



Отчёт по лабораторной работе № 23 по курсу практикум на ЭВМ

студента группы М8О-106Б Суханова Егора Алексеевича, № по списку 21

Адреса www, e-mail, skype egor-suhanov2012@yandex.ru

Работа выполнена: "16" апреля 2020 г.

Преподаватель: ст. преп. Дубинин А.В.

Входной контроль знаний с оценкой

Отчёт сдан " " 20 г., итоговая оценка

Подпись преподавателя

1. Тема: Динамические структуры данных. Обработка деревьев.

2. Цель работы: Научиться реализовывать и обрабатывать деревья.

3. Задание (вариант № 11): Составить программу на языке программирования Си для построения и обработки дерева общего вида. Проверить монотонность убывания ширины уровня дерева.

4. Оборудование (лабораторное):

ЭВМ, процессор, имя узла сети с ОП МБ
НМД ГБ. Терминал адрес. Принтер
Другие устройства

Оборудование ПЭВМ студента, если использовалось:

Процессор Intel® Pentium® 3550M, ОП 8192 МБ, ТН 250 ГБ. Монитор
Другие устройства

5. Программное обеспечение (лабораторное):

Операционная система семейства, наименование версия
Интерпретатор команд версия
Система программирования версия
Редактор текстов версия
Утилиты операционной системы

Прикладные системы и программы

Местонахождения и имена файлов программ и данных

Программное обеспечение ЭВМ студента, если использовалось:

Операционная система семейства windows, наименование Windows 10 версия 1909
Интерпретатор команд GNU bash версия 4.4.20
Система программирования MiKTeX версия 2.9.7364
Редактор текстов версия
Утилиты операционной системы

Прикладные системы и программы

Местонахождения и имена файлов программ и данных

6. Идея, метод, алгоритм решения задачи (в формах: словесной, псевдокода, графической [блок-В
Весь исходный код находится в директории src/, его компиляция производится с помощью Makefile
доступны два варианта: make / make release и make debug (последний отличается от первого ключом -g)

Программа будет состоять из следующих модулей:

Файл main.c

-> точка входа в программу

-> обработка ввода и выполнение команд

```
while(cmd.type != CMD_STOP){  
    cmd = get_command(stdin, stdout);    // Считывание новой команды  
    handle_command(&cmd, &root, stdout); // Обработка команды  
}
```

где cmd – структура, которая хранит код команды и её параметры. Считывание — считывает строку из потока stdin и парсит команду, получая cmd.type, cmd.path и cmd.val. handle_command обрабатывает команду, в зависимости от cmd.type.

Модуль tree

->Реализация бинарного дерева общего вида с типом значений unit

Модуль queue

->Реализация очереди на макросах.

Модуль log

->Немного макросов для логирования.

Модуль enum

->Тип unit представляет собой перечисление статусов в учебном заведении.

->Функции для конвертирования строк в unit и обратно

Алгоритм работы: (я не буду описывать каждую функцию, опишу самое важное)

main.c:

Пока текущая команда (cmd) не команда выхода, выполнять:

> считать команду с потока ввода и занести её в cmd

> обработать команду cmd

Модуль tree:

> функция t_func (идея)

Заведём флаг is_monotonous, который будет говорить, монотонна ли функция

Заведём переменные, которые хранят кол-во вершин в текущем и прошлом уровнях

Запускаем bfs с двумя очередями от корня дерева

Считаем каждый обрабатываемый узел

Если закончилось текущая очередь, то мы меняем текущую и очередь следующего уровня местами

Заодно сравниваем кол-во вершин в текущем и прошлом уровне. Обновляем флаг монотонности

Если флаг монотонности равен "ложь", то выходим из bfs

меняем местами переменные, которые хранят кол-во вершин в текущем и прошлом уровнях

Пока текущая очередь не пуста возвращаемся к 1 ому пункту

Печатаем результат работы

Эффективность:

По времени $O(n)$ - нужно пройти все узлы дерева в худшем случае

По памяти $O(n)$ - все узлы находятся на одном уровне

7. Сценарий выполнения работы [план работы, первоначальный текст программы в черновике (можно на отдельном листе) и тесты, либо соображения по тестированию].

ходе этапа планирования были выделены следующие задачи:

1> Реализовать ввод

2> Реализовать модуль tree

2.1> Добавление нового узла

2.2> Визуализация дерева

2.3> Удаление узла

2.4> Вычисление функции

3> Реализовать обработку команд

4> Протестировать (И исправить обнаруженные ошибки)

Пункты 1-7 отчёта составляются **строго до** начала лабораторной работы.

Допущен к выполнению работы. Подпись преподавателя _____

8. Распечатка протокола (подклеить листинг окончательного варианта программы с текстовыми примерами, подписанный преподавателем)

9. Дневник отладки должен содержать дату и время сеансов отладки, и основные ошибки (ошибки в сценарии и программе, не стандартные операции) и краткие комментарии к ним. В дневнике отладки приводятся сведения об использовании других ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

№	Лаб. или дом.	Дата	Время	Событие	Действие по исправлению	Примечание

10. Замечание автора по существу работы: _____

11. Выводы: Деревья общего вида могут быть использованы для хранения и обработки данных с иерархией.

12. Недочеты, допущенные при выполнении задания, могут быть устранены следующим образом:

Подпись студента _____