**БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

Международный Институт Дистанционного Образования

Кафедра “Информационные системы и технологии”

Лабораторная работа номер 4 по дисциплине «Системное программирование»

Тема работы:

**«Программирование на Си»**

Выполнил:   
студент 3 курса, гр. 41703120  
Реут Владислав Леонидович

Проверил: Бумай А.Ю.

Минск 2022

**ВВЕДЕНИЕ**

Язык программирования С (си) является одним из самых популярных и распространенных языков. Он представляет компилируемый язык программирования общего назначения со статической типизацией, разработанный в 1969—1973 годах в компании Bell Labs программистом Деннисом Ритчи (Dennis Ritchie).

Язык С нередко называют языком программирования "среднего уровня" или даже "низкого уровня", так как он сочетает элементы языков высокого уровня с функциональностью и производительностью ассемблера и работает близко к аппаратной части компьютера. В итоге мы можем манипулировать данными на низком уровне и при этом использовать высокоуровневые конструкции для управления работы программы.

Первоначально язык С предназначался для написания операционной системы Unix. Впоследствии Си стал одним из популярных языков, а его основной сферой применения стало системное программирование, в частности, создание операционных систем, драйверов, различных утилит, антивирусов и т.д. К слову сказать, Linux большей частью написан на Си. Однако только системным программированием применение данного языка не ограничивается. Данный язык можно использовать в программах любого уровня, где важны скорость работы и производительность. Так, мы можем писать с помощью Си и прикладные приложения, и даже веб-сайты (используя технологию CGI - Common Gateway Interface). Но, конечно, для создания графического интерфейса и веб-приложений, как правило, выбираются более подходящие инструменты и технологии, но тем не менее круг использования Си довольно широк. Это в немалой степени определило популярность языка. Например, в известном рейтинге языков программирования TIOBE язык С долгое время уверенно удерживает второе место.

Несмотря на большие возможности язык Си одновременно довольно прост. Он не содержит много конструкций, библиотек, его легко осваивать и изучать. Поэтому нередко его выбирают в качестве языка для изучения в целом программированию.

Си является компилируемым языком, а это значит, что компилятор транслирует исходный код на Си в исполняемый файл, который содержит набор машинных инструкций. Но разные платформы имеют свои особенности, поэтому скомпилированные программы нельзя просто перенести с одной платформы на другую и там уже запустить. Однако на уровне исходного кода программы на Си обладают переносимостью, а наличие компиляторов, библиотек и инструментов разработки почти под все распространенные платформы позволяет компилировать один и тот же исходный код на Си в приложения под эти платформы.

Развитие Си оказало большое влияние в целом на развитие языков программирования. В частности, его синтаксис стал основой для таких языков как С++, С#, Java, PHP, JavaScript. Особо следует сказать про связь с C++. C++ напрямую произошёл от Си. Но впоследствии их развитие происходило отдельно друг от друга, и даже появилась несовместимость между ними. Стандарт C99 добавил в язык Си ряд конфликтующих с C++ особенностей. В итоге в настоящее время оба языка являются фактически самодостаточными и развиваются независимо.

**Вариант №5**

Имеется элемент стека (дисциплина обслуживания LIFO):

struct Stack

{

char \* data; //указатель на данные

Stack \* prev; //указатель на предыдущий элемент

}

\* top; //указатель на вершину стека

Определите следующие функции:

void Push(Stack\*\* stack, char\* data); // поместить данные в стек;

char\* Pop(Stack\*\* stack); // извлечь данные из стека (при этом элемент удаляется из стека);

PrintStack(Stack\* stack); // вывод на экран содержимого стека.

Листинг:

typedef struct Stack

{

char \* data;//указатель на данные

struct Stack \* prev;//указатель на предыдущий элемент

}Stack;

void Push(Stack\*\* stack, char\* data) // поместить данные в стек;

{

Stack \*tmp = (Stack\*)malloc(sizeof(Stack));

tmp->data = data;

tmp->prev = (\*stack) ? \*stack : NULL;

\*stack = tmp;

}

void Pop(Stack\*\* stack)

{

if (!(\*stack))//если нечего удалять

return;

Stack \*tmp = \*stack;//создаем указатель на вершину

\*stack = (\*stack)->prev;//вершину сдвигаем вниз

free(tmp);//очищаем память (удаляем старую вершину)

}

void PrintStack(Stack \*stack) /\* Печать списка \*/

{

Stack \*tmp = stack;

while (tmp)

{

printf("%s\n", tmp->data);

tmp = tmp->prev;

}

}

int main()

{

Stack \*top = NULL;

Push(&top, "abc");

Push(&top, "efg");

Push(&top, "mmm");

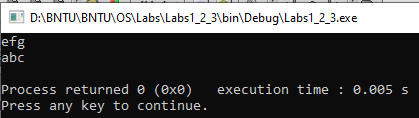
Pop(&top);

PrintStack(top);

return 0;

}

Результат работы функции:



**СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ**

1. ЛАБОРАТОРНЫЙ ПРАКТИКУМ ПО КУРСУ СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ
2. Сайт metanit.com