**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

Лабораторная работа №3

По дисциплине «Программирование»

 Массивы указателей.

Группа: АВТ-041

Студент: Дударь М.А.

Преподаватель: Балакин В.В.

НОВОСИБИРСК 2021

1. **Постановка задачи:**

Необходимо реализовать функцию, которая читает из файла текст по словам и возвращает двухуровневый динамический массив указателей на строки на строки, содержащие слова из исходного файла, упорядоченные по алфавиту. Размерность массива указателей нижнего уровня задана, каждый массив указателей ограничен NULL. Очередная строка вставляется с сохранением порядка, в первом цикле просматривается массив указателей первого уровня и в каждом – элемент с индексом 0 второго уровня. Если его значение больше нового, то выполняется вставка в предыдущий массив указателей нижнего уровня. Если при вставке происходит переполнение, массив создается новый массив указателей, в который копируется половина указателей текущего.

1. **Алгоритм программы:**
2. Считывание данных из файла.
3. Запись слова в строки согласно алфавиту.
4. **Методы решения:**

Для решения задачи использовался массив указателей. Он удобен тем, что к любому слову можно обратиться через указатель, без его поиска в строке.

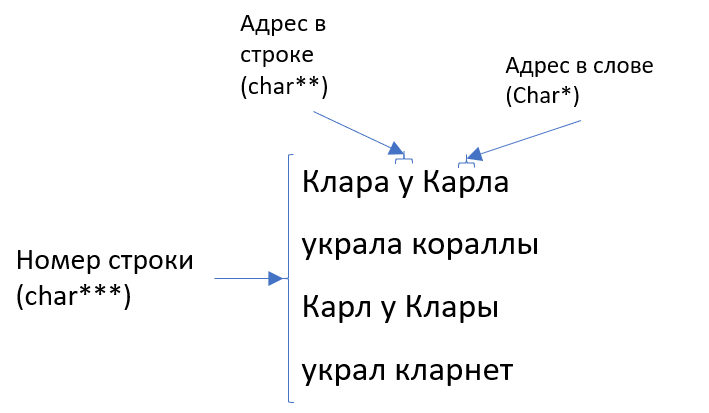


Рис. 1. Расположение слов в файле.

1. **Принцип работы:**

Функция вывода двухуровневого массива указателей.

void list\_print(char \*\*\*arr, const int \*height, const int \*width) {  
 for (int i = 0; i <= \*height; ++i) {  
 printf("\n");  
 for (int j = 0; j <= width[i]; ++j) {  
 printf("%s ", arr[i][j]);  
 }  
 }  
}

Функция чтения списка слов из файла. Используется цикл, который считывает слова из файла, пока это возможно (иначе выходит из цикла), причем вставляет слова в первый уровень массива указателей согласно алфавиту.

char \*\*\*File\_read(char \*\*\*arr) {  
 FILE \*fp;  
 char buf1[100], buf2[100];  
 char c;  
 int height = 0, len = 0, heightSize = 1;  
 int \*width = (int \*) malloc(sizeof(int));  
 if ((fp = fopen("read.txt", "r")) == NULL) {  
 printf("Can't open file 'read.txt'\n");  
 }  
  
 arr = (char \*\*\*) malloc(sizeof(char \*\*));  
 arr[0] = (char \*\*) malloc(sizeof(char \*));  
  
 while (fscanf(fp, "%s", buf1) == 1) {  
 width = (int \*) realloc(width, (height + 1) \* sizeof(int));  
 width[height] = len;  
 arr[height] = (char \*\*) realloc(arr[height], (len + 1) \* sizeof(char \*));  
 arr[height][len] = (char \*) malloc(sizeof(char));  
 arr[height][len] = (char \*) realloc(arr[height][len], strlen(buf1) \* sizeof(char));  
 if ((int) arr[height][0][0] >= (int) buf1[0]) {  
 for (int i = 0; i <= len; ++i) {  
 strcpy(buf2, arr[height][i]);  
 arr[height][i] = (char \*) malloc(strlen(buf1) \* sizeof(char));  
 strcpy(arr[height][i], buf1);  
 strcpy(buf1, buf2);  
 }  
 } else strcpy(arr[height][len], buf1);  
 len++;  
 fscanf(fp, "%c", &c);  
 if (c == '\n') {  
 height++;  
 len = 0;  
 if (height == heightSize) {  
 arr = (char \*\*\*) realloc(arr, 2 \* height \* sizeof(char \*\*));  
 heightSize = 2 \* height;  
 }  
 arr[height] = (char \*\*) malloc(sizeof(char \*));  
 }  
 }  
 list\_print(arr, &height, width);  
 fclose(fp);  
 return arr;  
}

1. **Вывод:**

В ходе выполнения задания была произведена работа с массивами указателей и файлами на языке C. Создан массив указателей, организована система данных, реализовано взаимодействие с ними, динамическое изменение их размерностей. Применена работа с файлами, считывание данных.

Приложение 1. Вывод программы.

Рис 2

Приложение 2. Текст программы.

#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
  
void list\_print(char \*\*\*arr, const int \*height, const int \*width) {  
 for (int i = 0; i <= \*height; ++i) {  
 printf("\n");  
 for (int j = 0; j <= width[i]; ++j) {  
 printf("%s ", arr[i][j]);  
 }  
 }  
}  
  
char \*\*\*File\_read(char \*\*\*arr) {  
 FILE \*fp;  
 char buf1[100], buf2[100];  
 char c;  
 int height = 0, len = 0, heightSize = 1;  
 int \*width = (int \*) malloc(sizeof(int));  
 if ((fp = fopen("read.txt", "r")) == NULL) {  
 printf("Can't open file 'read.txt'\n");  
 }  
  
 arr = (char \*\*\*) malloc(sizeof(char \*\*));  
 arr[0] = (char \*\*) malloc(sizeof(char \*));  
  
 while (fscanf(fp, "%s", buf1) == 1) {  
 width = (int \*) realloc(width, (height + 1) \* sizeof(int));  
 width[height] = len;  
 arr[height] = (char \*\*) realloc(arr[height], (len + 1) \* sizeof(char \*));  
 arr[height][len] = (char \*) malloc(sizeof(char));  
 arr[height][len] = (char \*) realloc(arr[height][len], strlen(buf1) \* sizeof(char));  
 if ((int) arr[height][0][0] >= (int) buf1[0]) {  
 for (int i = 0; i <= len; ++i) {  
 strcpy(buf2, arr[height][i]);  
 arr[height][i] = (char \*) malloc(strlen(buf1) \* sizeof(char));  
 strcpy(arr[height][i], buf1);  
 strcpy(buf1, buf2);  
 }  
 } else strcpy(arr[height][len], buf1);  
 len++;  
 fscanf(fp, "%c", &c);  
 if (c == '\n') {  
 height++;  
 len = 0;  
 if (height == heightSize) {  
 arr = (char \*\*\*) realloc(arr, 2 \* height \* sizeof(char \*\*));  
 heightSize = 2 \* height;  
 }  
 arr[height] = (char \*\*) malloc(sizeof(char \*));  
 }  
 }  
 list\_print(arr, &height, width);  
 fclose(fp);  
 return arr;  
}  
  
int main() {  
 char \*\*\*arr = NULL;  
 arr = File\_read(arr);  
 return 0;  
}