

МБОУ Академический лицей им. Г.А. Псахье г. Томска

Проектная работа

Архитектуры операционных систем. Создание оболочки операционной системы на языке программирования Python

Работу выполнили:

Нетаев Владислав, Логачёв Егор, Домнич Владимир, ученики 10 класса

Научный руководитель:

Калашникова Светлана Аверьяновна,

учитель информатики и ИКТ

Томск – 2019

Введение

В наше время нас окружает большое количество устройств, каждое из которых обладает собственной операционной системой. Это как уже давно привычные смартфоны и компьютеры, так и развивающийся рынок умной бытовой техники – телевизоры, чайники, холодильники и т.д. Многие, в том числе и мы, задумывались об устройстве всех этих устройств, в том числе и в плане программного обеспечения, в данном случае – операционных систем. Мы же решили пойти еще дальше и создать собственное подобие операционной системы на основе знаний об ее устройстве.

Таким образом, **цель** нашей проектной работы – создание оболочки операционной системы на языке программирования Python.

Мы определили следующий перечень **задач**:

1. Изучить общие принципы архитектуры операционных систем;
2. Спроектировать дизайн и функциональность планируемой оболочки операционной системы;
3. Изучить дополнительные необходимые модули языка программирования Python.

Практическая значимость заключается в применении полученных в школе в рамках курса информатики и ИКТ знаний о языке программирования Python и модулями tkinter, livewires.

В соответствии с имеющимися знаниями и навыками мы **распределили работу** над проектом между собой следующим образом:

- Домнич Владимир подготовил информацию об устройстве операционных систем, проектировал дизайн операционной системы;

- Нетаев Владислав разрабатывал саму оболочку и программы «Настройки» и «Калькулятор», так как подробно изучал модуль tkinter, предназначенный для создания графического интерфейса;
- Логачёв Егор разрабатывал программу «Блокнот» и игру «Pong», так как подробно изучал модуль livewires, предназначенный для создания игр.

Работа над проектом происходила по следующему *плану*:

1. Изучение тематической литературы, изучение архитектуры операционных систем – до 1 января 2019 года;
2. Создание общего дизайн-кода оболочки – параллельно с первым этапом, до 1 января 2019 года;
3. Написание основного кода оболочки – до середины февраля 2019 года:
 - 1) Первые экраны оболочки – до 1 января 2019 года;
 - 2) Экран блокировки, распознавание пароля – до 15 января 2019 года;
 - 3) Реестр – до 1 февраля 2019 года;
 - 4) Рабочий стол – до 15 февраля 2019 года;
4. Написание кода программ – до 15 марта 2019 года.

Принципы архитектуры операционных систем

Какое существует разнообразие операционных систем, такое существует и разнообразие архитектур операционных систем. Тем не менее, во всех них присутствуют схожие элементы, главное из которых – *ядро*.

Ядро служит переводчиком с программного языка на язык машинных кодов, загружается в память один раз и остается там постоянно. Среди функций ядра управление процессами, памятью, устройствами ввода-вывода, архивация информации и поиск необходимых фактов.

Важными элементами операционной системы также являются *подсистема управления ресурсами* (управляет вычислительными ресурсами компьютера),

управляющая программа (контролирует исполнение других программ и функционирование устройств ввода-вывода) и *пользовательский интерфейс*, являющийся наиболее знакомой для пользователя частью операционной системы. Интерфейс – графическая оболочка, красивая обёртка для скучного и непонятного для пользователя интерфейса. Поскольку именно с интерфейсом «общается» пользователь, а также из-за отсутствия необходимости низкоуровневых операций мы решили воссоздать именно оболочку операционной системы.

Использованные модули

Для создания оболочки нам потребовались некоторые модули из стандартной библиотеки Python и еще несколько модулей, не входящих в нее. Основной модуль – *tkinter*, позволяющий создать графический интерфейс. Поскольку важную роль играют изображения, нам пришлось использовать модуль *PIL* не из стандартной библиотеки.

В некоторых программах и местах операционной системы мы использовали другие модули. Например, модуль *math* использовался в калькуляторе, позволив осуществить вычисление тригонометрических и других функций. Также мы использовали модуль *datetime* для получения информации о времени и модуль *platform* для получения информации об используемой операционной системе и процессоре.

Для создания игры «Pong» использовался модуль *livewires*, реализующий в более понятной форме функции модуля *pygame*.

Полученный результат

В результате нашей работы нам удалось получить работающую оболочку операционной системы, внешний вид которой представлен ниже.

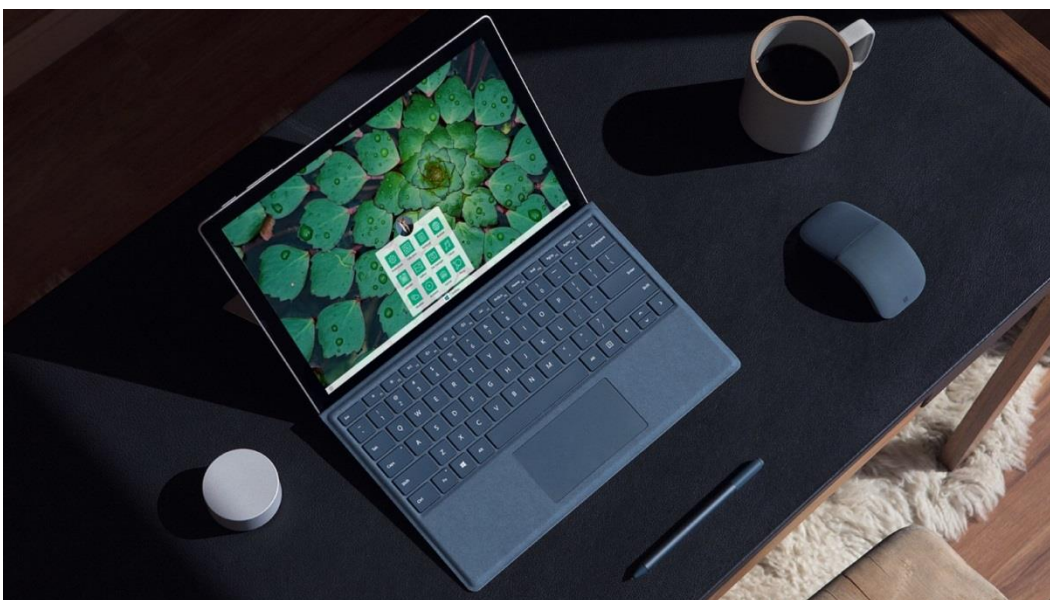


Рисунок 1. Интерфейс оболочки

В оболочке реализован *экран блокировки*, на котором пользователь должен ввести пароль. Пользователь сможет продолжить работу только при введении верного пароля.

На рабочем столе расположена *панель задач*, в которой есть два *меню* (одно позволяет выйти из оболочки или вызвать экран блокировки; второе является меню для запуска приложений), отображается текущая открытая программа и текущее время.

Из приложений реализованы следующие:

- *«Калькулятор»* позволяет выполнить 14 математических операций (три вида деления, прочие арифметические операции, тригонометрические функции, вычисление корня, числа в заданной степени и показательная функция).

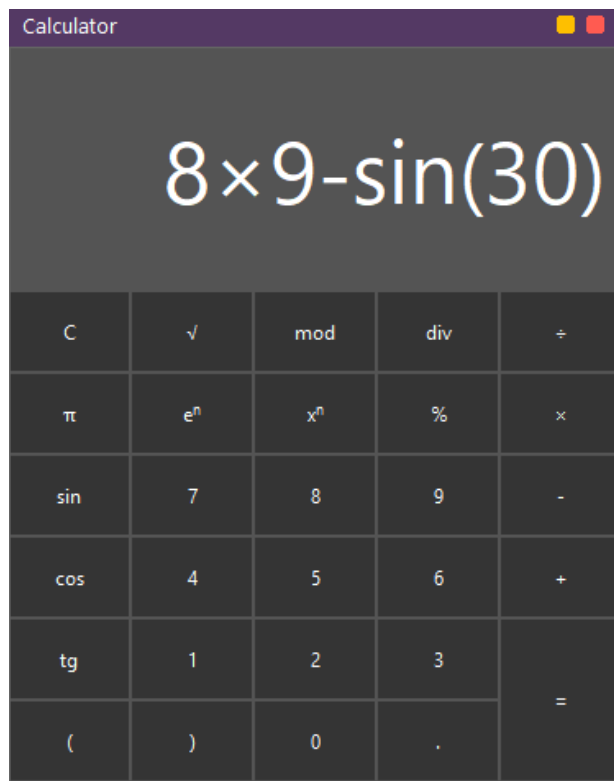


Рисунок 2. Интерфейс калькулятора

- **«Настройки»** позволяют выбрать один из предложенных фонов рабочего стола и акцентный цвет, а также сменить имя пользователя и пароль. Все изменения сохраняются в собственном реестре, поэтому после перезапуска оболочки все изменения останутся на месте.

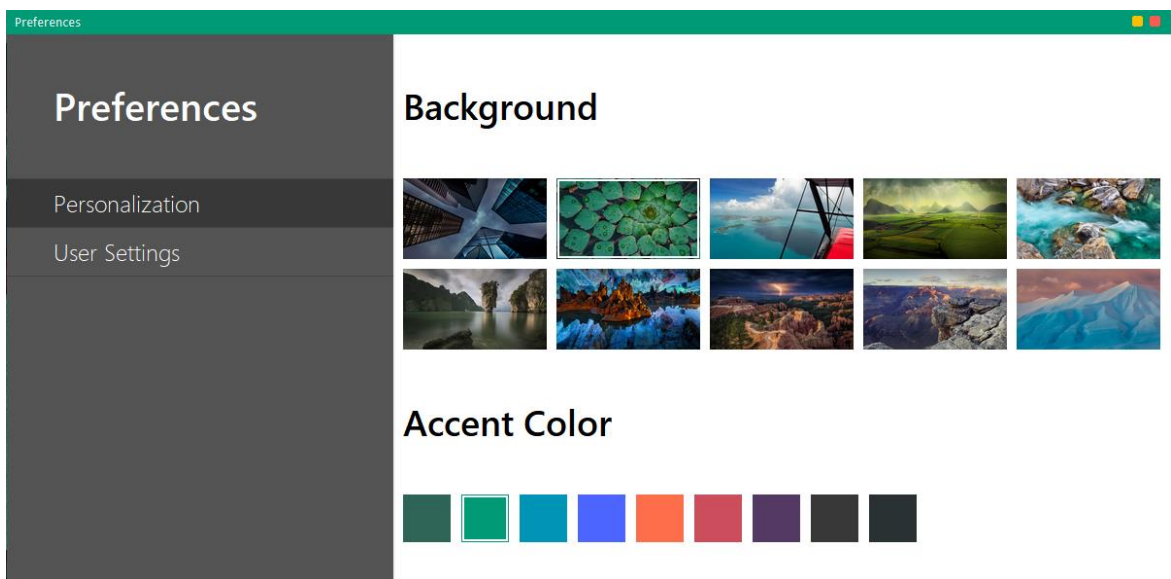


Рисунок 3. Настройки персонализации



Чопа Логачёв

Administator

Change Username

Apply

Change Password

Apply

Рисунок 4. Настройки пользователя

- **«Блокнот»** позволяет выполнить набор текста, сохранить его и в дальнейшем загрузить. Все файлы также сохраняются вне зависимости от перезагрузки оболочки.

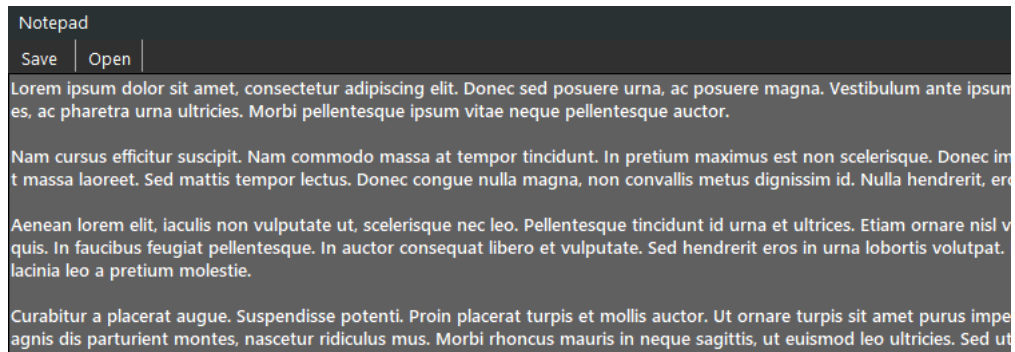


Рисунок 5. Интерфейс текстового поля блокнота

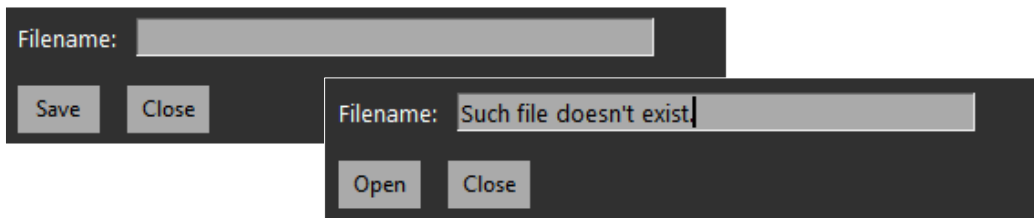


Рисунок 6. Диалоговые окна при сохранении и открытии файла

- Игра **«Pong»** – реализация настольного тенниса. В игре присутствует мультиплеер и сохранение результатов.

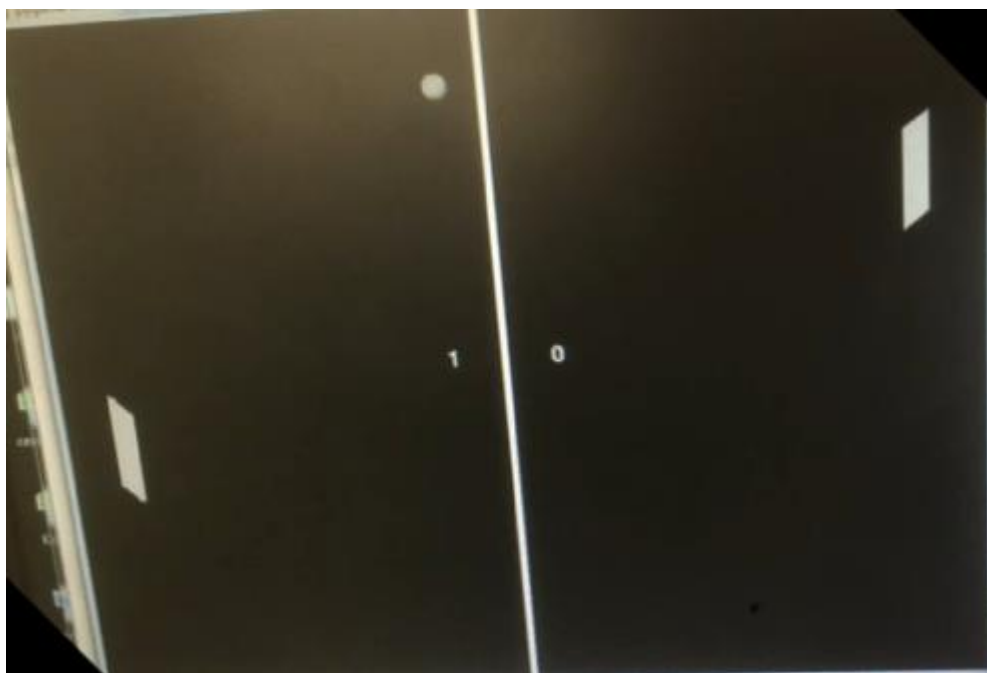


Рисунок 7. Скриншот из видео с демонстрацией игры

Значимость работы

Значимость работы заключается в том, чтобы она могла быть продолжена в том или ином виде. В нашем случае, мы начали разрабатывать учебное пособие на основе приобретенных знаний и опыта по созданию собственной оболочки на языке программирования Python для начинающих. В результате каждый сможет воссоздать свой аналог оболочки операционной системы и узнать о вещах, применяющихся в нашем проекте. Например, об объектно-ориентированном программировании, работе с файлами, модулях tkinter и livewires, нюансах работы с ними. Также мы расскажем о своем опыте проектной работы, программировании на Python, о часто встречающихся ошибках и путях их исправления.

Список литературы

1. An Introduction to Tkinter [Электронный ресурс]. – режим доступа: <http://effbot.org/tkinterbook> – (дата обращения: 17.12.2018).
2. Архитектура операционных систем [Электронный ресурс]. – режим доступа: <https://en.ppt-online.org/22333> – (дата обращения: 20.12.2018).