

Требования к работам

Все работы должны удовлетворять следующим условиям:

1. Программа должна использовать технологию CUDA (MPI, OpenMP).
2. Языки программирования: С или С++. При проверке используется компилятор nvcc версии 7.0(g++ версии 4.8.4) на 64-х битной Ubuntu 14.04 LTS. Параметры графического процессора:

```
Compute capability : 6.1
Name               : GeForce GTX 1050
Total Global Memory : 2096103424
Shared memory per block : 49152
Registers per block  : 65536
Warp size           : 32
Max threads per block : (1024, 1024, 64)
Max block           : (2147483647, 65535, 65535)
Total constant memory : 65536
Multiprocessors count : 5
```
3. Использовать в программах готовые библиотеки алгоритмов (thrust, cublas, cufft и др.), если не указано иного, запрещается.
4. **Формат ввода и вывода строго регламентирован** в каждой лабораторной работе.
5. Программа не должна создавать лишних (временных) файлов во время работы.
6. Везде, если не указано иного, подразумевается, что данные вводятся из **файла стандартного ввода (stdin)**, а выводятся в **файл стандартного вывода (stdout)**.
7. Программа не должна ни при каких условиях завершиться аварийно — если в какой-то момент времени программа не может быть выполнена по каким-либо причинам (отсутствие оперативной/графической памяти, проблемы с разбором входных данных и т.п.) она должна напечатать на стандартный вывод строку в формате ERROR: и дальше описание возникшей проблемы в свободной форме, после чего завершиться с кодом возврата 0.
8. Программа не должна пытаться навредить проверяющей системе (копировать / пересылать / удалять файлы, вешать систему и тп). Преднамеренные и непреднамеренные попытки будут строго караться.

Задания проверяются автоматически.

Компиляция производится командой (если не указано иного):

```
nvcc --std=c++11 -Werror cross-execution-space-call -lm
```

Сроки сдачи лабораторных работ.

На оценку 5: необходимо получить "ОК" в течении **двух недель** после даты выдачи ЛР.

На оценку 4: необходимо получить "ОК" в течении **четырёх недель** после даты выдачи ЛР.

На оценку 3: необходимо получить "ОК" до конца семестра.

На оформление отчета и защиту лабораторной работы дополнительно дается **одна неделя**.

Однократно можно сменить вариант ЛР с потерей одного балла.

Требования к отчету. Каждый отчет должен быть оформлен согласно выданному шаблону и прислан в формате **pdf** на checkpgp@gmail.com как минимум **ЗА СУТКИ (24 часа)** до непосредственной защиты лабораторной работы. В конце семестра все отчеты необходимо распечатать и сшить в одну папку.

Сдавать написанные не вами лабораторные и отчеты **категорически запрещено**. За попытку сдачи плагиата -- лишение автомата. Все присланные работы проходят проверку на антиплагиат.

Проверка лабораторных работ

В первую очередь, чтобы можно было отправлять на проверку программы, нужно создать gpg-ключи, и **лично** предоставить преподавателям открытый ключ.

Для проверки лабораторной работы, необходимо отправить исходный код программы и её gpg-подпись (файл с расширением .asc) на checkpgp@gmail.com. В теме письма указать "pgp:n", где n - номер ЛР. **Аналогично для отчетов**. Если программа состоит из нескольких файлов то необходимо написать Makefile и упаковать все в архив (пример далее).

Сгенерировать ключи:

```
gpg --gen-key
```

При создании ключа необходимо указать **фамилию и имя в транслитерации в нижнем регистре**, в виде **surname-name** (пример: ivanov-ivan), а также, ту **почту**, с которой будет выполняться отправка на проверку работ. Тип ключа выбрать: (1) **"RSA и RSA (по умолчанию)"**.

Посмотреть список своих ключей: `gpg --list-secret-keys`

Экспортировать открытый ключ: `gpg -a --export > surname-name.asc`

Создать подпись файла lab.cu: `gpg -ab lab.cu`

Если ключей несколько то при экспорте и создании подписи необходимо указать key id.

Экспортировать открытый ключ: `gpg -a --export "XXXXXXXX" > surname-name.asc`

Создать подпись файла lab.cu: `gpg -u "XXXXXXXX" -ab lab.cu`

где XXXXXXXX - key id.

Упрощенные варианты. Студент по своему желанию может выбрать упрощенные варианты ЛР (варианты "на два"). Варианты "на два" в четвертой и пятой ЛР связаны: выбор упрощенного варианта в одной из этих ЛР означает выбор такого же варианта и в другой ЛР. Аналогично для группы ЛР №7, №8 и №9. При выборе упрощенных вариантов максимальная оценка на экзамене - три балла.

Оценка автоматом за экзамен.

Оценка автоматом за экзамен - это средняя оценка* за ЛР округленная к ближайшему целому при условиях:

1. **Присутствие** как минимум на половине лекций (первое занятие не учитывается).
2. **Отсутствие** попыток сдать плагиат.
3. Все ЛР выполнены из основного списка вариантов (варианты не "на два").

Важно: "ОК" - от чекера не означает что ЛР принята. ЛР принята только после появления соответствующей отметки ("ОК") в гугл-таблице. После того как программа прошла все тесты (Вы получили на почту от чекера "ОК"), её исходный код просматривается вручную преподавателями на предмет соответствия всем требованиям задания. В случае если выявлены несоответствия (например: реализован другой алгоритм, неправильно выполнено распараллеливание и тд.) об этом сообщается студенту для последующего переделывания ЛР.

*При вычислении средней оценки учитываются необязательные ЛР (для ПГП - Доклад, для ПОД - OpenGL). Если соответствующая ЛР не выполнена, то оценка берется равная нулю.

Пример упаковки программы в архив и создания её электронной подписи:

```
sasha@work:~/mpi/7$ cd lab7/
sasha@work:~/mpi/7/lab7$ ls
makefile mpi7.cpp
sasha@work:~/mpi/7/lab7$ cat makefile
CC = /usr/local/bin/mpic++
CFLAGS = --std=c++11 -fopenmp -pedantic -Wall -Werror -Wno-sign-compare -Wno-long-long -lm
SOURCES = mpi7.cpp
BIN = lab7
all:
    $(CC) $(CFLAGS) -o $(BIN) $(SOURCES)
sasha@work:~/mpi/7/lab7$ make
/usr/local/bin/mpic++ --std=c++11 -fopenmp -pedantic -Wall -Werror -Wno-sign-compare -Wno-long-long
-lm -o lab7 mpi7.cpp
sasha@work:~/mpi/7/lab7$ ls
lab7 makefile mpi7.cpp
sasha@work:~/mpi/7/lab7$ rm lab7
sasha@work:~/mpi/7/lab7$ cd ..
sasha@work:~/mpi/7$ tar -cvf lab7.tar lab7
lab7/
lab7/mpi7.cpp
lab7/makefile
sasha@work:~/mpi/7$ ls
lab7 lab7.tar
sasha@work:~/mpi/7$ gpg -ab lab7.tar
```

Необходима фраза-пароль для доступа к закрытому ключу пользователя: "alexander-morozov
<alex-icex@mail.ru>"
2048-битный ключ RSA, ID 7E3A1A1E, создан 2016-06-08

```
sasha@work:~/mpi/7$ ls
lab7 lab7.tar lab7.tar.asc
sasha@work:~/mpi/7$
```

Важно чтобы имя исполняемого файла совпадало с именем каталога и названием архива.

Вместо утилиты tar можно использовать zip: sasha@work:~/mpi/7\$ zip lab7.zip lab7/*
или утилиту rar: sasha@work:~/mpi/7\$ rar a lab7.rar lab7/*

При написании makefile необходимо указывать следующие пути к компиляторам:

```
/usr/local/cuda/bin/nvcc
/usr/local/bin/mpic++
```