

Отчёт по программе, разработанной на C++ с поддержкой многопоточности.

Основная информация

Студент

Куликов Богдан, БПИ-204

Описание задания

17. N пчел живет в улье, каждая пчела может собирать мед и сторожить улей ($N > 3$). Ни одна пчела не покинет улей, если кроме нее в нем нет других пчел. Каждая пчела приносит за раз одну порцию меда. Всего в улей может войти тридцать порций меда. Вини-Пух спит пока меда в улье меньше половины, но как только его становится достаточно, он просыпается и пытается достать весь мед из улья. Если в улье находится менее чем три пчелы, Вини-Пух забирает мед, убегает, съедает мед и снова засыпает. Если в улье пчел больше, они кусают Вини-Пуха, он убегает, лечит укусы, и снова бежит за медом. Создать многопоточное приложение, моделирующее поведение пчел и медведя.

Структура проекта

```
project/
|
+-actors/           # классы, выполняющиеся в различных потоках.
+-objects/          # классы, являющиеся разделяемыми объектами.
+-docs/             # документация.
+-utils/            # вспомогательные классы.
+-stop_program.sh   # шелл-скрипт для корректной остановки программы.
+-main.cpp          # точка входа в приложение.
```

Спецификация

Спецификация ВС

- **Operating System:** Arch Linux
- **Kernel:** Linux 5.14.7-arch1-1
- **Architecture:** x86-64
- **RAM:** 16Gb

Спецификация средств разработки

- **IDE:** CLion(v2021.2.2)
- **Библиотеки:**
 - iostream
 - thread
 - vector
 - algorithm
 - random
 - cstdio
 - signal

- **Средство компиляции:** CMake(v3.20)

Флаги

Программа поддерживает 2 флага.

- `--help(-h)` - вывод информации о программе и подсказок.
- `--number-of-bees(-n)` - ввод количества пчел. Тип: положительное ненулевое целое число.

Окончание работы

Т.к. в ТЗ нет условия окончания работы, то программа будет работать бесконечно. Для того, чтобы завершить работу не аварийно, необходимо послать в процесс сигнал SIGINT. В консоли Unix систем для этого подойдет клавиш `Ctrl + C`. Однако, при работе в CLion локальная консоль CLion'a не позволяет использовать сочетание клавиш `Ctrl + C` для отправки сигнала в процесс. Поэтому был реализован скрипт `stop_program.sh`, при запуске которого будет послан сигнал в процесс, "название" которого совпадает с названием бинарного файла в папке `bin/`. Для того, чтобы этот скрипт работал, необходимо собирать и запускать бинарный файл из папки `bin/`.

Характеристики проекта

- Количество заголовочный файлов: 8
- Количество программных объектов: 5
- Размер исходных файлов: ~ 20 Kb
- Размер бинарного файла: ~ 22 Kb

Описание работы программы

Для реализации поведения, описанного в ТЗ, больше всего подходит архитектура "Взаимодействующие равные". Т.к. пчелы оперируют независимо друг от друга, аналогично и Винни-Пух выполняет свои задачи независимо от пчел. Разделяемым ресурсом в таком случае является улей, который хранит информацию о количестве меда и количестве пчел в улье. Критическими секциями являются:

- Количество меда в улье
- Количество пчел в улье

Для доступа к этим разделяемым переменным используются геттеры и сеттеры внутри класса `Beehive`, которые берут мьютекс, во время получения/изменения переменных. Такое решение позволяет пчелам и Винни-Пуху работать независимо и не создавать гонок данных. Каждый мьютекс покрывает свою переменную, что позволяет не блокироваться полностью, когда прилетела пчела - другие акторы еще имеют доступ к количеству меда. Для того, чтобы управлять потоками используется класс `context`. Он позволяет каждую итерацию проверять был ли закрыт контекст и если был закрыт, то реагировать на это. Аналогом контекста является семафор. Однако контекст также является потокобезопасным из-за мьютекса.