**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

отчет

**по лабораторной работе №3**

**по дисциплине «Программирование»**

Тема: Строки. Рекурсия. Циклы. Обход дерева.

|  |  |  |
| --- | --- | --- |
| Студент гр. 6303 |  | Шевченко Д.В. |
| Преподаватель |  | Берленко Т.А. |

Санкт-Петербург

2017

**Цель работы:**

Требуется написать программу, запускающуюся в корневой директории, содержащей одну папку с именем "**add**" или "**mul**" и вычисляющую с последующим выводом на экран результат выражения состоящего из чисел в поддиректориях по следующим правилам:

* Если в папке находится один или несколько текстовых файлов, то математическая операция определяемая названием папки (**add** = сложение, **mul** = умножение) применяется ко всем числам всех файлов в этой папке.
* Если в папке находится еще одна или несколько папок, то сначала вычисляются значения выражений, определяемые ими, а после используются уже эти значения.

Иерархия папок предназначенная для обработки программой составлена по следующим правилам:

* Название папок может быть только "**add**" или "**mul**"
* В папках могут находиться другие вложенные папки и/или текстовые файлы
* Текстовые файлы имеют произвольное имя с расширением **.txt**
* Содержимое текстовых файлов представляет собой строку, в которой через пробел записано некоторое количество целых чисел

**Описание работы программы:**

Программа представлена в виде трёх вспомогательных функций, и одной основной:

1. long long int DirMulAdd(char \* way, const char \* operation)

Основная рекурсивная функция, предназначенная для обработки директории, в частности содержащихся в ней файлов и поддиректорий. На вход подаётся путь, полученный в качестве аргумента командной строки, либо, при отсутствии аргументов, изначально заданный путь, в процессе своей работы извлекает числа из файлов вложенных в директорию, или же открывает поддиректории если таковые имеются, при переходе в поддиректорию так-же передается массив со строкой характеризующей операцию которую необходимо провести над всеми полученными в процессе обработки директории числами. Возвращаемое значение типа long long int – число, полученное в результате обработки всех файлов и поддиректорий директории, передаваемое в итерацию меньшей глубины.

1. Основная функция int main( int argc, char \* argv[] )

При наличии аргумента, переданного посредством командной строки, считает аргумент путем к корневому каталогу root, если в этом каталоге не создана требуемая иерархия файлов, работа программы завершается. Также, в случае отсутствия таковых аргументов, будет задействован путь к корневому каталогу по умолчанию. Также выполняется поиск папок с названиями mul или add, что определяет первую операцию, переданную в DirMulAdd. Также при вызове DirMulAdd передаётся полученный путь к корневой обрабатываемой директории, а результат её работы используется в качестве аргумента printf.

|  |
| --- |
| Приложение 1. Исходный код программы:  // lab №3 "Directory calculator" by Shevchenko Danil, 6303 group  #include <stdlib.h>  #include <stdio.h>  #include <string.h>  #include <dirent.h>  int sizeOfWay = 100;  const char mul[5] = { "mul/" };  const char add[5] = { "add/" };  long long int DirMulAdd(char \* way, const char \* operation);  int main(int argc, char \* argv[])  {  char \* way = (char\*)malloc(sizeof(char)\*sizeOfWay);  if (argc > 1)  strcat(way, argv[1]);  else  strcat(way, "/home/xex/Документы/Wtf\_dir\_test/root/");  DIR \*dir;  dir = opendir(way);  if (!dir)  {  printf("Dir on this way %s doesn't root\n", way);  return 0;  }  struct dirent \*entry;  while (entry = readdir(dir))  {  if (!strcmp(entry->d\_name, "mul"))  printf("> Result is: %lli\n", DirMulAdd(way, mul));  if (!strcmp(entry->d\_name, "add"))  printf("> Result is: %lli\n", DirMulAdd(way, add));  }  closedir(dir);  free(way);  return 0;  }  long long int DirMulAdd(char \* way, const char \* operation)  {  char opFlag = 0;  if (!strcmp(operation, mul))  opFlag = 1;  int wayLen = strlen(way);  long long int fileNum = opFlag;  DIR \*dir;  if (wayLen > (sizeOfWay - 40))  {  sizeOfWay += 100;  way = (char\*)realloc(way, sizeof(char)\*sizeOfWay);  }  strcat(way, operation);  struct dirent \*entry;  dir = opendir(way);  while (entry = readdir(dir))  {  if (strstr(entry->d\_name, "file"))  {  char buf = entry->d\_name[4];  entry->d\_name[4] = '\0';  if (!strcmp(entry->d\_name, "file"))  {  entry->d\_name[4] = buf;  strcat(way, entry->d\_name);  FILE \* file = fopen(way, "r");  int bufNum;  while (fscanf(file, "%d", &bufNum) > 0)  {  if (opFlag)  fileNum \*= bufNum;  else  fileNum += bufNum;  }  way[wayLen + 4] = '\0';  fclose(file);  }  }  if (!strcmp(entry->d\_name, "mul"))  {  if (opFlag)  fileNum \*= DirMulAdd(way, mul);  else  fileNum += DirMulAdd(way, mul);  }  if (!strcmp(entry->d\_name, "add"))  {  if (opFlag)  fileNum \*= DirMulAdd(way, add);  else  fileNum += DirMulAdd(way, add);  }  }  closedir(dir);  way[wayLen] = '\0';  return fileNum;  } |

**Составление отчёта и заливка решения на git:**

* По лабораторной работе составлен отчет Lr\_3\_orchet.docx и файл с кодом программы dir\_calc.c  
  В репозитории на Github создана от мастера ветка 2sem\_Shevchenko\_Danil\_Lab\_3 с помощью команды:

«git checkout -b 2sem\_Shevchenko\_Danil\_Lab\_3».

* Папка с файлами отчета и кода была помещена в ветку репозитория командами:  
  «git add Lab\_3\_Shevchenko\_Danil\_2\_sem»  
  «git commit»   
  «git push origin 2sem\_Shevchenko\_Danil\_Lab\_3»
* После чего был сделан request на ветку master.

**Вывод**: Были изучены методы рекурсивной обработки деревьев, образованных вложенными директориями, также были приобретены навыки работы с библиотекой dirent.h, исследованы представленные в ней методы, позволяющие обрабатывать данные о файлах, содержащихся в директории, которые впоследствии были использованы для решения поставленной задачи.