Лабораторная работа: «Определение архитектуры операционной системы в Windows»

Цель работы: *Изучить и практически определить ключевые характеристики архитектуры ОС Windows (разрядность, версию, ядро, основные компоненты), а также понять, как эти параметры влияют на работу системы и совместимость ПО.*

Планируемые результаты

После выполнения работы студент должен:

- уметь определять версию и разрядность Windows штатными средствами;
- понимать разницу между пользовательским и ядерным режимами в архитектуре Windows;
- интерпретировать данные о системе для решения практических задач (установка драйверов, выбор ПО и т. п.).

Оборудование и ПО

- ПК с ОС Windows (10/11);
- стандартные утилиты ОС: cmd, PowerShell, «Сведения о системе», «Диспетчер задач»;
- (опционально) утилита wmic или современные PowerShell-командлеты.

Теоретическая справка (кратко)

Архитектура Windows — это многоуровневая модульная система:

- **Ядро (Kernel)** — низкоуровневые функции (управление памятью, процессами, прерываниями).
- **Исполнительная система (Executive)** — управление объектами, безопасностью, вводом-выводом.
- **Пользовательский режим** — приложения и подсистемы (Win32, POSIX и др.).
- **Драйверы устройств** — работа с оборудованием.
- **HAL (Hardware Abstraction Layer)** — абстракция от железа, чтобы ядро не зависело от конкретной платформы.

Ключевые архитектурные параметры для практики:

- разрядность (32- или 64-битная);
- версия и сборка ОС;
- тип ядра (в современных Windows — гибридное ядро);
- архитектура процессора (x86, x64, ARM).

Ход работы

Задание 1. Определение версии и разрядности ОС

1. Откройте «Параметры» → «Система» → «О системе».
2. Выпишите:
   - версию Windows (например, Windows 11 23H2);
   - тип системы (64-разрядная операционная система, процессор x64).
3. Альтернативно: откройте командную строку (cmd) и выполните:

systeminfo | findstr /C:"OS Name" /C:"System Type"

4. Сравните результаты из двух источников.

Что зафиксировать в отчёте: версия ОС, разрядность, тип процессора, источник данных.

Задание 2. Получение детальной информации о системе

1. Запустите утилиту «Сведения о системе» (msinfo32).
2. Изучите разделы:
   - «Сведения о системе»: имя ОС, версия, сборка, тип системы.
   - «Компоненты» → «Дисплей»: информация о видеокарте и драйверах.
   - «Программная среда» → «Системные драйверы»: список загруженных драйверов.
3. Сделайте скриншоты ключевых окон.

Что зафиксировать: основные параметры из msinfo32, список 3–5 системных драйверов с их назначением (например, nvlddmkm.sys — драйвер видеокарты NVIDIA).

Задание 3. Анализ процессов и режимов работы

1. Откройте «Диспетчер задач» → вкладка «Подробности».
2. Найдите процессы: explorer.exe, svchost.exe, csrss.exe.
3. Определите для каждого:
   - разрядность процесса (если отображается);
   - предполагаемый режим работы (пользовательский или ядерный контекст).

4. **Запустите PowerShell** (через меню «Пуск» или Win + X → Windows PowerShell).

5. Получение информации о системе

1. Получите краткие сведения об ОС:

```
Get-CimInstance Win32_OperatingSystem | Select-Object Caption, Version, BuildNumber
```

2. Узнайте архитектуру системы:

```
[Environment]::Is64BitOperatingSystem
```

3. Получите данные о процессоре:

```
Get-CimInstance Win32_Processor | Select-Object Name, NumberOfCores, MaxClockSpeed
```

4. Получите данные об оперативной памяти:

```
Get-CimInstance Win32_PhysicalMemory | Select-Object Capacity, Speed, Manufacturer
```

Что осваиваем: работу с CIM-классами, выбор конкретных свойств через Select-Object, вывод булевых значений.

6. Фильтрация и сортировка данных

1. Выведите список запущенных процессов, отсортированный по потреблению памяти (в МБ), и покажите только топ-5:

```
Get-Process | Sort-Object WorkingSet -Descending | Select-Object -First 5 Name, WorkingSet
```

Что осваиваем: конвейер (|), сортировку (Sort-Object), фильтрацию (Where-Object)

7. В PowerShell выполните:

Get-Process | Select-Object Name, Id, ProcessorArchitecture

4. Проанализируйте вывод: какие процессы 32-битные, какие 64-битные.

Что зафиксировать: таблица с процессами, их разрядностью и кратким описанием роли в архитектуре ОС.

Задание 5. Практический вывод: как архитектура влияет на установку ПО

На основе собранных данных ответьте на вопросы:

1. Можно ли установить 64-битный драйвер на 32-битную ОС? Почему?
2. Что произойдёт, если запустить 64-битное приложение на 32-битной Windows?
3. Как HAL помогает Windows работать на разном оборудовании?

Требования к отчёту

Отчёт должен содержать:

- титульный лист;
- цель работы;
- ход работы с заполненными таблицами и скриншотами;
- ответы на вопросы из Задания 5;
- общий вывод: какие инструменты оказались наиболее удобными для определения архитектуры Windows и почему.

Контрольные вопросы для защиты

1. Чем отличается 32-битная архитектура от 64-битной с точки зрения адресного пространства?
2. Что такое HAL и зачем он нужен в архитектуре Windows?
3. Почему в Windows есть отдельные папки Program Files и Program Files (x86)?
4. Как по списку драйверов можно судить об аппаратной конфигурации ПК?
5. В чём разница между процессом и потоком в контексте архитектуры ОС?

Методические рекомендации

- Все команды выполняйте в обычной командной строке или PowerShell без прав администратора, если явно не требуется иное.
- Для скриншотов используйте встроенные средства Windows (Win + Shift + S) или любой удобный инструмент.
- При анализе драйверов не пытайтесь их изменять — только просматривать и описывать.