

ДОМАШНЕЕ ЗАДАНИЕ №4 ПО КУРСУ АВС

ВАРИАНТ 22

Пояснительная записка

Выполнила студентка группы БПИ197

ФКН ПИ НИУ ВШЭ

Тарасова Дарья

2020 год

Описание задачи

Первая задача о Винни-Пухе, или неправильные пчелы.

Неправильные пчелы, подсчитав в конце месяца убытки от наличия в лесу Винни-Пуха, решили разыскать его и наказать в назидание всем другим любителям сладкого. Для поисков медведя они поделили лес на участки, каждый из которых прочесывает одна стая неправильных пчел. В случае нахождения медведя на своем участке стая проводит показательное наказание и возвращается в улей. Если участок прочесан, а Винни-Пух на нем не обнаружен, стая также возвращается в улей. Требуется создать многопоточное приложение, моделирующее действия пчел. При решении использовать парадигму портфеля задач.

Используемая модель вычислений

Для решения задачи использовалась парадигма портфеля задач.

По сути, это способ распределения работ для реализации модели построения многопоточных приложений «взаимодействующие равные». При ее реализации управляющий поток, не производящий вычислений, исключается. Задача в данном случае представляет собой независимую единицу работы. Задачи помещаются в «портфель», разделенный между всеми процессами, работающими по алгоритму:

```
while(true){
    [получить задачу из портфеля];
    if([задач больше нет])
        break;
    fi
}
```

Выделим основные шаги, необходимые для реализации:

1. Дать представление задач
2. Определить портфель
3. Задать алгоритм выполнения задачи
4. Определить критерий окончания

Конкретно в данной задаче портфелем является лес, распределен он по задачам () между отрядами пчел (процессами). Окончание работы происходит, когда весь лес проверен пчелами.

Поиск медведя есть поиск нуля в массиве единиц. То есть, каждый поток получает свою часть вектора и проверяет его на наличие нуля.

Источники:

Демьянович Ю.К., Иванцова О.Н. Технология программирования для распределенных параллельных систем, параграф 4, страница 67: https://www.studmed.ru/view/demyanovich-yuk-ivancova-on-tehnologiya-programmirovaniya-dlya-raspredelelennyh-parallelnyh-sistem_138cec52b44.html?page=7

Блог программиста, раздел 4 Парадигма параллельного программирования взаимодействующие равные: <https://pro-prof.com/forums/topic/parallel-programming-paradigms>

Лекция 5. Парадигмы параллельного программирования (ЮФУ): http://staff.mmcs.sfedu.ru/~dubrov/files/sl_parallel_05_paradigm.pdf

Алгоритм работы программы

1. Происходит ввод и проверка на допустимость количества деревьев пользователем с помощью метода `input_length()`. Если ввод был некорректным, программа завершается с соответствующим уведомлением;
2. Метод `create_forest(int length)` создает лес, то есть заполняет вектор длины `length` единицами и одним нулём (где единицы обозначают деревья, ноль – Винни Пуха) и вычисляет количество задач для леса данной длины (считаем, что пчелам удобно проверять по 10 деревьев/отряд);
3. Метод `get_tasks()` создает вектор задач (`vector<tuple<int, int>> tasks`), в котором каждый элемент – пара индексов: начало и конец отрезка «леса», за который будет ответственен данный поток;
4. Метод `run_bag_of_tasks()` запускает распараллеленный с помощью библиотеки `openMP` цикл по задачам – каждый поток берет свою задачу (участок леса) и выполняет поиск Винни, после чего отчитывается о результатах работы;

```
[код метода run_bag_of_tasks()]
void run_bag_of_tasks()
{
#pragma omp parallel shared(forest) num_threads(number_of_tasks)
{
#pragma omp for
    for (int i = 0; i < number_of_tasks; i++)
    {
        // getting a task from the bag
        search_for_Winnie(tasks[i]);
    }
}}
```

5. Векторы чистим, задерживаем выведенное экран, программа завершается.



Руководство оператора

Для запуска программы необходимо открыть файл `Winnie_openMP.exe`. Программа попросит ввести количество деревьев. Проверка на некорректный ввод предусмотрена. После того, как будет произведен корректный ввод, можно наслаждаться результатом работы пчел.

Скриншоты, демонстрирующие работу программы на корректных и некорректных входных данных находятся в папке `screenshots`.