

Задания к работе №1 по Операционным системам.

Все задания реализуются на языке программирования C (стандарт C99 и выше).

Реализованные приложения не должны завершаться аварийно.

Все ошибки, связанные с операциями открытия файла, должны быть обработаны; все открытые файлы должны быть закрыты.

1. В этом задании используйте функционал: FILE*, fopen, fclose, fwrite, fread, fseek.

Создайте файл (путь к файлу передаётся как аргумент командной строки), в котором содержится следующая последовательность байт: 3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5. После создания файла откройте и **прочитайте** этот файл с побайтовым выводом содержимого этого файла на консоль. На каждой итерации побайтового чтения также выводите в консоль **состояние всех полей** структуры FILE, через указатель на которую взаимодействуете с файлом. Закройте файл и откройте его на чтение снова. С помощью функции fseek выполните перемещение указателя чтения записи на 3 относительно начала файла (флаг SEEK_SET). С помощью функции fread считайте 4 байта из файла в байтовый буфер. Что будет содержать буфер, когда завершится работа функции fread?

2. Напишите примитивную оболочку для командной строки. При запуске вашей программы должно быть выведено приглашение к авторизации пользователя. У каждого пользователя есть login (длиной не более 6 символов, который состоит из символов латинского алфавита и цифр) и PIN-код (целое число в системе счисления с основанием 10, значение числа варьируется в диапазоне **0 до 100000**). Для работы с оболочкой пользователь должен авторизоваться или зарегистрироваться (интерфейс взаимодействия с пользователем для его регистрации продумайте самостоятельно) в приложении. После удачной авторизации пользователя ему доступен следующий набор команд:

- **Time** - запрос текущего времени в стандартном формате **чч:мм:сс**;
- **Date** - запрос текущей даты в стандартном формате **дд:мм:гггг**;
- **Howmuch <time> flag** - запрос **прошедшего времени** с указанной даты в параметре <time>, параметр **flag** определяет тип представления результата (**-s в секундах, -m в минутах, -h в часах, -y в годах**)
- **Logout** - выйти в меню авторизации
- **Sanctions username <number>** - команда позволяет ввести ограничения на работу с оболочкой для пользователя username, а именно данный пользователь не может в одном сеансе выполнить более <number> запросов. Для подтверждения ограничений после ввода команды необходимо ввести значение 12345.

3. Напишите консольное приложение для **копирования файлов**. Аргументы в Ваше приложение передаются как аргументы командной строки: первый параметр — это имя файла, который является источником данных, второй параметр — это имя файла, в который необходимо выполнить копирование. Данные во входном файле могут быть **произвольной структуры**; структура содержимого во входном файле не должна влиять

на процесс копирования; входной и выходной файл должны быть идентичны после выполнения операции копирования.

4. Реализуйте консольное приложение для побайтовой обработки файлов. Аргументы в Ваше приложение передаются как аргументы командной строки: первый параметр — путь ко входному файлу, второй параметр — это флаг, который определяет действие, которое необходимо выполнить с файлом:
- **xor8** - сложение по модулю 2 всех байтов файла;
 - **xor32** - сложение по модулю 2 групп четырехбайтовых значений из файла (если размер последней группы байтов файла $\neq 4$ байтам, недостающие байты заполняются значением 00_{16});
 - **mask <hex>** - подсчет четырехбайтовых целых чисел из файла, которые соответствуют маске <hex>, которая задана в системе счисления с основанием 16.