aliquet
        nec vulputate eget arcu In enim justo rhoncus ut imperdiet a venenatis vitae
        justo Nullam dictum felis eu pede mollis pretium Integer tincidunt Cras dapibus
        Vivamus elementum semper nisi Aenean vulputate eleifend tellus Aenean leo
        ligula porttitor eu consequat vitae eleifend ac enim Aliquam lorem ante dapibus
        in viverra quis feugiat a tellus Phasellus viverra nulla ut metus varius
        laoreet Quisque rutrum Aenean imperdiet Etiam ultricies nisi vel augue
        Curabitur ullamcorper ultricies nisi Nam eget dui Etiam rhoncus Maecenas tempus
        tellus eget condimentum rhoncus sem quam semper libero sit amet adipiscing sem
        neque sed ipsum Nam quam nunc blandit vel luctus pulvinar hendrerit id lorem
        Maecenas nec odio et ante tincidunt tempus Donec vitae sapien ut libero
        venenatis faucibus Nullam quis ante Etiam sit amet orci eget eros faucibus
        tincidunt Duis leo Sed fringilla mauris sit amet nibh Donec sodales sagittis
        magna Sed consequat leo eget bibendum sodales augue velit cursus nunc
3
4   Search in text:
5   Enter key: aaaaaa
6   Your value: Lorem
7   Enter key: aaaabc
8   Your value: quam
9   Enter key:
10  ggame@ggame:~/newlabs/kp9/massiv_struktur$
```

8 Распечатка протокола

Листинг 4: vector.h

```
1   #ifndef __VECTOR_H__
2   #define __VECTOR_H__
3
4   #include <stdbool.h>
5   #include <string.h>
6
7   typedef int key_type;
8   typedef struct
9   {
10      char key[7];
11      char value[100];
```

```
12      } value_type;
13
14      typedef struct {
15          value_type *begin;
16          size_t size;
17          size_t allocated;
18      } vector;
19
20      void create(vector *v);
21
22      void destroy(vector *v);
23
24      bool is_empty(vector *v);
25
26      bool is_full(vector *v);
27
28      void push(vector *v, value_type value);
29
30      value_type pop(vector *v);
31
32      void print(vector *v);
33
34      size_t size(vector *v);
35
36      int comparator(value_type x, value_type y);
37
38      void bin_insertion_sort(vector *v);
39
40      value_type bin_search(vector *v, char key[7]);
41
42      #endif
```

Листинг 5: vector.c

```
1       #include <stdio.h>
2       #include <stdlib.h>
3       #include <string.h>
4       #include "vector.h"
5
6       void print(vector *v)
7       {
8           for (int i = 0; i < v->size; ++i) {
9               printf("%s ", v->begin[i].value);
10          }
11          printf("\n");
12      }
13
14      bool is_empty(vector *v)
15      {
16          return v->size == 0;
17      }
18
19      bool is_full(vector *v)
20      {
21          return v->size == v->allocated;
22      }
23
24      size_t size(vector *v)
25      {
26          return v->size;
27      }
28
29      void create(vector *v)
30      {
31          v->size = 0;
32          v->allocated = 0;
```

```
33          v->begin = NULL;
34      }
35
36      void destroy(vector *v)
37      {
38          free(v->begin);
39          v->size = 0;
40          v->allocated = 0;
41          v->begin = NULL;
42      }
43
44      void push(vector *v, value_type value)
45      {
46          if (is_full(v)) {
47              v->allocated += 10;
48              v->begin = realloc(v->begin, v->allocated * sizeof(value_type));
49          }
50          v->begin[v->size] = value;
51          v->size++;
52      }
53
54      value_type pop(vector *v)
55      {
56          value_type res = v->begin[v->size - 1];
57          v->size--;
58          return res;
59      }
60
61      int comparator(value_type x, value_type y)
62      {
63          if (strcmp(x.key, y.key) > 0) {
64              return 1;
65          } else if (strcmp(x.key, y.key) < 0) {
66              return -1;
67          } else {
68              return 0;
69          }
70      }
71
72      void bin_insertion_sort(vector *v)
73      {
74          for (int i = 1; i < size(v); i++) {
75              int left = 0, right = i - 1, mid = -1;
76              while (left <= right) {
77                  mid = (left + right) / 2;
78                  if (comparator(v->begin[i], v->begin[mid]) >= 0) {
79                      left = mid + 1;
80                  } else {
81                      right = mid - 1;
82                  }
83              }
84              int k = left;
85              value_type temp = v->begin[i];
86              for (int j = i - 1; j >= k; j--) {
87                  v->begin[j + 1] = v->begin[j];
88              }
89              v->begin[k] = temp;
90          }
91      }
92
93      value_type bin_search(vector *v, char key[7])
94      {
95          value_type def = { "", "" };
96          int left = 0, right = size(v) - 1, mid;
97          while (left <= right) {
98              mid = (left + right) / 2;
```

```
99              if (strcmp(key, v->begin[mid].key) <= -1) {
100                 right = mid - 1;
101             } else if (strcmp(key, v->begin[mid].key) >= 1) {
102                 left = mid + 1;
103             } else {
104                 return v->begin[mid];
105             }
106         }
107         return def;
108     }
```

Листинг 6: main.c

```
1   #include <stdio.h>
2   #include <stdlib.h>
3   #include <string.h>
4   #include "vector.h"
5
6   int main(int argc, char const *argv[])
7   {
8       if (argc != 2) {
9           printf("Choose 1 file\n");
10          return 1;
11      }
12
13      FILE *test = fopen(argv[1],"r");
14
15      vector v;
16      create(&v);
17
18      value_type t;
19      while (fscanf(test, "%s%s", t.key, t.value) == 2) {
20          push(&v, t);
21      }
22
23  //  print(&v);
24      bin_insertion_sort(&v);
25      print(&v);
26
27      char find[7];
28      printf("\nSearch in text:\n");
29      while (1)
30      {
31          printf("Enter key: ");
32          if (scanf("%6s", find) == EOF) {
33              break;
34          }
35          if (strcmp(bin_search(&v, find).key, "")) {
36              printf("Your value: %s\n", bin_search(&v, find).value);
37          } else {
38              printf("This key no exists\n");
39          }
40
41      }
42
43      printf("\n");
44      destroy(&v);
45      fclose(test);
46
47      return 0;
48  }
```

Листинг 7: Makefile

```
1   CC = gcc
2   CFLAGS ?= -g -Wall -Wextra -pedantic -std=c99 -w -pipe -O3 -lm
```

```
3   main: main.o vector.o
4       $(CC) -o main main.o vector.o
5
6   main.o: main.c
7       $(CC) $(CFLAGS) -c main.c
8
9   vector.o: vector.c vector.h
10      $(CC) $(CFLAGS) -c vector.c
11
12  clean:
13      rm -rf *.o main
```

9 Вывод

Благодаря данной курсовой работе я научился работать с некоторыми алгоритмами сортировки и поиска.