**Язык программирования C++**

*Отчет*

|  |  |
| --- | --- |
| Лабораторная работа 1 | *17.09.2019* |

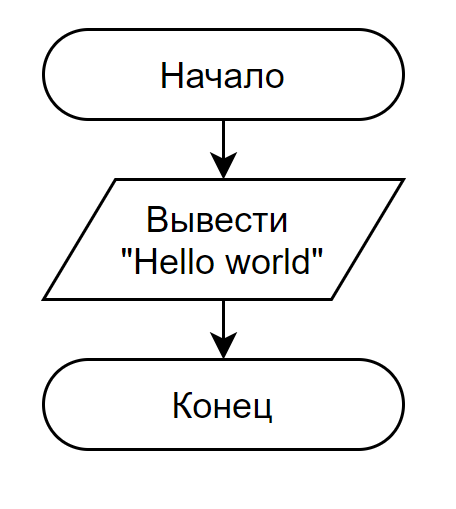
Основы программирования для Linux

*Задача №1*

**1. Условия задачи**

*Раздел №1.1 из курса“Основы программирования для Linux”.* Вывести “Hello world” в терминале.

**2. Алгоритм. Блок-схема**

****

**3. Исходный код программы**

#include <stdio.h>

int main()

{

printf("Hello world\n");

return 0;

}

**4. Формат входных и выходных данных**

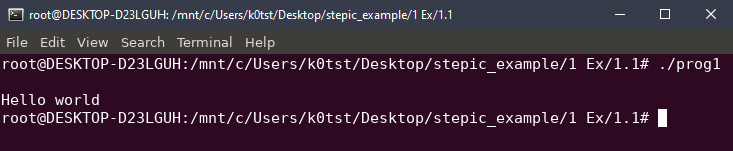
Формат входных данных

Программа ничего не принимает на входе.

Формат выходных данных

Программа выводит на консоль “Hello world”.

**5. Результат работы программы**



**6. Выводы и комментарии к решению задачи**

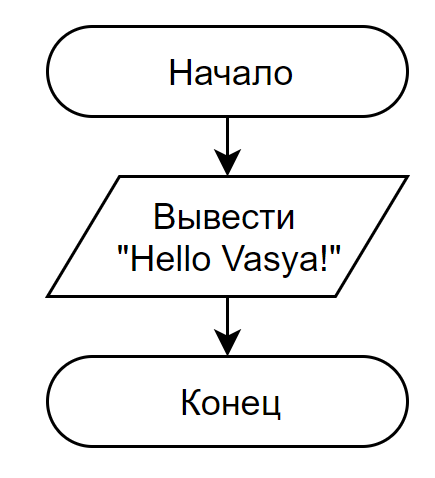
* Самая первая программа

*Задача №2*

**1. Условия задачи**

*Раздел №1.2 из курса “Основы программирования для Linux”.* Вывести “Hello Vasya!” в терминале.

**2. Алгоритм. Блок-схема**

****

**3. Исходный код программы**

//main.c

#include "hello.h"

int main()

{

hello\_message("Vasya");

return 0;

}

//hello.c

#include "hello.h"

#include <stdio.h>

void hello\_message(const char\* name)

{

printf("Hello %s!\n", name);

}

//hello.h

#ifndef \_\_HELLO\_\_

#define \_\_HELLO\_\_

#ifdef \_\_cplusplus

extern "C" void hello\_message(const char \*name);

#else

void hello\_message(const char \*name);

#endif

#endif

**4. Формат входных и выходных данных**

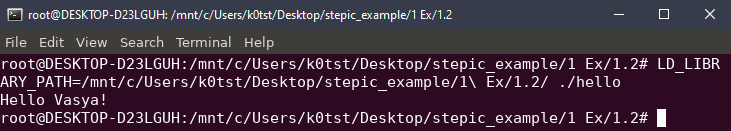
Формат входных данных

Программа ничего не принимает на входе.

Формат выходных данных

Программа выводит на консоль “Hello Vasya!” с помощью подключаемых библиотек.

**5. Результат работы программы**



**6. Выводы и комментарии к решению задачи**

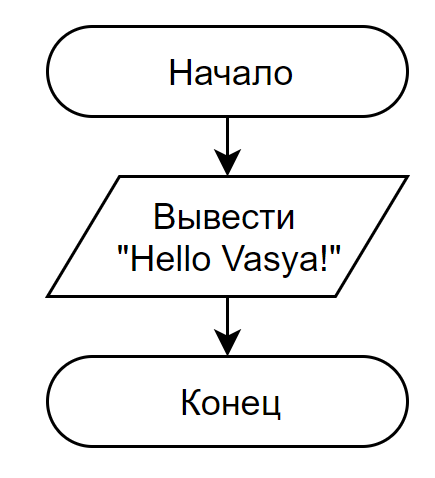
* Необходимо заранее подключить библиотеку перед запуском программы
* В главном файле выполняется функция, для доступа к которой необходимо подключить файл hello.c

*Задача №3*

**1. Условия задачи**

*Раздел №1.3 из курса “Основы программирования для Linux”.* Вывести “Hello Vasya!” в терминале.

**2. Алгоритм. Блок-схема**

****

**3. Исходный код программы**

//main.c

#include <stddef.h>

#include <stdbool.h>

#include <stdio.h>

#include <dlfcn.h>

void (\*hello\_message)(const char \*);

bool init\_library()

{

void \*hdl = dlopen("./libHello.so", RTLD\_LAZY);

if(NULL == hdl)

return false;

hello\_message = (void(\*)(const char \*))dlsym(hdl, "hello\_message");

if(NULL == hello\_message)

return false;

return true;

}

int main()

{

if(init\_library())

hello\_message("Vasya");

else

printf("Library was not loaded\n");

return 0;

}

//hello.c

#include "hello.h"

#include <stdio.h>

void hello\_message(const char\* name)

{

printf("Hello %s!\n", name);

}

//hello.h

#ifndef \_\_HELLO\_\_

#define \_\_HELLO\_\_

#ifdef \_\_cplusplus

extern "C" void hello\_message(const char \*name);

#else

void hello\_message(const char \*name);

#endif

#endif

**4. Формат входных и выходных данных**

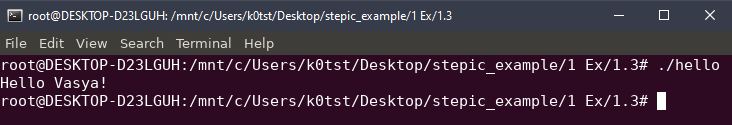
Формат входных данных

Программа ничего не принимает на входе.

Формат выходных данных

Программа выводит на консоль “Hello Vasya!” с помощью подключаемых библиотек.

**5. Результат работы программы**



**6. Выводы и комментарии к решению задачи**

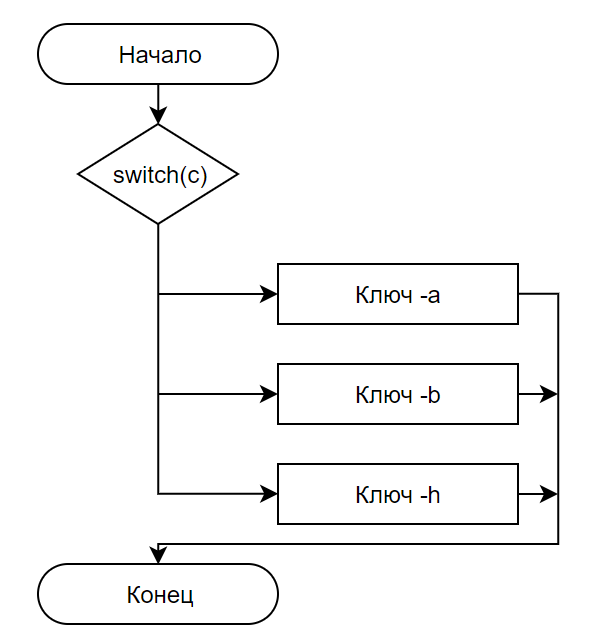
* Нет необходимости заранее подключать библиотеку
* В главном файле выполняется функция, для доступа к которой необходимо подключить файл hello.c

*Задача №4*

**1. Условия задачи**

*Раздел №1.4 из курса “Основы программирования для Linux”.* Обработка параметров запуска программы.

**2. Алгоритм. Блок-схема**

****

**3. Исходный код программы**

//getopt.c

﻿#include <unistd.h>

#include <stdlib.h>

#include <stdio.h>

int

main(int argc, char \*argv[])

{

int flags, opt;

int nsecs, tfnd;

nsecs = 0;

tfnd = 0;

flags = 0;

while ((opt = getopt(argc, argv, "nt:")) != -1) {

switch (opt) {

case 'n':

flags = 1;

break;

case 't':

nsecs = atoi(optarg);

tfnd = 1;

break;

default: /\* '?' \*/

fprintf(stderr, "Usage: %s [-t nsecs] [-n] name\n",

argv[0]);

exit(EXIT\_FAILURE);

}

}

printf("flags=%d; tfnd=%d; optind=%d\n", flags, tfnd, optind);

if (optind >= argc) {

fprintf(stderr, "Expected argument after options\n");

exit(EXIT\_FAILURE);

}

printf("name argument = %s\n", argv[optind]);

/\* Other code omitted \*/

exit(EXIT\_SUCCESS);

}

//getopt\_long.c

﻿#include <stdio.h> /\* for printf \*/

#include <stdlib.h> /\* for exit \*/

#include <getopt.h>

int

main(int argc, char \*\*argv)

{

int c;

int digit\_optind = 0;

while (1) {

int this\_option\_optind = optind ? optind : 1;

int option\_index = 0;

static struct option long\_options[] = {

{"add", required\_argument, 0, 0 },

{"append", no\_argument, 0, 0 },

{"delete", required\_argument, 0, 0 },

{"verbose", no\_argument, 0, 0 },

{"create", required\_argument, 0, 'c'},

{"file", required\_argument, 0, 0 },

{0, 0, 0, 0 }

};

c = getopt\_long(argc, argv, "abc:d:012",

long\_options, &option\_index);

if (c == -1)

break;

switch (c) {

case 0:

printf("option %s", long\_options[option\_index].name);

if (optarg)

printf(" with arg %s", optarg);

printf("\n");

break;

case '0':

case '1':

case '2':

if (digit\_optind != 0 && digit\_optind != this\_option\_optind)

printf("digits occur in two different argv-elements.\n");

digit\_optind = this\_option\_optind;

printf("option %c\n", c);

break;

case 'a':

printf("option a\n");

break;

case 'b':

printf("option b\n");

break;

case 'c':

printf("option c with value '%s'\n", optarg);

break;

case 'd':

printf("option d with value '%s'\n", optarg);

break;

case '?':

break;

default:

printf("?? getopt returned character code 0%o ??\n", c);

}

}

if (optind < argc) {

printf("non-option ARGV-elements: ");

while (optind < argc)

printf("%s ", argv[optind++]);

printf("\n");

}

exit(EXIT\_SUCCESS);

}

**4. Формат входных и выходных данных**

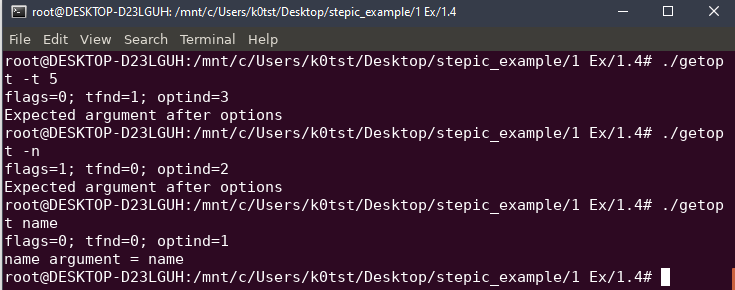
Формат входных данных

Программа ничего не принимает на входе.

Формат выходных данных

Программа выводит на консоль “Hello Vasya!” с помощью подключаемых библиотек.

**5. Результат работы программы**



**6. Выводы и комментарии к решению задачи**

* Использование “switch” облегчает работу с большим количеством ключей запуска

*Выводы по лабораторной работе 1*

**1.** Рассмотрены 4 лекционные задач по теме «*Основы программирования для Linux*»:

1) вывод “Hello world”,

2) работа с функциями и подключение динамической библиотеки,

3) работа с функциями и подключение динамической библиотеки в коде программы,

4) обработка параметров запуска программы.

СПИСОК ИСТОЧНИКОВ И ЛИТЕРАТУРЫ

1. Основы программирования для Linux. – Режим доступа: <https://stepik.org/lesson/26302/step/1?unit=8180>, свободный. Загл. с экрана. – Дата обращения: 10.09.2019