***1.Виртуальные и физические ресурсы в вычислительной системе. Понятие уровней взаимодействия компонент ВС (пользовательские программы, абстракции ОС, драйверы, аппаратура)***

*Вычислительная система- совокупность аппаратных и программных средств, функционирующих в единой системе и предназначенных для решения задач определенного класса. Структура ВС:*

*1)прикладные программы*

*2)системы программирования*

*3)управление логическими ресурсами*

*4)управление физическими ресурсами*

*5)аппаратура Компоненты каждого уровня, получив от уровня более высокого соответствующее обращение, переводят его в термины уровня более низкого и передают ниже, а полученный ответ переводят, наоборот, в термины уровня более высокого и отправляют наверх. Таким образом, по мере движения от аппаратуры к пользователю нарастает уровень абстрагирования, а сложность описания падает. Прикладная программа для вывода использует библиотечную функцию ввода-вывода высокого уровня. Библиотека функции, относящаяся к слою систем программирования переводит полученный запрос на язык, понятный более низкому; таким языком является интерфейс системных вызовов, а переведенный запрос становится вызовом с соответствующими параметрами. ОС на уровне управления логическими устройствами преобразует параметры системного вызова в последовательность действий, необходимых для записи полученной информации в файл. Драйвер файловой системы переводит запрос в последовательность операций по модификации секторов логического дискового устройства, а драйвер соответствующего логического диска в последовательность операций над секторами диска физического. Соответствующая последовательность запросов передается более низкоуровневым подпрограммам ядра, включающим драйвер физического диска. По окончании операции ввода-вывода сообщение об этом сначала получает обработчик соответствующего прерывания, который уже передает полученную информацию нужному драйверу. Обработчики прерываний относятся к слою управления физическими устройствами. Уровни взаимодействия компонент ВС:*

*1)пользовательские программы*

*2)аппаратно-независимые компоненты ОС*

*3)драйверы*

*4)обработчики прерываний*

*5)аппаратура*

* ***Файловые системы и устройства хранения данных***
* ***Типы файлов. права доступа на файлы и каталоги. Текущий каталог, системный вызов chroot. Индексный дескриптор, системные вызовы stat, fstat, lstat. Структура каталогов файловой системы UNIX, назначение каталогов и некоторых особенно важных файлов.***

*Под файлом в терминологии, связанной с компьютерами, обычно понимается некий набор хранимых на внешнем запоминающем устройстве данных, имеющий имя. Подсистема ОС, отвечающая за хранение информации на внешних запоминающих устройствах в виде именованных файлов, называется файловой системой. При создании на диске ФС создается один каталог, называемый корневым. В ОС UNIX ФС представляет собой единое дерево каталогов. В имя файла не входит имя устройства, на котором этот файл находится. chroot — операция изменения корневого каталога. Программа, запущенная с изменённым корневым каталогом, будет иметь доступ только к файлам, содержащимся в данном каталоге. Поэтому, если нужно обеспечить программе доступ к другим каталогам или файловым системам. В ОС UNIX каталоги хранят только имя файла и некоторый номер, позволяющий идентифицировать соответствующий файл. Вся прочая информация связывается с этим номером. Хранимая на внешнем запоминающем устройстве структура данных, содержащая всю информацию о файле, исключая его имя, называется индексным дескриптором. В ОС UNIX допускается, чтобы два или более имен файлов, расположенных как в разных каталогах, так и в одном, ссылались на один и тот же номер индексного дескриптора. Дополнительные имена файл может получить с помощью системного вызова int link(const char \*oldpath, const char \*newpath); где oldpath-существующее имя файла, а newpath- новое имя. Такие имена называются жесткими ссылками. Функция, предназначенная для удаления файла, называется unlink(). Чтобы создать жесткую ссылку средствами командной строки, можно воспользоваться командой ln(создание для файла нового имени). Создание жестких ссылок на каталоги система запрещает, это может привести к созданию циклов в дереве каталогов. В ОС UNIX поддерживается большое количество файлов специального типа: файлы байт-ориентированных и блок-ориентированных устройств, имена сокетов, именованные каналы и символические ссылки. Символическая ссылка представляет собой файл, содержащий имя другого файла. Операция открытия символической ссылки на чтение или запись приводит на самом деле к открытию файла, на который она ссылается, а не ее самой. В отличие от жесткой ссылки, символическая ссылка имеет свой собственный номер индексного дескриптора. Создание и удаление символической ссылки не затрагивает ни файл, ни его индексный дескриптор. Символические ссылки создаются вызовом int symlink(),удаление unlink(),создание командной строкой ln –s. Права доступа к файлу определяют, кто из пользователей какие операции может с данным файлом произвести. Права хранятся в индексном дескрипторе в виде 12-битного слова. Младшие 9 бит объединены в три группы по три бита; каждая группа задает права доступа для владельца файла, для группы владельца и для всех остальных пользователей. Три бита в каждой группе отвечают за право чтения файла, право записи в файл и право исполнения файла.ls –l (узнать права доступа к файлу). Для изменения прав доступа к файлам используется команда chmod. Чтобы начать работу с файлом, его необходимо открыть. Это делается системным вызовом int open(const char \*name, int mode); Параметр mode задает режим. Основные режимы O\_RDONLY, O\_WRONLY,O\_RDWR. Параметр perms следует задавать только в случае, если значение второго параметра, предполагает возможность создания файла(O\_CREATE). Чтение файла int read(int fd, void \*buf, int len); fd задает файловый дескриптор; buf указывает на буфер; len сообщает вызову размер буфера.*