

Лаб. 3: Синхронизация потоков

Задание

1. Ознакомиться с теоретическим материалом для данной работы.
2. Изучить предложенные демонстрационные примеры многопоточных приложений, использующие механизм взаимного исключения: задача об обедающих философах (phil.cpp) и задача читателей-писателей (rw.cpp).
3. **Быть готовым ответить на вопросы по "Войне потоков" (WarThreads.cpp).**
Примерный (но далеко не полный!) список тем:

- Общая архитектура решения;
- Реализация отложенного старта игры;
- Механизм генерации пуль;
- Механизм генерации противников;
- Механизм разделения доступа к общему ресурсу - буферу консоли;
- ...
- Реализация завершения игры.

Замечания

При желании, "Война потоков" может быть переписана на любом предпочитаемом ЯП с нативной поддержкой потоков, примитивов синхронизации и прочих необходимых инструментов.

В этом случае вопросы будут задаваться с учетом реализации; кроме того, при хорошем уровне работы возможны бонусы в виде дополнительных баллов.

Запуск "Войны потоков"

Внутри каталога .build в соответствующих подкаталогах лежат проекты для MSVS 2010, MSVS 2012 и MSVS 2015. Если по каким-то причинам эти проекты не устраивают/не работают/не подходят, возможны следующие варианты:

- Продвинутый вариант: скачать утилиту premake5, поправить premake5.lua и выполнить команду premake5 <название_цели>.
- Более простой вариант: создать пустой проект WinAPI в вашей IDE (MSVS, Code::Blocks, ...) и импортировать туда файл WarThreads.cpp.

Само приложение, разумеется, реализовано на чистом WinAPI без использования возможностей C++, поэтому практически без изменений может быть скомпилировано Сишным компилятором (достаточно сменить расширение файла).

Критерии оценивания

- Вопросы по семафорам - 1 балл
- Вопросы по мьютексам - 1 балл
- Вопросы по критическим секциям - 1 балл
- Вопросы по атомарным операциям - 1 балл
- Общие вопросы - 1 балл

