Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
ТОМСКИЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)
Кафедра комплексной информационной безопасности электронно-вычислительных систем (КИБЭВС)

# ДВУСВЯЗНЫЕ СПИСКИ

Отчёт по практической работе №4 по дисциплине «Технологии и методы программирования»

Студент гр.728-2
_____ Полонский Е. В.
24 марта 2020 г.

Руководитель
Аспирант кафедры КИБЭВС
_____ Перминов П. В.
_____

Томск 2020

# 1 Введение

Целью данной работы является закрепление теоритических знаний об двусвязных списках, реализация двусвязного списка на практике.

Задание: Написать на языке программирования C программу, позволяющую работать с двусвязными списками.

# 2 Ход работы

Для реализации списка необходимы две структуры:

– list – непосредственно сам список, с полями head, tail и length – указателями на первый и поледний элементы в списке, а также длина списка;

– node – элемент списка, содержит в себе 3 поля, value – значение элемента и next – указатель на следующий элемент списка и prev – указатель на предыдущий элемент списка.

Также необходимо реализовать несколько функций и процедур:

– void init(list *l) – инициализирует пустой список;

– void is_empty(list *l) – проверяет список на отсутствие элементов;

– void clean(list *l) – удаляет все элементы списка;

– node *find_last(list *l) – вспомогательная функция, которая проходит по всему списку и возвращает последний элемент;

– node *find(list *l, int value, bool revert) – находит и возвращает первый элемент со значением value, если указан флаг revert, начинает поиск с конца;

– int push_back(list *l, int value) – вставляет элемент в конец списка;

– push_front(list *l, int value) – вставляет элемент в начало списка;

– node *go_to_index(list *l, int i) – возвращает элемент по индексу;

– int insert_after(list *l, int i, int value) – вставка значения после указанного узла (по индексу);

– int insert_before(list *l, int index, int value) – вставка элемента перед указанным узлом (по индексу);

– void __remove_node(list *l, node *nodeptr) – удаляет указанный элемент;

– int remove_first(list *l, int value) – удаляет первый элемент из списка с указанным значением;

– int remove_last(list *l, int value) – удаляет последний элемент из списка с указанным значением;

– void print(list *l) – выводит все значения в списке в прямом порядке, через пробел.

– void print_invers(list *l) – выводит все значения в списке в обратном порядке, через пробел.

Для реализации некоторых функций необходима работа с кучей, а именно выделение чанка памяти при добавлении нового элемента в список (процедура – calloc()). И удаление чанка, соответственно при удалении элемента списка (процедура free()).

Весь исходный код программы можно посмотреть на гите.

Для компиляции программ использовалась команда gcc имя файла

Далее исходные коды программ были запушены на гитлаб командой git push. Все пайплайны были пройдены успешно (Рисунок 2.1).

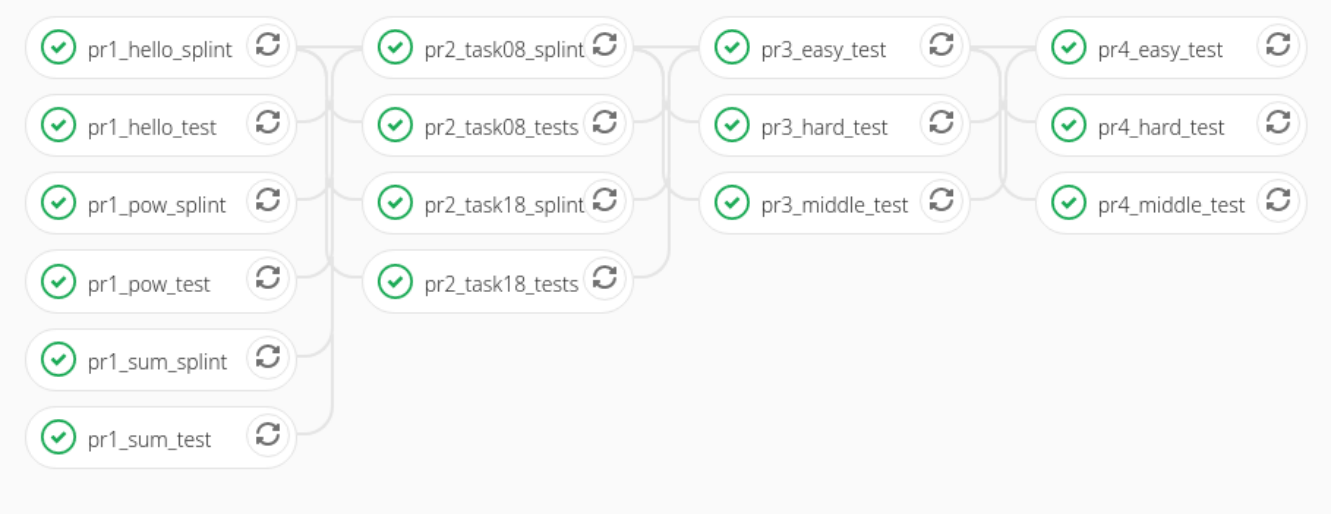


Рисунок 2.1 – Пройденные пайплайны

# 3 Заключение

В результате выполнения практической работы были закреплены теоритические знания об двусвязных списках, написана программа, реализующая двусвязный список.