

ПРИЛОЖЕНИЯ **UNIX** СИСТЕМ

Лекция №1

Ст. преподаватель
кафедры ПДСиМ
Квиткова Ирина Геннадьевна

История развития ОС UNIX

1965 г. - Bell Telephone Laboratories (подразделение AT&T) совместно с General Electric Company и Массачусетским институтом технологии (MIT) начали разрабатывать новую операционную систему, названную **MULTICS**.

1969 г. - Кен Томпсон и Дэннис Ритчи создали небольшую операционную систему, включавшую файловую систему, подсистему управления процессами и небольшой набор утилит. Эта ОС получила название UNIX.

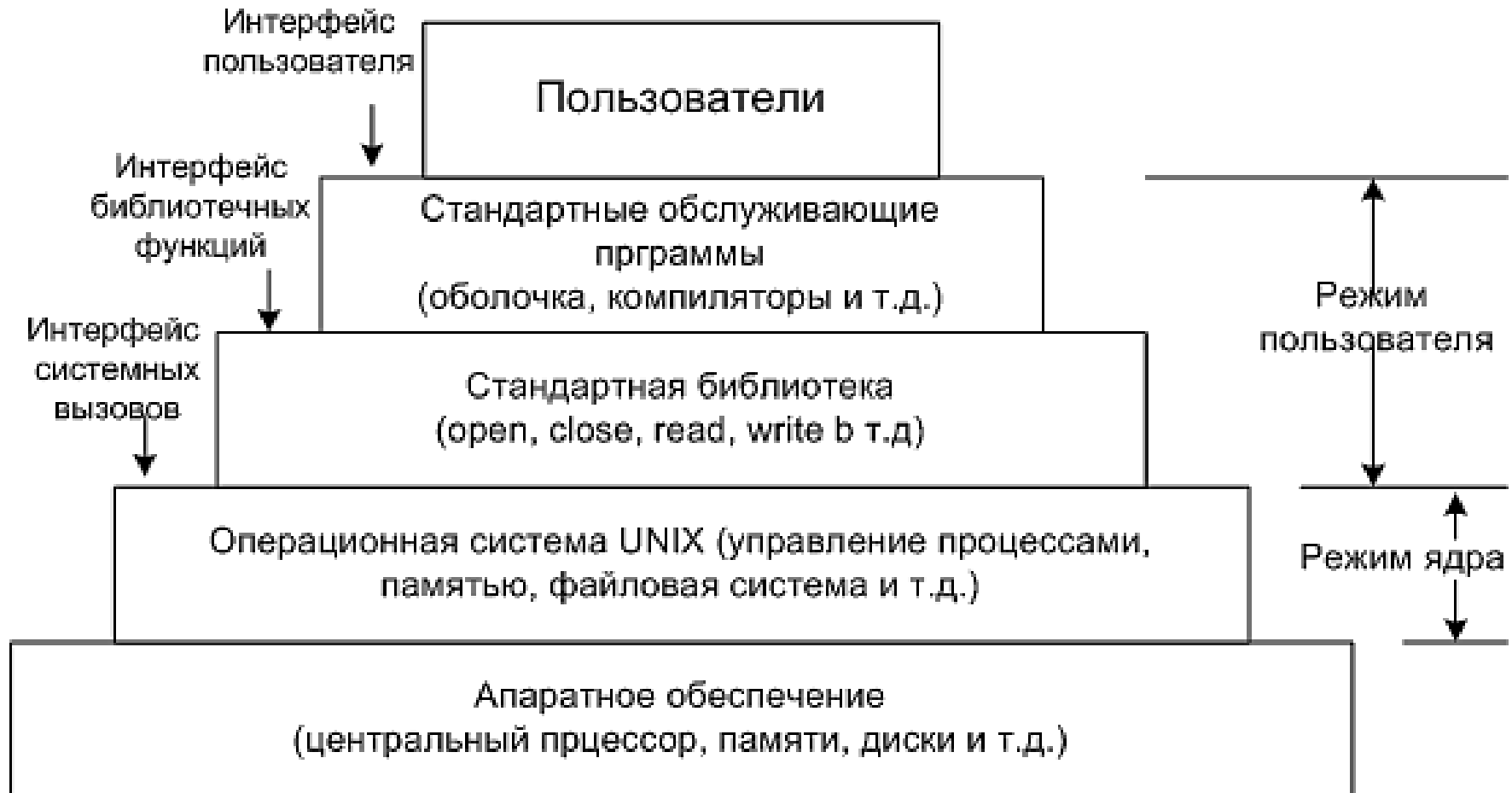
1971 г. - UNIX выбрана в качестве системы обработки текста.

1973 г. - ядро ОС UNIX переписано **на языке C**. UNIX может быть перенесена на другие аппаратные платформы за короткий срок. Для сопровождения UNIX сформирована группа UNIX System Group (USG).

Основные характеристики ОС UNIX

- Переносимость.
- Вытесняющая многозадачность на основе процессов, работающих в изолированных адресных пространствах в виртуальной памяти.
- Поддержка одновременной работы многих пользователей.
- Поддержка асинхронных процессов.
- Иерархическая файловая система.
- Поддержка независимых от устройств операций ввода-вывода (через специальные файлы устройств).
- Стандартный интерфейс для программ (программные каналы, IPC) и пользователей (командный интерпретатор, не входящий в ядро ОС).
- встроенные средства учета использования системы.

Архитектура ОС UNIX



Основные функции ядра

Основные функции ядра UNIX (которое может быть **монолитным** или **модульным**) включают:

- планирование и переключение процессов;
- управление памятью;
- обработку прерываний;
- низкоуровневую поддержку устройств (через драйверы);
- управление дисками и буферизация данных;
- синхронизацию процессов и обеспечение средств межпроцессного взаимодействия.

Системные вызовы

Системные вызовы (СВ) обеспечивают:

- ✓ сопоставление действий пользователя с запросами драйверов устройств;
- ✓ создание и прекращение процессов;
- ✓ реализацию операций ввода-вывода;
- ✓ доступ к файлам и дискам;
- ✓ поддержку функций терминала.

Они преобразуют процесс, работающий в режиме пользователя, в защищенный процесс, работающий в режиме ядра.

СВ обеспечивают программный интерфейс для доступа к процедурам ядра и оформлены в виде библиотеки времени выполнения.

Файловая система

ОС выполняет две основные задачи:
манипулирование данными и их **хранение**.



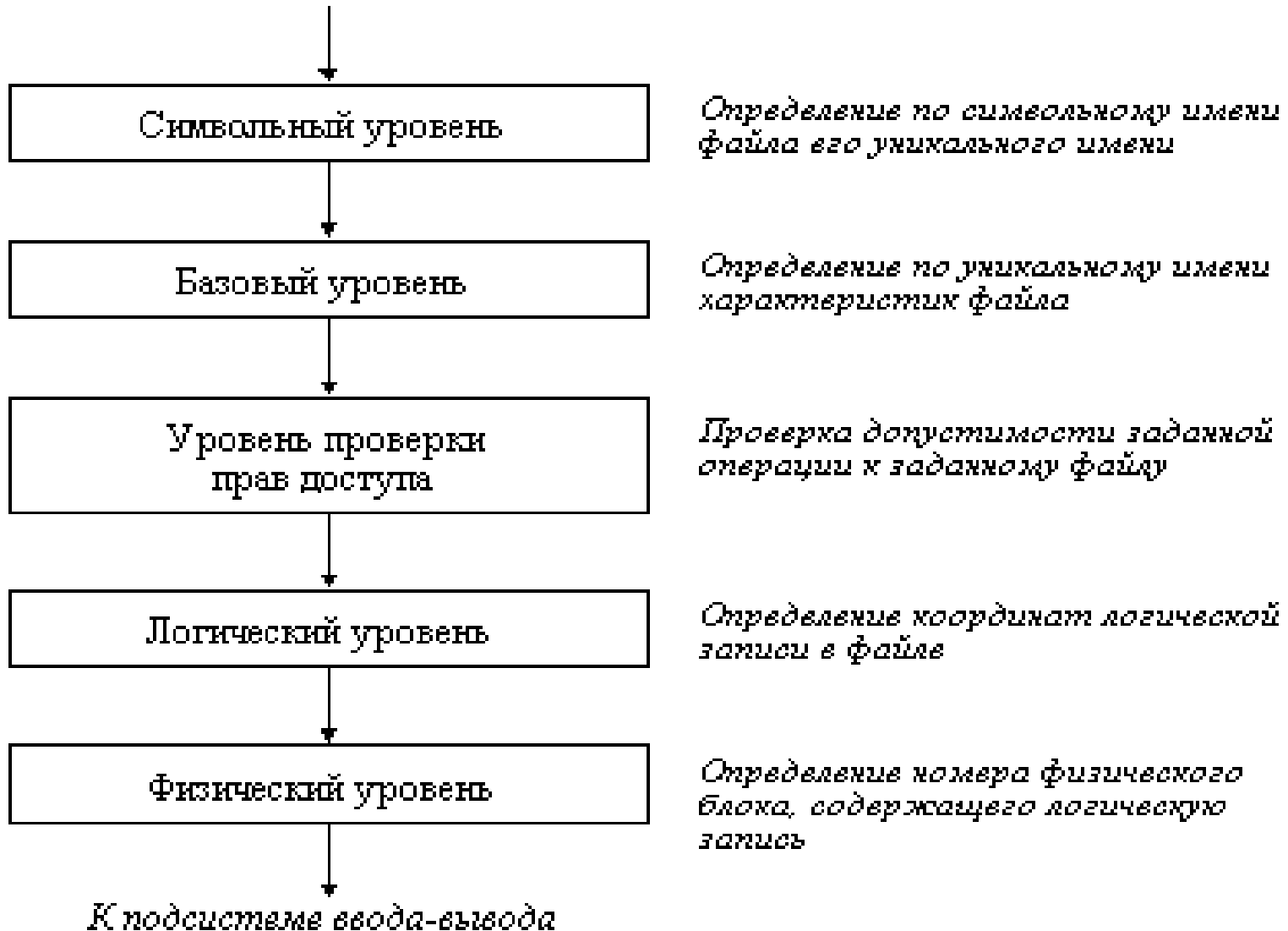
Файловая система (ФС)

- часть ОС, которая обеспечивает пользователю:
 - ✓ доступ к данным, расположенным на различных носителях, и к периферийным устройствам,
 - ✓ удобный интерфейс при работе с данными, хранящимися на диске,
 - ✓ совместное использование файлов несколькими пользователями и процессами.

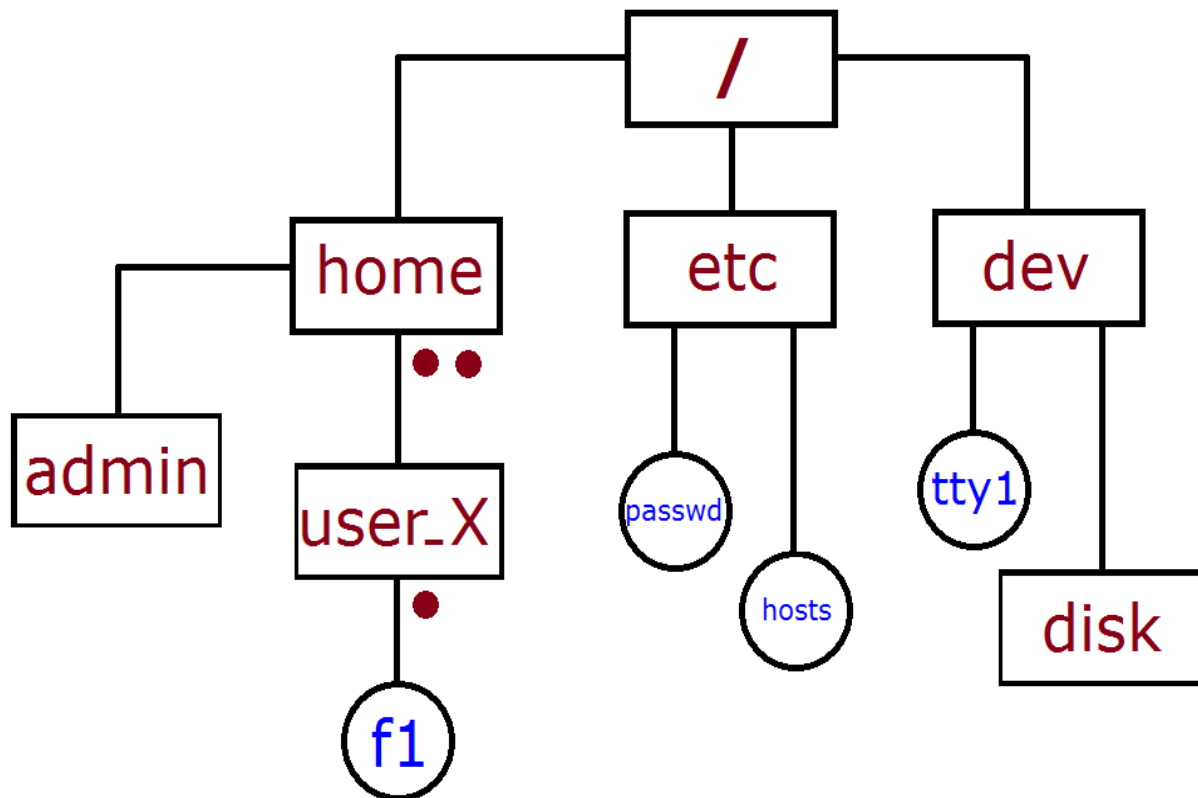
□ Логическая ФС в ОС UNIX - иерархически организованная структура всех каталогов и файлов в системе, начинается с корневого каталога.

Файловая система

Запрос к файлу
(операция, имя файла, логическая запись)



Файловая система



- – текущий каталог (ссылается сам на себя):
- **./f1** – файл f1 текущего каталога;
- **..** – родительский каталог (ссылается на каталог выше):
- **../admin** – каталог admin родительского каталога.

Файловая система

Логическое (символьное) имя файла:

абсолютное (полный путь к файлу от /):

/etc/passwd

/home/user_X/f1 = ~ /f1

(/home/user_X – домашний каталог пользователя)

относительное (относительно текущего каталога):

находясь в /home/user_X,

к файлу можно обратиться просто по **имени** (f1);

Аналогично обращение к каталогу:

/dev/disk – абсолютный путь (от корня /)

неправильно: dev/disk

disk – относительный путь, если находимся в
/dev

Иерархия каталогов в Unix

/	- Корневой каталог;
bin	- Файлы основных команд (утилит);
boot	- Файлы загрузки системы;
dev	- Файлы устройств;
etc	- Файлы конфигурации системы;
home	- Домашние каталоги пользователей;
lib	- Основные разделяемые библиотеки и модули ядра;
mnt	- Точка монтирования для временно; подключаемых файловых систем
root	- Домашний каталог суперпользователя;
opt	- Дополнительные пакеты ПО;
sbin	- Основные системные исполняемые файлы;
tmp	- Временные файлы;
usr	- Все установленные пакеты программ, справочная документация;
var	- Переменные данные (журналы ОС, кэш-файлы)

Типы файлов

- **Обычный файл (-):**
Блок информации, текст, программа;
- **Каталог (d):** информация о других файлах и каталогах;
- **Специальный файл устройства (b, c):**
Обеспечивает доступ к физическому устройству.
Реализует программный обмен информацией с устройствами;
- **Именованный канал (FIFO) (p):**
Системная структура для обмена данными между процессами;
- **Символическая ссылка (l):** Указатель на файл.
- **Сокет:** Системная структура для сетевого или межпроцессного взаимодействия.
- ✓ **Жесткая ссылка** – дополнительное имя для обычного файла. Основное имя файла также является жёсткой ссылкой.

Атрибуты файлов:

- ✗ тип файла (каталог, обычный, ссылка и т.д.);
- ✗ режим доступа (права на чтение, запись, запуск для 3-х видов пользователей);
- ✗ число жёстких ссылок (дополнительных имён);
- ✗ «владелец» файла;
- ✗ группа-владелец;
- ✗ размер файла в байтах;
- ✗ дата и время создания или последней модификации;
- ✗ имя файла.

```
-rwxr-xr-x  2  Andrey  A61  1024   Aug 30 2009  file-1
```

Права доступа к файлу

Три базовых класса доступа к файлу:

- User access (**u**) -для владельца-пользователя;
- Group access (**g**)-для членов группы, являющейся владельцем файла;
- Other access (**o**) - для остальных пользователей (кроме root).

Три типа прав доступа для каждого класса:

- ✓ на чтение (read - **r**);
- ✓ на запись (write - **w**);
- ✓ на выполнение (execute - **x**).

Операции изменения прав:

- '+' — добавить право,
- '-' — удалить,
- '=' — присвоить.

Символ «-» - права нет, «r, w, x» - право есть.

Команды работы с файлами

\$ команда –ключи параметры (через пробел)
Ключи – символы, уточняющие значение команды.
Параметры – имена файлов.

man *команда* (*команда --help*)

Справка ОС UNIX по команде и её ключам.

ls -ключи имя

Просмотр содержимого каталога (списка файлов).

ls -l /usr/sbin – получить подробную информацию о содержимом каталога /usr/sbin с атрибутами файлов;

ls -l *.c – получить сведения о файлах текущего каталога, имена которых заканчиваются на «.c» (* обозначаются все файлы каталога).

ls -R – рекурсивный просмотр текущего каталога.

cd каталог

Смена текущего каталога на указанный.

cd /usr/sbin – перейти в каталог */usr/sbin*;

cd .. – перейти в каталог выше (*parental*);

cd text – перейти в каталог *text*,
расположенный в текущем каталоге.

cd ~ - перейти в свой домашний каталог.

mkdir каталог

Создание каталога с указанным именем.

mkdir test

mkdir /home/john/programs

mkdir ../test

pwd - Выдает полное имя текущего каталога.

rmdir каталог

Удаление указанного каталога (без ключа
удаляет только пустой каталог).

touch файл

Создать файл с указанным именем (1-й способ).

touch a1 – *создать пустой файл a1.*

cat файл (less файл – просмотр вверх и вниз)

Выдает содержимое указанного файла на экран.

cat a1 – *посмотреть содержимое файла a1.*

cat > файл – 2-й способ создания файла.

(CTRL+c (z) – прекращение работы с командой.)

head, tail -n файл

Посмотреть только первые n строк (head) или последние n строк (tail) файла.

cp файл(ы) куда

Копирование файла в указанный файл или каталог.

cp ~/a1 a2 – *копирование файла a1 в файл a2;*

cp a1 /home/john/D1 – *файла a1, в указанный каталог D1.*

mv файл(ы) куда

Переименование или перемещение файлов.

mv a1 a2 – переименовать файл *a1* в *a2*.

mv a1 ~/D1 – переместить *a1* в каталог *D1*

rm файл(ы)

Удаление указанного файла или группы файлов, заданных шаблоном имени.

rm a1– удаление файла *a1*;

rm -rf * - очистка текущего каталога.

ln файл имя_ссылки

Создание жёсткой ссылки на указанный файл с указанным именем.

ln a1 D1/D2/a1_h – создать жёсткую ссылку с именем *a1_h* в каталоге *D2* на файл *a1*.

ln -s файл имя_ссылки

Создание символической ссылки.

vi файл (vim-tiny файл)

Редактирование файла.

(если указанный файл существует, он будет открыт в редакторе, иначе редактор создаст новый файл с заданным именем)

Клавиша

i, a – режим редактирования;

ESC – выход из режима редактирования;

:w – сохранение текущего файла;

:w file – сохранение файла под именем *file*;

:wq - сохранение файла и выход из *vi*;

:r file - вставить в текущий файл содержимое файла *file*.

nano файл – тоже редактор файла.

ctrl+o enter – сохранить изменения;

ctrl+x – выход из редактора.

chmod mode file

Изменяет права доступа к файлу

chmod g-wx a1 – у группы забрать права на запись и исполнение файла a1.

chmod a+w a1 – всем добавить право на запись

chmod go=r a1 – для файла a1 присвоить права только для группы и остальных на чтение.

chmod 777 a1 – у всех все права.

Числовой способ изменения прав доступа:

1 – право есть; **0** – права нет.

Пример: хотим ***rwxr-xr--***, т.е.

все права владельцу (rwx = 111)

группе – чтение и запуск (r-x = 101)

остальным – только чтение (r-- = 100).

Переводим полученные двоичные числа в десятичные: 111 = 7, 101 = 5, 100 = 4.

Вводим команду **chmod 754 file**.

grep “образец” файл

Поиск образца в строках текста файла.

grep 3 * – найти 3 в строках всех файлов
текущего каталога.

grep vova a1 – показать строки файла a1 с «vova».

Ключи:

- w** – поиск по целому слову;
- v** – вывести всё, кроме указанного образца;
- o** – вывести только указанный образец.

Пример шаблона:

grep "[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}"
/etc/resolv.conf – поиск строк с IP-адресами в тексте
файла /etc/resolv.conf.

grep "\<x" file – поиск **x** в начале слов файла file.

grep "x\>" file – поиск **x** в конце слов файла file.

grep "^x" file – поиск **x** в начале строки.

grep "x\$" file – поиск **x** в конце строки.

cut -fn -d"разделитель" файл

Вывести содержимое n-го столбца файла с указанным разделителем.

cut -f2,5 -d" " a1 – вырезать 2-й и 5-й столбцы файла a1, отделённые пробелом.

cut -f2 -d":" a1 – вырезать 2-й столбец, отделённый двоеточием.

sort файл

Сортирует строки файла (по алфавиту).

-r - сортирует в обратном порядке;

-o файл - сохраняет результат сортировки в выходном файле;

-u - выводит повторяющиеся строки только один раз;

-n - сортирует по числовому коду символа;

-kn - по указанному номеру поля (столбцу).

sort -r a1 -o a2 – отсортировать файл a1 в обратном порядке, результат вывести в файл a2.

find имя

Поиск файла (каталога) в файловой системе.

find /usr -name "*.html" -type f

wc -ключ файл

Подсчет символов, строк и слов.

-l – количество строк,

-w – количество слов,

-c – количество символов.

wc -l /etc/passwd – вывести число строк в файле */etc/passwd* (число зарегистрированных в системе пользователей).

echo параметры

Выдает на экран свои параметры.

echo "Hello People"– выдать на экран
«Hello people».

du каталог – показывает размер каталога.

Перенаправление стандартного вывода (Standard Output)

Используется знак >:

cat a1 - вывод на экран;

cat a1 > a2 - вывод в файл a2;

cat a1 >> a2 - добавить в конец файла a2
содержимое файла a1.

Перенаправление стандартного ввода (Standard Input)

Используется знак <:

mail - ввод сообщения с экрана;

mail < mes - ввод сообщения из файла mes.

Соединение команд конвейером (pipe)

Обозначается знаком |.

Результат выполнения одной команды переходит на ввод другой команды.

who | wc -l - подсчет числа активных
пользователей;

ls -l /tmp | grep maryann | sort -u | lpr
листинг поиск записей сортировка печать

Стандартный поток ошибок (Diagnostic Output)

Сообщения об ошибках, возникающих при выполнении команд, выводятся на экран.

Поток ошибок обозначается **2**.

Его можно перенаправить в какой-либо файл.

grep 3 * 2> ~/error – сообщения о том, что в каких-то файлах не найдены строки с 3, отправлены в файл *error* домашнего каталога.

ls -l D1 2>error – поток ошибок отправлен в файл текущего каталога.