

Что такое Операционная Система?

- Современный «компьютер» состоит из:
 - Одного или нескольких процессоров
 - Основной памяти
 - Дополнительной памяти
 - Периферийного оборудования
 - Ввод, вывод
 - Хранение
- Для управления всем этим нужно особое ПО –
операционная система

Что такое ОС?

Прикладное ПО (браузер, мессенджер, и т.п.)
Интерфейс пользователя
Операционная система
«Железо»

- ОС – это всё то, что не нужно писать программисту при разработке прикладного ПО

Что такое ОС?

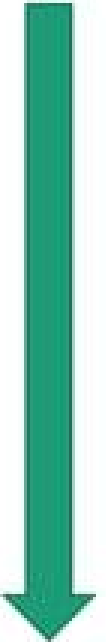
- ОС – это библиотека, и
 - с одной стороны это так:
 - все операции ввода/вывода требуют вызовов ОС (системные вызовы)
 - с другой стороны это не так:
 - Процессор и память используются без системных вызовов
 - ОС может сама вмешиваться в процесс выполнения
- ОС – это расширенная машина
 - Абстрагирует плохой интерфейс железа
- ОС – это менеджер ресурсов
 - Позволяет программам работать одновременно
 - Управляет памятью, вводом/выводом, и пр.

История операционных систем

- Почему ОС существуют в том виде, в котором они есть сейчас
- Определить функции современных ОС
- Развитие смартфонов повторяет историю развития ПК

Эволюция ОС

- Последовательное выполнение заданий
- Простая пакетная обработка (batch systems)
- Мультипрограммирование в пакетных системах
- Разделение времени (time sharing)
- Современные операционные системы



Последовательное выполнение

- Самый ранний этап
- Операционная система отсутствует
- Программист работает напрямую с аппаратурой
- Проблемы
 - Много времени тратится на то, чтобы подготовить программу к запуску
 - Прямой доступ ко всему оборудованию
 - Сложно программировать
 - Отсутствие понятий автоматизированного планирования заданий
 - Пользователи резервировали своё вычислительное время в специальном расписании

Простая пакетная обработка

- Пакетная обработка (batch processing)
 - Уменьшение времени подготовки программ за счёт группировки подобных заданий
- Это и были первые мейнфреймы
- Автоматическое последовательное выполнение заданий, автоматическая передача управления от одной задачи к другой. Возник язык управления заданиями Job Control Language (JCL)
- Резидентный монитор
 - Предшественник операционной системы
 - Управление процессором переключалось между монитором и пользовательской программой

Простая пакетная обработка

- Резидентный монитор
 - Постоянно находится в памяти
 - Управляет последовательностью событий
 - Включает интерпретатор языка JCL
- Работа монитора
 - Загрузка заданий
 - Пользовательская программа
 - Дополнительная программа (напр. компилятор)
 - Данные для обработки
 - Загрузка дополнительных нерезидентных частей монитора и общих функций, необходимых для программ (т.н. «загрузка по требованию»)

Простая пакетная обработка

- Монитор
 - Управляет последовательностью действий
 - Цикл выполнения
 - Монитор загружает задачу и передаёт управление загруженной программе
 - Программа выполняется
 - Когда программа завершила выполнение, управление возвращается монитору
- Процессор
 - Вначале выполняет инструкции из памяти, в которой находится монитор
 - Во время выполнения может быть загружена задача и процессор будет выполнять пользовательскую программу
 - Передача управления заданию: процессор выполняет инструкции из области памяти, где находится пользовательская программа
 - Передача управления обратно монитору: процессор выполняет инструкции из области памяти, где находится монитор

Простая пакетная обработка

- Общие наблюдения
 - Пользовательская программа может содержать ошибки
 - Опасно для всей вычислительной системы
 - Может перезаписать ту область памяти, в которой находится монитор (или операционная система)
 - Задача может не передать управление обратно в монитор (зациклилась)
 - «Не изобретать велосипед»
 - Многие пользовательские программы выполняет похожие действия
 - Нужна библиотека подпрограмм, которые реализуют функции, требуемые всеми программами, напр. операции ввода/вывода
- Эти проблемы и определяли дальнейшее развитие и архитектуру операционных систем

Поддержка в аппаратуре

- Защита памяти для ОС
 - Разделение памяти на область ОС и область пользователя
 - Процессор обнаруживает недопустимое обращение к памяти и прерывает выполнение задания
- Привилегированные инструкции
 - Могут выполняться только операционной системой
 - Процессор обнаруживает недопустимую инструкцию и прерывает выполнение задания
- Таймер
 - Устанавливать временные ограничения на задачи
- Прерывания
 - Дают ОС больше гибкости в управлении пользовательскими программами

Режимы работы в современных ОС

- *режим пользователя (user mode)*
 - некоторые области памяти защищены
 - некоторые инструкции запрещены
- *режим ядра (kernel mode)*
 - функции операционной системы
 - разрешён доступ к защищённым областям памяти
 - разрешено исполнение привилегированных инструкций

Системы разделения времени

- Интерактивность
 - Несколько пользователей могут одновременно работать с системой через терминалы
 - Пользователи взаимодействуют через терминальную сессию или оболочку, которая понимает команды, необходимые для запуска программ
- ОС распределяет процессорное время между пользовательскими программами
- Каждой программе пользователя выделяется короткий период или *квант* процессорного времени
 - Когда с системой работают N пользователей, каждому пользователю будет выделено $1/N$ эффективной вычислительной мощности (а также ещё есть накладные расходы на саму ОС)
 - Т.к. человек «медленнее» в сравнении со скоростью процессора, то время ответа такого разделённого компьютера может быть близко ко времени ответа выделенного компьютера
- Цели
 - Быстрое время ответа
 - Пользователю нужен ответ от компьютера как можно быстрее
 - Разделение времени создаёт иллюзию, что ему доступен весь компьютер целиком
 - Максимизация использования процессора: больше программ – больше время ответа

Проектирование ОС

- Цели с точки зрения пользователя
 - ОС должна быть удобна в использовании, легка в освоении, надёжна, быстра и безопасна
- Системные цели
 - ОС должна быть легко проектируемой, реализуемой и поддерживаемой
 - ОС должна быть гибкой, надёжной, эффективной и не содержать ошибок
- Нет универсального решения
 - Большое разнообразие ОС

Менеджер ресурсов

- Операционная система – это менеджер ресурсов
 - Управляет использованием аппаратных ресурсов
- Аппаратное обеспечение поддерживает
 - Выполнение основных инструкций
 - Обработка исключений
 - Базовые механизмы адресации памяти
 - Поддержка режимов пользователя и ядра для защиты ресурсов
- Операционная система управляет программной реализацией этих аппаратных сервисов
 - Управление процессами
 - Управление виртуальной памятью
 - Хранение файлов и управление коммуникациями

Системная инфраструктура



Сервисы операционной системы

- Планирование ЦП
 - Распределяет время ЦП между несколькими процессами, которые должны выполняться в одно время
- Управление памятью
 - Разделяет физическую память между несколькими процессами
- Своппинг
 - Перемещает процессы и их данные между основной памятью и диском для создания иллюзии большей памяти
- Ввод/вывод
 - Предоставляет специализированный код для оптимального соответствия требованиям устройств ввода/вывода
- Файловая система
 - Организует хранение данных в виде файлов и директорий

Сервисы операционной системы

- Утилиты
 - Учёт, установка/ограничение доступа к системным ресурсам, работа с файловой системой
- Командный интерфейс
 - Текстовый или графический, для возможности интерактивного взаимодействия и управления операционной системой
- Системные вызовы
 - Контролируемое обращение к внутреннему слою ОС
- Защита
 - Позволяет самим процессам и их данным не мешать друг другу и самой ОС, но предоставляет совместный доступ по требованию
- Взаимодействие
 - Позволяет пользователям и их программам взаимодействовать в рамках одной машины либо через сети

Основные достижения

- Основные достижения в разработке операционных систем
 - Многозадачность
 - Выполнение нескольких задач на одном компьютере
 - Концепция менеджеров в ОС
 - Менеджер процессов
 - Менеджер памяти
 - Менеджер ресурсов и планировщик
 - Защита информации и безопасность
 - Структурированные подходы к проектированию ОС
 - Архитектуры ядер

Структуры ПО и «железа»



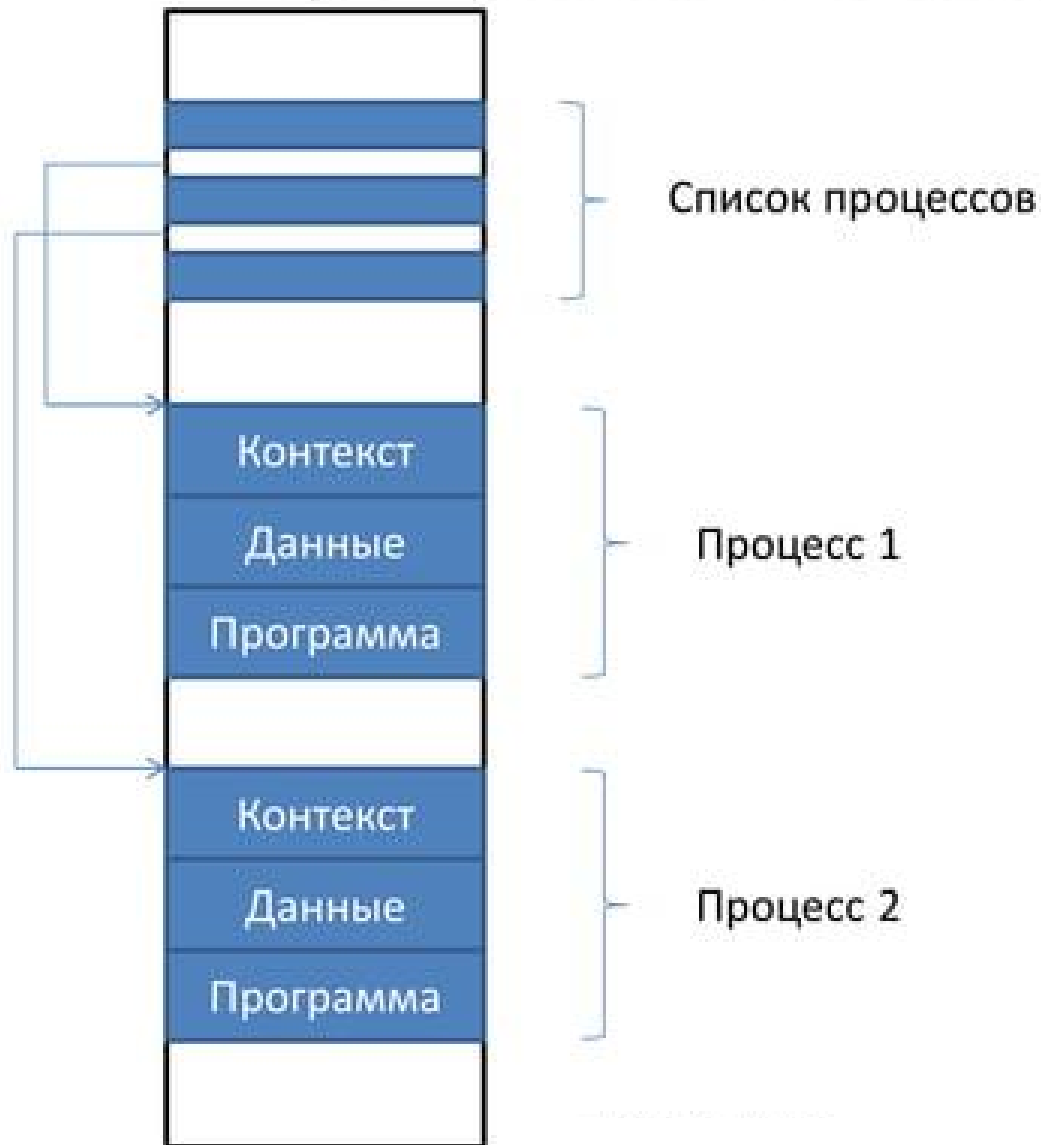
Процессы

- Процессы – базовая исполняемая единица в операционной системе
- Это выполняющаяся программа
 - Один (или больше) потоков
 - Текущее состояние
 - Множество используемых процессом системных ресурсов

Управление процессами

- Процессы управляются операционной системой
- Каждому процессу присваивается своё вычислительное время и память
- Планирование и управление памятью
 - Справедливое распределение
 - Защита и разделение процессов в памяти

Процессы в памяти



Многозадачность

- Одновременное выполнение процессов называется многозадачностью
- Многозадачность создаёт иллюзию «параллельного» вычисления
- Переключение между процессами
 - ОС запоминает значение регистра Program Counter того процесса, который приостанавливается
 - Когда выполнение процесса возобновляется, сохраненное значение Program Counter загружается в регистр процессора
 - У каждого процесса свой «виртуальный» Program Counter
- Принцип разделения времени
 - Процессорное время выделяется процессу очень маленькими частями
 - Создается иллюзия того, что одному процессу принадлежит вся вычислительная система

Процессы

- Операционная система осуществляет
 - Создание/завершение процессов
 - Приостановку/возобновление процессов
 - Синхронизацию для организации доступа к общим ресурсам
 - Межпроцессное взаимодействие
 - Разрешение тупиковых ситуаций

Управление памятью

- Основная память
 - Большой массив байтов
 - Адресуемая единица хранения информации – «слово»
 - Напр. 32 битная процессорная архитектура адресует 32-битные/4-байтные слова
 - Непостоянная
 - При сбое в системе, память теряет своё содержание
 - Репозиторий для данных, к которым ЦП получает доступ (и устройств ввода/вывода в случае отображаемого в память ввода/вывода)

Управление памятью

- Менеджер памяти реализует
 - Изоляцию процессов
 - Автоматическое выделение памяти и управление
 - Поддержку модульного программирования
 - Защиту и контроль доступа
 - Долговременное хранение
- Современные механизмы управления памятью
 - Страничная память
 - Виртуальная память

Страничная память

- Память организована в блоки фиксированного размера, называемые страницами
- Это позволяет управлять и выделять память «виртуально»
 - Страница памяти может быть где угодно в физической памяти
 - Процессу выделяется множество страниц (им не обязательно располагаться одним блоком в физической памяти)
- Программы обращаются к памяти с помощью «виртуального адреса»
 - Мы можем дать программам ощущение того, что у них у всех одинаковый начальный адрес
 - Процессор вычисляет физический адрес на основе номера страницы + смещения внутри страницы
- Позволяет динамически строить отображение адресное пространство процесса и реальной физической памяти

Виртуальная память

- Виртуальная память расширяет принцип страничной организации за счёт использования вторичной памяти
- Убирает ограничения на саму физическую память и таким образом даёт процессам ощущение того, что памяти больше, чем есть физически
- Использует замену страниц и механизм подкачки (своппинг) страниц из памяти на диск и наоборот
- Позволяет выполнять много процессов – при переключении контекста может потребоваться выгрузить память процесса на диск

Кэширование

- Кэширование – важный принцип, выполняемый на многих уровнях в компьютере (на уровне «железа», ОС, ПО)
 - Используемая информация копируется из медленной памяти в быструю для обработки
 - С диска в основную память
 - Из основной памяти в кэш память процессора
 - Из кэш памяти процессора в регистры
- Сперва производится проверка, есть ли уже нужная информация в быстрой памяти (кэш)
 - Если есть, то обработка производится быстро
 - Если нет, то данные нужно сначала загрузить из медленной памяти

Иерархия запоминающих устройств

- Регистры
- Кэш процессора
- Основная память
- Электронный диск
- Магнитный диск
- Оптический диск
- Магнитная кассета

Кэширование

- Функция отображения
 - Определяет, какой блок кэша будет занят
- Проблема
 - Кэш может быть заполнен, нужно выбрать один блок для замены новым блоком
 - Аппаратная поддержка для такого поиска может быть сложной
- Нужен алгоритм, который может быть реализован на аппаратном уровне

Кэширование

- Алгоритм замены
 - Last Recently Used (LRU). Заменяет самый «старый» блок в кэше
 - Требуется аппаратный механизм определения такого блока
- Политика записи
 - Когда содержимое кэша меняется, его необходимо записать обратно в основную память
 - Когда должна происходить операция записи
 - Каждый раз при обновлении блока
 - При замене блока: минимизируется время записи, но основная память после записи в кэш будет содержать устаревшие данные

Планирование

- Планирование – процедура, которая определяет последовательность действий
 - Решает, когда процесс готов к выполнению, когда ему выделяется ЦП и насколько долго, когда происходят операции ввода/вывода
- Алгоритмы планирования оказывают сильное влияние на общую производительность системы

Планирование

- Долгосрочное
 - Планировщик принимает решение о добавлении программы в пул процессов для выполнения
 - Планирование задачи: программа подготавливается для выполнения (создается контекст процесса)
- Среднесрочное
 - Выгрузка и обратная загрузка процессов
 - Планировщик принимает решение поместить процесс в очередь диспетчера (процесс частично или полностью находится в памяти, ресурсы выделены)
- Краткосрочное
 - Диспетчеризация, планирование ЦП
 - Планировщик принимает решение какой из доступных процессов в памяти будет выполняться на процессоре
- Планирование Ввода/вывода
 - Планировщик решает, когда обработать запрос на отложенный ввод/вывод

Планировщик

- Операционная система организует эффективное и справедливое управление вычислительными ресурсами
 - Определяет, какой из доступных в памяти процессов будет выполнен на процессоре
 - Определяет, какой именно процесс будет выполняться и как долго
 - Реагирует на внешние события, такие как прерывания ввода/вывода
 - Использует алгоритм планирования, который пытается оптимально использовать ЦП, увеличить быстродействие, уменьшить время ответа (в зависимости от требований к ОС)

Управление вводом/выводом



Управление вводом/выводом

- Два метода
 - Синхронный В/В
 - После того, как начался В/В, пользовательская программа вынуждена ожидать, пока В/В завершится (например, распечатается текст на принтере)
 - Инструкция ожидания останавливает ЦП до поступления следующего прерывания на В/В
 - Нет одновременной обработки операций В/В
 - Асинхронный В/В
 - После начала В/В, управление сразу же возвращается пользовательской программе (и ОС в целом)
 - Программы выполняет системный вызов

Управление вводом/выводом

- Для управления доступом к устройствам ввода/вывода
 - Драйверы устройств скрывают специфику от пользователя
 - ОС предоставляет пользовательским программам независимый от устройства API для программистов
 - ОС выдаёт низкоуровневые команды (выполняет функции драйвера устройств) устройствам, обрабатывает прерывания и ошибки
 - Управляет буферами В/В и планированием операций В/В