

## Введение

Программные модели, которые эмулируют какую-либо систему, достаточно плотно используются на лабораторных работах по дисциплине «Операционные системы». Ни одна лабораторная не проходит без лабораторной установки.

К сожалению, качество некоторых приложений оставляет желать лучшего, что затрудняет изучение нового материала.

## Функции планировщика процессов

Например, на первой лабораторной работе изучается планировщик (или диспетчер) процессов ОС, в чьи функции входит:

- Планирование выполнения процессов и потоков (задач)
- Создание и уничтожение процессов
- Обеспечение процессов необходимыми системными ресурсами
- Поддержание взаимодействия между процессами

Планирование выполнения процессов включает в себя решение следующих задач:

- Определение момента времени для смены текущего активного потока
- Выбор потока для выполнения из очереди готовых потоков

В данной программной модели исследуются не все функции планировщика.

## Модель подсистемы управления процессами

В текущей лаб. установке реализована следующая упрощенная модель диспетчера процессов:

- В системе имеется 16 очередей процессов
- Процесс определяется следующими параметрами:
  - Идентификатор процесса (PID)
  - PID родителя
  - Базовый и текущий приоритеты
  - Время выполнения
  - Заявленное время выполнения
  - Состояние (ожидание/готовность/исполнение)
- Приоритет процесса может быть от 0 (самый низкий) до 15 (самый высокий)

## Текущая программная модель

В данной программной модели пользователю предлагается обработать поступающие в диспетчер заявки. Заявки бывают следующих видов:

- Создание нового (дочернего) процесса
- Завершение процесса
- Запрос на ввод/вывод
- Завершение ввода/вывода
- Передача управления операционной системе
- Истечение кванта времени

Пользователь может:

- Создать/удалить процесс
- Перевести процесс в состояние ожидания/готовности/исполнения
- Добавлять/удалять процессы из очередей
- Проигнорировать заявку, ничего не изменяя

## Недостатки

В текущей программной модели были найдены следующие недостатки:

- приложение доступно только для ОС Windows. Пользователи других операционных систем должны страдать;
- нестабильность. В ходе выполнения лабораторной работы программная модель несколько раз аварийно завершалась, из-за чего результаты выполненной работы безвозвратно терялись;

- ошибки при проверке пользовательских действий. В некоторых случаях было замечено, что программная модель принимала правильную последовательность действий как ошибочную. Это в совокупности с нестабильностью программы усложняет изучение студентами программной модели и, как следствие, увеличивает время на выполнение лабораторной работы;
- нет генератора заданий;
- нечеткость, «размытость» интерфейса на дисплеях со сверхвысоким разрешением (HiDPI) – ну это уже косметические придирки.

## Актуальность

Сначала я попытался исправить часть проблем путем дизассемблирования с помощью IDA Freeware, но это оказалось крайне сложно. Т. к. без этой установки лабораторную не сделать, было принято решение о разработке новой программной модели, повторяющей функционал текущей, в которой будут исправлены вышеописанные ошибки и недостатки.

## Требования к программе

Были составлены следующие требования к программе:

- Возможность генерации заданий
- Возможность загрузки и сохранения заданий в файл с сохранением прогресса выполнения
- Возможность подсчета количества ошибок, сделанных в ходе выполнения задания
- Обеспечить просмотр и отмену действий, выполненных в ходе выполнения задания
- Доступность под различные ОС (Windows и Ubuntu)

## Модульная структура

Я решил разбить приложение на 4 модуля:

- Модуль обработки заявок. Отвечает за обработку заявок и проверку выполненных пользователем действий
- Модуль-генератор заданий. Отвечает за генерацию заданий с достаточной степенью уникальности для каждого пользователя без необходимости разработки заданий вручную
- Модуль загрузки и сохранения заданий в файл. Отвечает за чтение/сохранение заданий в файл и защиту от непредвиденного изменения файлов
- Модуль графического интерфейса. Является связующим звеном между приложением и пользователем; отображает данные о ходе выполнения задания в текстовом и графическом виде

## Алгоритмы функционирования

Разработаны были следующие алгоритмы:

- Алгоритмы обработки заявок:
  - о Создание нового процесса
  - о Завершение процесса
  - о Запрос на ввод/вывод
  - о Завершение ввода/вывода
  - о Передача управления ОС
  - о Истечение кванта времени
- Алгоритм генерации заданий

## Диаграмма классов

В ходе проектирования была составлена вот такая диаграмма классов.

Разработаны отдельные классы для:

- Дисциплин планировщика
- Заявок
- Задания
- Состояния системы (процессы и очереди)
- Справочного, основного окна и виджета задания

## Инструменты разработки

Приложение должно запускаться на разных ОС и быть простым в запуске. Для этого лучше всего подходит связка Qt и C++. C++ - компилируемый язык, позволяет получить родные для каждой ОС исполняемые файлы. Qt – кроссплатформенный фреймворк, используется для построения GUI, предоставляет удобную IDE QtCreator со встроенным редактором форм.

В качестве системы сборки была выбрана CMake

## Разработка пользовательского интерфейса

Проанализировав интерфейс предыдущей установки, было принято решение внести в него несколько небольших изменений:

- Добавить кнопку «Сбросить». При нажатии на нее все изменения, сделанные в ходе обработки текущей заявки, будут сброшены;
- Вместо пиктограмм на кнопках будут содержаться соответствующие их смыслу надписи;
- Для кнопок «Подтвердить» и «Сбросить» добавлены клавиатурные сочетания «Alt+Enter» и «Ctrl+Z» соответственно;
- Для состояний процессов сделаны более заметные пиктограммы в виде кружков разного цвета;
- При попытке закрыть основное окно программы будет выводиться диалог подтверждения, чтобы исключить потерю результатов из-за случайного закрытия программы;
- На случай поиска и устранения непредвиденных неполадок сделано автоматическое сохранение сгенерированного задания в файл во временной папке пользователя. Для доступа к ней в меню приложения добавлен отдельный пункт

## Дальнейшее развитие

В ходе выполнения курсового проекта было разработано программное обеспечение – лабораторная установка «Модель диспетчера процессов операционной системы». Изначально предполагалось исправление и доработка существующей установки, однако такая задача оказалось крайне сложной в сравнении с разработкой нового приложения.

В качестве направления дальнейшего развития можно выбрать разработку отдельного конструктора заданий для преподавателя и обеспечение шифрования файлов заданий. Также планируется портирование приложения на macOS.