**Лабораторная работа №1**

1. Теоретические сведения.

Поток (thread) определяет последовательность исполнения кода в процессе. При инициализации процесса система всегда создает первичный поток. Каждый поток начинает выполнение с некоей входной функции. В первичном потоке таковой является *main.* Если Вы хотите создать вторичный поток, в нем тоже должна быть входная функция.

**Создание потока:**

Функция pthread\_create, описанная в заголовочном файле <pthread.h>, позволяет создать новый поток и начать его выполнение с указанной функции. Прототип этой функции:

#include <pthread.h>

int pthread\_create(pthread\_t \*thread, const pthread\_attr\_t \*attr,

void \*(\*start\_routine) (void \*), void \*arg);

Параметры:

* + thread — возвращаемое значение, идентификатор созданного потока.
  + attr — указатель на структуру с атрибутами создаваемой задачи.
  + start\_routine — указатель на функцию, с которой надо начать выполнение потока.
  + arg — аргумент этой функции.

Когда создан новый поток ему нужно отдать управление, для этого существует функция int pthread\_join(pthread\_t pthread, void \*\*thread\_return)

Параметры:

* + thread — идентификатор потока, завершения которого надо ожидать.
  + thread\_return — указатель на переменную, где будет сохранено значение задачи (может быть 0, если это не требуется)

Замечание.

При использовании потоков, программу нужно компилировать следующим образом:

g++ -o main.out main.cpp -lpthread

2. Пример использования:

#include <stdio.h>

#include <pthread.h>

/\*Функция потока \*/

void \*thread\_f(void \*arg)

{

printf("hello! \n");

printf("It is thread -- %i \n",arg);

}

void main\_evalution(void)

{

/\* Описываем переменную-поток \*/

pthread\_t thread;

int arg = 1;

/\* создаем поток \*/

pthread\_create(&thread, NULL, thread\_f,(void \*)arg);

/\* Останавливаем первичный поток, запускаем новый поток\*/

pthread\_join(thread, 0);

}

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

{

printf("Hello. It is First Thread!!!\n");

main\_evalution();

printf("By-By\n");

}

3. Задание:

1. Написать программу, которая будет запускать несколько потоков. Каждый поток должен проработать какое-то время, выдать информацию на экран.
2. Изменить программу таким образом, чтобы поток мог принимать и изменять аргумент, первичный поток передает аргумент, принимает его новое значение и распечатывает.