

[Главная](#) » [СПО](#)

СПО. Лабораторная работа №6

Разработка многопоточных приложений

[▶ Оглавление](#)

Задания

1. Написать последовательный вариант программы для расчёта интеграла из [л/р №4 С.](#)
2. Написать параллельный вариант с использованием `pthread` .

Материалы:

- Справка в вашем дистрибутиве Linux `man pthread`
- Глава 11 книги [Разработка Linux-приложений Д. Колисниченко](#)
- Хабр - [Pthreads: Потоки в русле POSIX](#)
- Руководство на сайте [RANDU](#)
- Руководство на сайте [университета Карнеги-Меллона](#)

3. Написать параллельный вариант с использованием OpenMP.

Материалы:

- Учебное пособие [А.С. Антонова “Параллельное программирование с использованием технологии OpenMP”](#)
- Руководства на сайте [OpenMP](#)
- Руководство на сайте [Ливерморской национальной лаборатории](#)

Для заданий 1-3 интеграл считать с точностью около $1e-15$ (длительность расчёта последовательного варианта от нескольких десятков секунд), значения выводить до 20 знака после запятой. Используйте тип данных `long double` для достижения требуемой точности.

4. Сравнить время выполнения программ из пп. 1-3 с помощью команды `time`.
5. Написать по своему варианту 2 программы с последовательной реализацией алгоритма и многопоточной с `pthread`, сравнить время выполнения.

Пример

В качестве примера проанализируйте простую программу, умножающую матрицу на вектор. Обратите внимание на размеры матриц, число потоков и передачу данных каждому потоку. Файлы исходного кода примеров и `makefile` для сборки [тут](#).

Варианты заданий

1. Многопоточное вычисление определителя матриц порядка N .
2. Многопоточное решение задачи о рюкзаке.
3. Многопоточный поиск простых чисел в заданном диапазоне.
4. В заданной директории лежат текстовые файлы вида:

```
Polynom: cn * x ^ pn + c2 * x ^ p2 + c1 * x ^ p1  
Interval: [a, b]  
Step: s
```

Каждый файл содержит 3 строки: запись полинома с коэффициентами c_i и степенями p_i , интервалом $[a, b]$ и шагом s для построения графика. Строки могут находиться в любом порядке. Реализовать многопоточное приложение, осуществляющее парсинг текстовых файлов, генерацию выходных файлов с таблицами для построения графика и вызов `gnuplot` по окончании построения таблицы.

5. Многопоточное вычисление значения числа π с использованием [задачи Бюффона о бросании иглы](#):

$$\pi \approx \frac{2Ln}{rh},$$

при условии $r > L$, где L - длина иглы, n - число игл, r - расстояние между прямыми, h - число игл, пересекающих прямые.

6. Многопоточный поиск в текстовых файлах. На вход подаётся путь к директории и искомый текст. Все .txt файлы в этой директории проверяются и, в случае нахождения текста в файле, выводится путь к файлу, строка и позиция в строке. Вложенные директории проверяются аналогично.
7. Многопоточное решение задачи об упаковке кругов равного размера и минимизации пустого пространства внутри квадрата с заданной стороной.
8. Многопоточное решение задачи об оптимальном пути в графе с использованием муравьиного алгоритма. В качестве графа используйте станции метро Санкт-Петербурга, Бухареста, Вены, Милана или Стокгольма со временем между станциями на рёбрах.
9. Многопоточная сортировка массива произвольной длины.
10. Многопоточный обход дерева каталогов для подсчёта размера файлов.
11. Поток-генератор порождает расчётные задачи, поток-решатель сохраняет условие задачи и результат решения в файл. Найти соотношение генераторов к решателям, при котором длина очереди заданий:
 - преимущественно нулевая,
 - остаётся близкой к некоторой фиксированной константе $c \neq 0$.

В качестве задач можно взять вычисление алгебраических выражений, интегралов, матричные расчёты и т.д.

12. Многопоточное умножение матриц большого размера.

← ПРЕДЫДУЩАЯ

СЛЕДУЮЩАЯ →

СПО. Лабораторная работа №5

СПО. Лабораторная работа №7