# Системное программирование
# Л.р.8. Некоторые служебные и технологические задачи

## Цель:

Отдельные задачи, связанные с конфигурированием, мониторингом, управлением системой, а также технологические аспекты: библиотеки и сборки

## Теоретическая и методическая часть

(В зависимости от конкретной задачи)

Подходы к хранению и использованию конфигураций.
Реестр Windows: назначение; структура; типы и характер хранимых данных; интерфейс, группы функций API; порядок использования реестра и типичные задачи.

Журналирование: назначение, решаемые задачи, основные используемые подходы.
Журналы Windows: структура; виды хранимых данных; интерфейс, группы функций API; порядок использования журналов.

Механизм WinHook – перехват и обработка оконных сообщений Windows.

Библиотеки: назначение, место в технологическом процессе программирования; виды библиотек; использование библиотек.
Динамические библиотеки (DLL) Windows: структура модуля; динамическое и статическое подключение, явный и неявный импорт; импортируемые объекты (символы); типовой каркас библиотеки DLL; порядок использования библиотек DLL, типичные применения и проблемы.
Смешанные сборки (mixed assembly) Windows.

## Практическая часть

### Общая постановка задачи:

Приложение, демонстрирующее функциональность в соответствии с выбранным заданием. При необходимости – взаимодействие с другим ПО (прикладным и/или системным).

Для библиотек и сборок постановка задачи может включать написание подключаемого модуля, головной программы или всего программного комплекса.

**Варианты заданий:**

– работа с реестром: поиск и экспорт
– работа с реестром: контроль изменений
– работа с реестром: очистка
– просмотр информации о системе
– работа с журналами
– протоколирование оконных сообщений
– технологии многомодульных приложений
– ...

### 1 Работа с реестром: поиск и экспорт

Поиск в реестре ключей (ветвей реестра) и/или значений по заданному образцу:
– key – по имени; результат – иерархия подключей и значений данного ключа (фактически поддерево)
– value – по имени и/или собственно значению; результат – единственное значение или список значений.

Найденные данные выводятся на консоль и/или в файл, желательно в формате reg-файлов (перспектива возможного импорта обратно в реестр).

Интерфейс может быть минимальным (простейший GUI или командная строка).

### 2 Работа с реестром: контроль изменений

Утилита (утилиты), обеспечивающая сохранение «снимков» реестра (для сокращения объема обрабатываемых данных – части реестра) и в дальнейшем сравнение текущего состояния с сохраненными эталонами. Отображение найденных отличий (изменений) в удобном для анализа виде.

Опционально:
– выходной reg-файл как форма отображения информации об изменениях (файл, исполнение которого привело бы к аналогичным изменениям)
– выходной «инверсный» reg-файл, выполнение которого откатывает внесенные изменения
– возможность более детально задавать контролируемые ключи и значения

### 3 Работа с реестром: очистка

Утилита, находящая и удаляющая в реестре утратившие актуальность подключи и значения.
Ввиду сложности, разнообразия и неоднозначности критериев «неактуальности»,для поиска можно ограничиться некоторой их группой. Например, неактуальными можно считать значения, содержащие имена файлов, уже отсутствующих в файловой системе.

Дополнительный функционал: вывод сведений о выполненных изменениях, для контроля и для возможности откатить изменения.

### 4 Просмотр информации о системе

Утилита, обеспечивающая сбор информации о системе и отображение ее в удобном виде: аппаратное обеспечение, операционная система, количественные характеристики и т.д.

Состав информации – на усмотрение разработчика, в качестве ориентира – стандартные «системные» сведения, можно расширить или специализировать на конкретном разделе.

Источники информации: в основном реестр и/или специализированные системные функции, содержимое файловой системы, WMI, собственные измерения характеристик и т.д.

### 5 Работа с журналами

Добавление функционала (демонстрационного уровня) записи событий в журналы в приложение, например от заданий предыдущих лабораторных работ.

Состав протоколируемых событий – можно ограничиваться уже определенными типами, т.е. без регистрации собственных новых типов событий.

Опционально:
– управление уровнем подробности протоколирования
– средства поиска «своих» сообщений в журналах.

### 6 Протоколирование оконных сообщений

Перехват с помощью WinHook и запись в файл сообщений, поступающих в заданное окно (окна заданного приложения).

Возможный подход к реализации: программа, получающая исполняемый файл и порождающая из него child-процесс, к которому применяются WinHook-и (parent-процесс имеет существенно больше прав на свой child, чем на произвольный «посторонний» процесс).

### 7 Технологии многомодульных приложений

Перестройка уже имеющейся программы (например, от предыдущих лабораторных) с выносом части функционала в библиотеку, DLL и т.п.

### 8 ...