

Домашнее задание №5

Пояснительная записка

Описание полученного задания

Вариант: 2

Условие задачи:

Задача о Винни-Пухе - 1 или правильные пчелы. В одном лесу живут n пчел и один медведь, которые используют один горшок меда, вместимостью N глотков. Сначала горшок пустой. Пока горшок не наполнится, медведь спит. Как только горшок заполняется, медведь просыпается и съедает весь мед, после чего снова засыпает. Каждая пчела многократно собирает по одному глотку меда и кладет его в горшок. Пчела, которая приносит последнюю порцию меда, будит медведя. Создать многопоточное приложение, моделирующее поведение пчел и медведя.

Дополнительные данные:

Так как в условии не указано, какое количество горшков Винни-Пух должен съесть, программа не может быть закончена (пчелы будут приносить в горшок мёд, горшок будет заполняться, Винни-Пух будет съедать заполненный горшок мёда, а потом весь процесс повторяется, и так до бесконечности). Чтобы решить эту проблему, был добавлен дополнительный параметр ввода – количество порций, которые насыщают Винни-Пуха. Таким образом, теперь программа не будет бесконечной, а закончит свою работу, когда Винни-Пух съест указанное количество горшков.

Интерпретация задачи на языке программирования

Каждая пчела представляет собой поток. В определённый момент времени только одна пчела имеет доступ к горшку, т.е. код с логикой взаимодействия пчелы, горшка и Винни-Пуха блокируется мьютексом (`pthread_mutex_lock`). После данного взаимодействия мьютекс снимется (`pthread_mutex_unlock`), пчела улетает на определённое время `t`, т.е. делает `usleep(t)` (время может быть от 100 до 500 миллисекунд). Аналогично заполнять горшок начинает пчела, отвечающая за другой поток (пчелу определяет процессор). Заполнение горшка *i*-ой пчелой, её улёт, состояние горшка на каждом шаге, съедание Винни-Пухом мёда из горшка выводятся в консоль, а также записываются в файл. После того, как Винни-Пух съест необходимое для его насыщения количество горшков, выводятся сообщения о том, что пчёлы перестают приносить мёд (об окончании работы потоков).

Модель приложения ^{[1][2]}

Для решения поставленной задачи была выбрана модель *«портфель задач»*. Задача делится на конечное число подзадач: чтобы наполнить горшок, каждой пчеле нужно приносить какую-то часть его объёма (в написанной программе каждая пчела приносит единицу объёма горшка). Распределение работ между потоками определяется динамически: пчела (процесс) кладёт мёд в горшок, если Винни-Пух ещё не наелся, в противном случае улетает навсегда (процесс завершается). «Портфелем» в этом случае является Винни-Пух и горшок. Доступ к ним в определённый момент времени осуществляется только одним процессом (одной пчелой). Все процессы (пчёлы) выполняют одну и ту же задачу – наполняют горшок.

Требуемые метрики, определяющие характеристики программы, для различных тестовых прогонов

Программа содержит 5 интерфейсных модулей (заголовочных файлов):

- `Headers/bee.h` – 669 байт
- `Headers/honey_pot.h` – 757 байт
- `Headers/package.h` – 476 байт
- `Headers/random.h` – 828 байт
- `Headers/winnie_the_pooh.h` – 627 байт

А также 4 модуля реализации (файлов с определением программных объектов):

- `Sources/bee.cpp` – 964 байт
- `Sources/honey_pot.cpp` – 877 байт
- `Sources/main.cpp` – 6,19 КБ
- `Sources/winnie_the_pooh.cpp` – 617 байт

Общий размер исходных текстов: 11,8 КБ.

Тестирование производилось на машине с 8 ядрами. Количество ядер было найдено при помощи функции:

`std::thread::hardware_concurrency()`

Время выполнения программы для различных тестовых наборов данных (для всех тестов количество горшков, которые насыщают Винни-Пуха равно 1):

Количество пчёл	Количество глотков (вместимость горшка)	Время работы программы при вводе данных из файла	Время работы программы при вводе данных с консоли
1 пчела	5 глотков	0,0005 sec	0,0005 sec
2 пчелы	5 глотков	0,0006 sec	0,0005 sec
4 пчёлы	5 глотков	0,0007 sec	0,0007 sec
8 пчёл	5 глотков	0,001 sec	0,001 sec
20 пчёл	5 глотков	0,002 sec	0,002 sec
50 пчёл	5 глотков	0,003 sec	0,004 sec
100 пчёл	5 глотков	0,005 sec	0,008 sec
1 пчела	20 глотков	0,0009 sec	0,0009 sec
2 пчелы	20 глотков	0,0007 sec	0,0009 sec
4 пчёлы	20 глотков	0,001 sec	0,001 sec
8 пчёл	20 глотков	0,002 sec	0,001 sec
20 пчёл	20 глотков	0,002 sec	0,003 sec
50 пчёл	20 глотков	0,006 sec	0,004 sec
100 пчёл	20 глотков	0,007 sec	0,007 sec
1 пчела	50 глотков	0,002 sec	0,001 sec
2 пчелы	50 глотков	0,0015 sec	0,002 sec
4 пчёлы	50 глотков	0,0015 sec	0,002 sec
8 пчёл	50 глотков	0,002 sec	0,003 sec
20 пчёл	50 глотков	0,004 sec	0,004 sec
50 пчёл	50 глотков	0,006 sec	0,005 sec
100 пчёл	50 глотков	0,008 sec	0,008 sec
1 пчела	70 глотков	0,004 sec	0,002 sec
2 пчелы	70 глотков	0,004 sec	0,002 sec
4 пчёлы	70 глотков	0,005 sec	0,002 sec
8 пчёл	70 глотков	0,005 sec	0,003 sec
20 пчёл	70 глотков	0,005 sec	0,004 sec
50 пчёл	70 глотков	0,003 sec	0,006 sec
100 пчёл	70 глотков	0,009 sec	0,008 sec
1 пчела	100 глотков	0,003 sec	0,003 sec
2 пчелы	100 глотков	0,003 sec	0,002 sec
4 пчёлы	100 глотков	0,004 sec	0,003 sec
8 пчёл	100 глотков	0,004 sec	0,004 sec
20 пчёл	100 глотков	0,005 sec	0,006 sec
50 пчёл	100 глотков	0,007 sec	0,006 sec
100 пчёл	100 глотков	0,009 sec	0,009 sec

Формат аргументов командной строки и тестов

В программе есть две возможности задать исходные значения: с помощью входного файла, с помощью аргументов командной строки.

Для ввода данных из файла необходимо ввести следующую команду:

```
-f test/test.txt output/output.txt
```

Данные в файле должны быть написаны в следующем формате:

```
<bees_number> <honey_pot_capacity> <portions_number>
```

Для ввода из командной строки нужно ввести следующую строку:

```
-n <bees_number> <honey_pot_capacity> <portions_number> output/output.txt
```

В данных обозначениях:

- `bees_number` – количество пчёл (в условии задачи обозначается как **n**)
- `honey_pot_capacity` – вместимость горшка (в условии задачи обозначается как **H**)
- `portions_number` – количество горшков, которые нужно съесть Винни-Пуху, чтобы насытиться

Все три параметра могут принимать целые числа от 1 до 100.

Дополнительные задачи

Из дополнительных задач было реализовано: модульная структура, поясняющие комментарии, обработка некорректных чисел во входных данных.

Список используемой литературы

[1] Карепова, Е. Д. ОСНОВЫ МНОГОПОТОЧНОГО И ПАРАЛЛЕЛЬНОГО ПРОГРАММИРОВАНИЯ / Е. Д. Карепова. – Красноярск : Библиотечно-издательский комплекс Сибирского федерального университета, 2016. – 356 с.

[2] Издательский дом "Вильямс" [Электронный ресурс]. – Режим доступа: <http://www.williamspublishing.com/PDF/5-8459-0388-2/part.pdf>. – Дата доступа: 15.12.2021.