

Введение в распараллеливание алгоритмов и программ

Ведущий практической части курса – Пальян Рубен Левонович

E-mail: palian.rl@mipt.ru

Организационные вопросы. Критерии оценки.

- **Обязательные задачи (без крайнего срока)**
- 0. Вводные задачи по изучаемым технологиям
- 1. Лабораторная работа по первой части семестра
- 2. Задача по второй части семестра
- 3. Контрольные работы (1 или 2, по желанию лектора)
- Оценка усредняется и ограничивается отл(8).
- *Дополнительные задачи (имеют крайний срок)*
- Баллы суммируются с оценкой, полученной в обязательной части.
- Итоговая оценка округляется (8.5 -> 9, 8.49 -> 8).

Организационные вопросы. Доступ к учебному кластеру.

- - Доступен персональный логин каждому студенту
- - Возможен удалённый доступ по протоколу ssh с широкого круга устройств
- - Логин предназначен исключительно для выполнения задач по курсу «Параллельное программирование»
- - Возможно выполнение учебных задач на собственном оборудовании, при наличии там минимум 4 физических процессорных ядер. Ответственность за настройку и работоспособность оборудования в таком случае студент берёт на себя

Теоретическая часть - лекции

- Численные методы, алгоритмы и программы. Введение в распараллеливание (Карпов Владимир Ефимович, Лобанов Алексей Иванович), 2014 год, Физматкнига,
- ISBN 978-5-89155-234-0



Серия
Суперкомпьютерное
Образование

В. Е. Карпов
А. И. Лобанов

Численные методы,
алгоритмы и программы.
Введение в
распараллеливание



Суперкомпьютерный
консорциум
университетов России

НОЦ СКТ МФТИ



OpenMP (Open Multi-Processing)

- API (application programming interface) (интерфейс прикладного программирования), предназначенный для распараллеливания программ
- Изначально для систем с общей памятью (в последних версиях стандарта возможны иные варианты)
- Кросс-платформенный
- Основные реализации – изначально для языков C/C++ и Fortran
- Сосуществование, взаимодействие и конкуренция стандартов - де-факто сосуществование с MPI на данный момент, перспективы – имеем возможность наблюдать за развитием, пересечение возможностей с библиотеками и средствами C++, OpenCL и др.

Поддержка стандарта компиляторами

- gcc,
- LLVM (Clang),
- Microsoft Visual C++,
- Intel(C++ compiler, Fortran, Parallel Studio), AMD (AOCC), ARM,
- XL C/C++ (IBM), Oracle (ex. Sun Studio в том числе), PGI, Absoft Pro Fortran, Lahey/Fujitsu Fortran, PathScale, Cray, NAG, OpenUH Research Compiler, LLNL Rose Research Compiler, Appentra Solutions parallware compiler, Texas Instruments (C), Barcelona Supercomputing Center, HPE CCE, NVIDIA HPC Compiler, Oracle Developer Studio, Siemens.... и другие.

Основной информационный ресурс стандарта

www.openmp.org

И документация на Вашу версию компилятора, где указано, какая версия стандарта и с какими особенностями реализована.

Организация - OpenMP Architecture Review Board



AMD



ARM



cOMPunity



IBM



Intel



NASA



NEC



NVIDIA



Siemens



SUSE



Argonne National
Laboratory



ASC/Lawrence Livermore
National Laboratory



Brookhaven National
Laboratory



CSC – IT Center for
Science



Edinburgh Parallel
Computing Centre



Fujitsu



Hewlett-Packard
Enterprise



Lawrence Berkeley
National Laboratory



Leibniz Supercomputing
Centre



Los Alamos National
Laboratory



Oak Ridge National
Laboratory



Pawsey Supercomputing
Research Centre

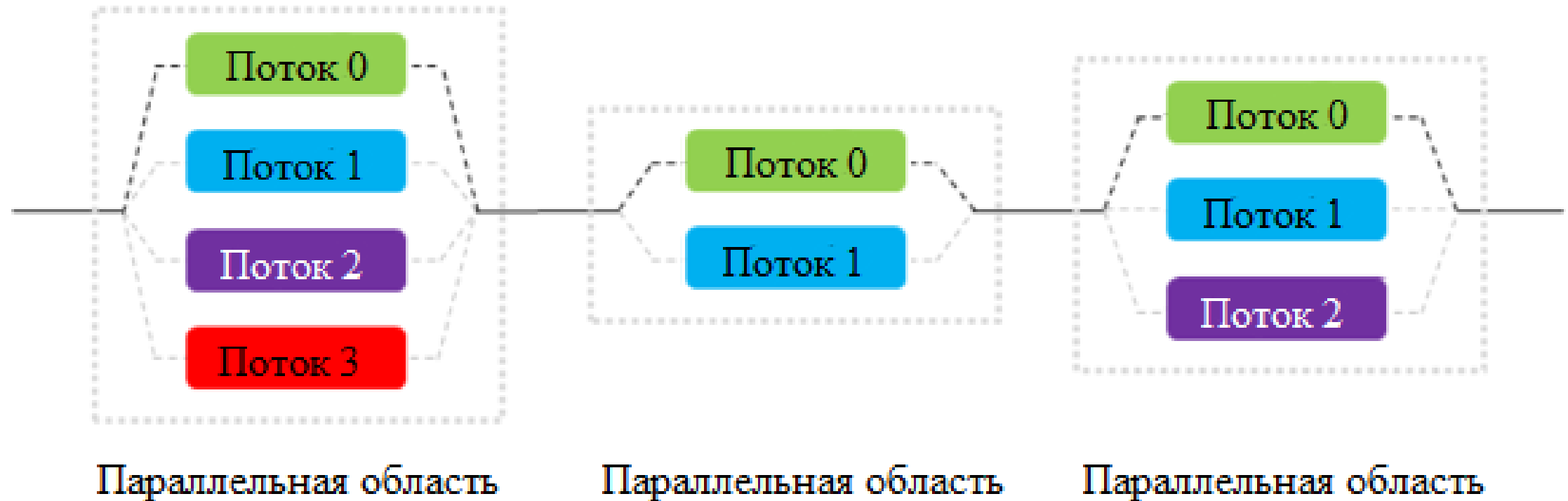
Что входит в стандарт?

- Для пользователя предоставляются:
- - директивы компилятора (`#pragma` в C)
- - библиотечные функции (процедуры)
- - переменные окружения

Изначально – для PRAM (parallel random-access machine)
(SMP - symmetric multiprocessing и т.д. – подвиды)

(дополнительное задание – как работают директивы компилятора)

«Классическая» модель OMP



Первые шаги

- `#include <omp.h>` - подключение библиотеки (для языка C)
- Опция компилятора (gcc): **gcc -fopenmp** (и остальные опции, если требуются)
- Все директивы OpenMP начинаются с **#pragma omp**
- Также имеется набор функций
- Одна из директив – **parallel**

```
#pragma omp parallel
{
    // Код внутри блока выполняется параллельно.
    printf("Hello! \n");
}
```

Директива parallel

