

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 (особенно этой программы), что затрудняет изучение нового материала.

Функции планировщика памяти

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

- **отслеживание** свободной и занятой памяти
- первоначальное и динамическое **выделение** памяти процессам приложений и самой ОС и **освобождение** памяти по завершении процессов
- **настройка адресов программы** на конкретную область физической памяти
- полное или частичное **вытеснение кодов** и данных процессов из ОП на диск, в случае нехватки места, и возвращение их в оперативную память
- **защита** памяти, выделенной процессу, от возможных вмешательств со стороны других процессов
- **дефрагментация** памяти

В данной программной модели исследуются **не все функции** планировщика.

Модель памяти

Представлена достаточно **упрощенная модель** памяти:

- Адресное пространство разбито на 256 страниц памяти по 4096 байт каждая
- Непрерывная область памяти, состоящая из одной и более страниц – блок памяти
- Параметры блока:
 - адрес начала блока
 - размер блока в страницах
 - идентификатор процесса, которому принадлежит данный блок

Память процессам при каждом запросе выделяется в виде одного блока памяти

Текущая программная модель

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

- Создание нового процесса/завершение процесса
- Выделение памяти существующему процессу/освобождение памяти

Пользователь может:

- Ответить отказом
- Выделить/освободить память
- Объединить соседние блоки/дефрагментировать память

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

Нужно еще сказать о том, что программа должна запускаться на Windows и на 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.

Диаграмма классов

В ходе проектирования была составлена вот такая диаграмма классов.

Алгоритмы функционирования

Разработаны были следующие алгоритмы:

- Алгоритмы обработки заявок (4 штуки)
- Алгоритм генерации заданий

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.

Разработка пользовательского интерфейса

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

- кнопку «Отклонить» заменить на кнопку «Сбросить»;
- вместо пиктограмм на кнопках будут содержаться соответствующие их смыслу надписи;
- для удобства пользования для кнопок «Подтвердить» и «Отклонить» добавлены клавиатурные сочетания «Alt+Enter» и «Ctrl+Z» соответственно;
- для свободных и занятых блоков памяти сделаны более заметные пиктограммы в виде кружков разного цвета;
- при попытке закрыть основное окно программы будет выводиться диалог подтверждения;
- на случай поиска и устранения непредвиденных неполадок сделано автоматическое сохранение задания в файл во временной папке пользователя.

Дальнейшее развитие

В ходе выполнения курсового проекта было разработано программное обеспечение – лабораторная установка «Модель диспетчера памяти 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 разработка лабораторной установки для планировщика процессов операционной системы.