

Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Высшая школа интеллектуальных систем и суперкомпьютерных технологий

Отчёт по лабораторной работе № 5

Дисциплина: Низкоуровневое программирование

Тема: Программирование на языке С

Вариант: 1

Выполнил студент гр. 3530901/90002 _____ Д. Е. Бакин
(подпись)

Принял старший преподаватель _____ Д. С. Степанов
(подпись)

“ ____ ” _____ 2021 г.

Санкт-Петербург
2021

Постановка задачи:

1. Разработать статическую библиотеку, реализующую определенный вариантом задания абстрактный тип данных.
2. Разработать демонстрационную программу – консольное приложение, обеспечивающее ввод данных из файла (файлов), их обработку и вывод в файл (файлы); имена файлов передаются в качестве параметров командной строки.

Требования к ПО:

1. Язык разработки – С.
2. Реализация абстрактного типа данных должна использовать динамическое выделение памяти, при этом должна быть предусмотрена функция деинициализации, обеспечивающая освобождение всей выделенной памяти.
3. Библиотека и демонстрационная программа должны быть снабжены модульными тестами.
4. Разработанный исходный код должен компилироваться gcc без ошибок и предупреждений со следующими параметрами: `-std=c11 -pedantic -Wall -Wextra`.
5. Сборка библиотеки, демонстрационной программы и модульных тестов должна осуществляться утилитой `make`.

Вариант:

Дерево (с узлами произвольной степени). Каждый узел может иметь разное количество потомков. Содержимое узла - целое число.

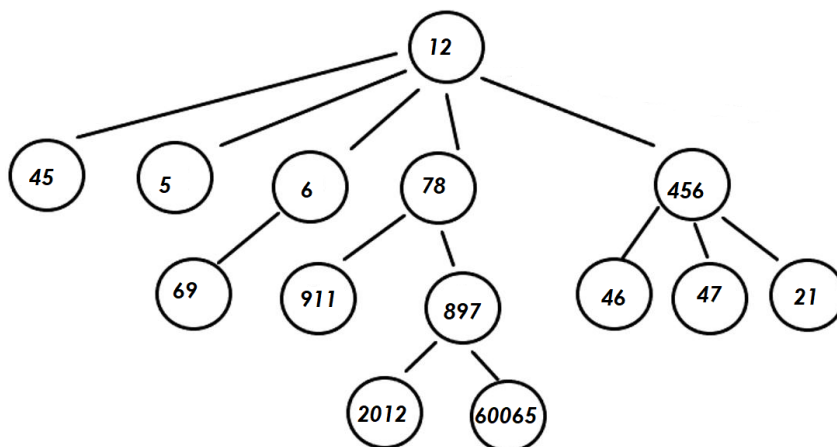


Рис. 1 Пример дерева с узлами произвольной степени.

Описание API библиотеки:

Дерево было реализованно по динамической структуре узла:

int data – значение узла

*Node *down* – указатель на самого левого ребенка

*Node *right* – указатель на следующий узел с таким же родителем

Структура дерева содержит только ссылку на корень дерева (узел).

Библиотека имеет следующие функции:

struct tree create(int x); - создание корневого узла с заданным значением.

*Node *addNode(int x, Node *tree);* - добавление ребенка к узлу.

*void removeNode(Node *node, Tree *tree);* - удаление узла по ссылке в дереве

вместе со всеми его потомками.

*Node *findNode(int x, Tree *tree);* - нахождение узла в дереве с определенным параметром.

*void tprint(Tree *node, int d);* - вывод дерева в консоль.

*int calcChild(Node *node);* - подсчет количества детей узла.

*Node *findMax(Tree *tree);* - поиск узла в дереве с максимальным значением.

Демонстрационная программа:

Представляет собой демонстрацию функционала библиотеки. Для использования программы, ее нужно собрать с помощью Makefile'ов.

Сначала идет сбор библиотеки, потом сбор приложения, после чего появляется сборочный файл.

Вывод:

Была написана статически линкуемая библиотека дерева с произвольным количеством узлов. Также была написана демонстрационная программа, показывающая функционал библиотеки.