**ТЕМЫ КУРСОВЫХ ПРОЕКТОВ «ОПЕРАЦИОННЫЕ СИСТЕМЫ И СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ»**

1. **Синхронизация процессов при работе с общим файлом.**

Создать приложение – например с четырьмя процессами-писателями и одним читателем. Каждый процесс писатель в случайный момент времени пишет в файл число (случайное целое число от 1 до 100). Процесс-читатель считывает число и находит его квадрат. Помещает квадрат в другой файл (также общий для всех процессов), откуда его считывает соответствующий процесс-писатель. Синхронизацию сделать на основе механизма критической секции. Проект реализовать на с++. Продумать схему работы системы. В отчете в теоретической части описать механизмы синхронизации в современных ОС.

**2. Синхронизация потоков при работе с общей ячейкой памяти.**

Создать приложение – например с двумя потоками-вкладчиками и двумя потоками-потребителями. Каждый поток-вкладчик в случайный момент времени пишет в ячейку (случайное целое число от 1 до 100), представляющую сумму, помещаемую на счет. Поток-потребитель считывает число (снимает случайную сумму с вклада). Синхронизацию сделать на основе механизма семафора или критической секции. Проект реализовать на с++. Продумать схему работы системы. В отчете в теоретической части описать механизмы синхронизации в современных ОС. При выполнении операции на счете поток выводит на консоль свой номер, величину снимаемой (докладываемой) суммы и значение остатка.

**3. Программная реализация алгоритма банкира.**

Изучить алгоритм банкира. Разобраться с его реализацией для выделения памяти процессам в динамике. Создать динамически пополняемую очередь процессов. Для каждого процесса, поступающего в очередь в случайное время, получить запрос памяти как случайное число от 10 до 100. Общий объем памяти взять равным 200. На консоль выводить, какой процесс получает память, ее объем, объем свободной области. Оценить среднее время пребывания процесса в системе по результатам моделирования. Написать программу на с (с++).

**4. Программная реализация алгоритма Джонсона.**

Изучить эвристическую модификацию алгоритма Джонсона для планирования процессов на n>2 ресурсах. Условно говоря, матрица затрат разбивается на две подматрицы (левая подматрица рассматривается как первый ресурс, вторая – как второй). Использовать точный алгоритм Джонсона для двух ресурсов. Построить программную реализацию. Найти и представить информацию по использованию алгоритма Джонсона в задачах планирования вычислительного процесса. Провести программный эксперимент с построением диаграмм Ганта.

1. **Моделирование сетевого взаимодействия процессов по сети на основе сокетного TCP-соединения.**

Один процесс работает как сервер и обслуживает процессы-клиенты. Каждый процесс-клиент передает на сервер случайное целое число в диапазоне от 1 до 100. Сервер возводит его в квадрат и возвращает клиенту. Клиент выводит на консоль исходное число и значение, возвращенное сервером. Через некоторое случайное время клиент повторяет запрос, но с другим числом. Вывести результаты работы всего комплекса, ограничив число запросов клиентов, равным 10.

В отчете описать технологию сокетного TCP-взаимодействия, основные API, привести листинг программы с описанием ее работы, привести скриншоты, показывающие работу системы в динамике.

1. **Моделирование сетевого взаимодействия процессов по сети на основе сокетного UDP-соединения.**

(UDP – User DataGram Protocol)

Один процесс работает как сервер и обслуживает процессы-клиенты. Каждый процесс-клиент передает на сервер случайное целое число в диапазоне от 1 до 100. Сервер возводит его в квадрат и возвращает клиенту. Клиент выводит на консоль исходное число и значение, возвращенное сервером. Через некоторое случайное время клиент повторяет запрос, но с другим числом. Вывести результаты работы всего комплекса, ограничив число запросов клиентов, равным 10.

В отчете описать технологию сокетного UDP-взаимодействия, основные API, привести листинг программы с описанием ее работы, привести скриншоты, показывающие работу системы в динамике.

1. **Управление динамиком через порт.**

Написать программу, которая воспроизводит несколько музыкальных мелодий в случайном порядке, используя порт динамика. Программу написать на языке Ассемблер. Управление звуком реализуется подачей на порт динамика частоты звучания. В отчете описать принципы управления внешними устройствами (например, мышью, клавиатурой, динамиком) через ассемблерный код и функцию прерывания 21h.

1. **Работа через COM-порт.**

Продемонстрировать использование COM-порта в качестве файла. Привести сведения по использованию COM-порта и работе с ним из с++(с).

Написать программу, которая записывает текст и считывает текст из COM-порта. Реализовать работу двух потоков через COM-порт (один поток пишет в COM-порт, второй читает c COM-порта).

1. **Работа с файлом, отображаемым в память.**

Изучить принципы и системные API для отображения файлов в память. В отчете описать эти API. Написать программу с двумя процессами – один пишет данные в файл, отображаемый в память, а второй читает. Использовать механизм синхронизации, например, на основе критической секции.

1. **Работа с потоками.**

Изучить системные API для работы с потоками. Рассмотреть способы обмена данными между потоками через ячейку памяти. Написать приложение, где один из потоков «потребитель» - снимает случайное число денег со счета, а поток заносит новое пополнение на счет. На консоли отобразить динамику счета. Предусмотреть передачу параметра между потоками, а именно – запрос потребителя снять деньги у поставщика. Деньги снимаются, только если поток-поставщик даст разрешение.

1. **Разработка процедуры завершения асинхронной операции с файлом, отображенным в памяти**

Изучить принципы реализации процедур завершения асинхронных операций с файлами. Рассмотреть назначение системного API. Процедура завершения асинхронных операций с файлами запускается в следующем случае:

* над файлом выполняется асинхронная операция ввода-вывода;
* когда асинхронная операция ввода-вывода будет завершена, запускается указанная процедура завершения.

До начала операции ввода-вывода основной поток делает файл скрытым.

Создать процедуру завершения, привязанную к асинхронной операции ввода-вывода в файл, спроецированный в память. Процедура завершения должна снова делать файл видимым.

1. **Использование команд Windows Script Shell для управления программой.**

Написать скрипт, который запускает макрос в Excel. Макрос рисует диаграмму. Через какое-то время скрипт закрывает окно Excel. В отчете описать язык Windows Script Shell и примеры его использования в VBS (или JavaScript). Описать решение поставленной задачи. Привести скриншоты.

1. **Использование команд Windows Script Shell для управления программой.**

Написать скрипт, который запускает макрос в PowerPoint. Макрос запускает презентацию на выполнение. В отчете описать язык Windows Script Shell и примеры его использования в VBS (или JavaScript). Описать решение поставленной задачи. Привести скриншоты.

1. **Передача сообщений из одного оконного приложения в другое.**

Сообщения посылать по команде SENDMESSAGE. Предусмотреть два типа сообщений – для сворачивания окна и развертки окна. В отчете описать средства Windows для работы с сообщениями: POST MESSAGE, каналы PIPE, очереди сообщений – QUEUEMESSAGE. Привести реализованную программу. Рабочий язык с++ (с).

1. **Управление консолью**

Реализовать через WINAPI бегущую строку в консоли. Строка каждый раз изменяет размер букв и цвет букв. В отчете описать существующее API для управления консолью и буфером экрана. Привести текст программы и скриншоты.

1. **Планирование запуска процессов.**

Написать приложение для запуска IEXPLORER через 40 секунд после включения питания и завершения IEXPLORER через 2 минуты. В отчете описать API для планирования задач и системную оснастку. Привести программную реализацию на языке с (с++) с описанием работы.

1. **Шифрование/дешифрование файлов**.

Написать приложение для шифрования/дешифрования текстового файла на основе алгоритма AES. Ключ хранить в реестре, доступ к ключу по паролю. В отчете изложить системные средства Windows для шифрования, а также описать алгоритм AES. Описать работу с реестром.

1. **Рисование на холсте (CANVAS)**

Запрограммировать метод onDraw оконной процедуры для рисования графика многочлена вида *xn*+*a*1*xn*-1+...+*an*-1*x*+*b*. На графике отобразить текст с определением многочлена. Использовать цветовую заливку. В отчете описать принципы работы с холстом и системные API. Получить также несколько различных графиков на холсте.

1. **Работа с медиаплейером.**

Написать приложение на с (с++) для проигрывания музыкальных файлов. Проигрывание файла выполнить в отдельном потоке. Предусмотреть возможность приостанова и возобновления музыкального воспроизведения. В отчете дать описание API.

1. **Работа с файлами**

Написать приложение для поиска текстового файла, имя которого, возможно, содержит ошибку. Продумать, как использовать стандартный поиск на основе wild-символов \* и ?. Найденный файл открыть в блокноте.

1. **Создание ActiveX**

Создать компонент ActiveX в среде Visual Studio MFC. Компонент должен читать атрибуты файла (имя файла вводить с консоли). Прочитанные атрибуты вывести на консоль. В отчете описать технологию создания и регистрации компонентов ActiveX, а также работу с атрибутами файла.

1. **Работа с реестром**

Написать программу для работы с реестром. Программа позволяет выводить url, посещаемые пользователем, и при необходимости удалять их из реестра.