***1)Понятие о программном обеспечении ЭВМ.Состав,Структура,многоуровневый иерархический характер.***

**Понятие программного обеспечения.   
Классификация программного обеспечения.**

Программное обеспечение-это совокупность программ, выполненных вычислительной системой.

 К программному обеспечению (ПО) относится также вся область деятельности по проектированию и разработке (ПО):

* технология проектирования программ (нисходящее проектирование, структурное программирование и др.)
* методы тестирования программ.
* методы доказательства правильности программ.
* анализ качества работы программ и др.

Программное обеспечение - неотъемлемая часть ЭВМ. Оно является логическим продолжением технических средств ЭВМ, расширяющие их возможности и сферу использования.

**Классификация программного обеспечения.**

|  |
| --- |
|  |
|  | http://www.infosgs.narod.ru/23.files/image001.gif |

Существует три категории:

**1) Прикладные программы**, непосредственно обеспечивающие выполнение необходимых пользователям работ.

**2) Системные программы**:

* управление ресурсами ЭВМ.
* создание копий используемой информации.
* проверку работоспособности устройств компьютера.
* выдачу справочной информации о компьютере и др..

**3)** **Инструментальные программные системы**, облегчающие процесс создания новых программ для компьютера.

 Более или менее определенно сложились следующие группы программного обеспечения:

* операционные системы.
* системы программирования.
* инструментальные системы.
* интегрированные пакеты.
* динамические электронные таблицы.
* системы машинной графики.
* системы управления базами данных (СУБД).
* прикладное программное обеспечение.

***2)Понятие об ОС.***

Операционная система (ОС) - комплекс системных и управляющих программ, предназначенных для наиболее эффективного использования всех ресурсов вычислительной системы (ВС) (Вычислительная система - взаимосвязанная совокупность аппаратных средств вычислительной техники и программного обеспечения, предназначенная для обработки информации) и удобства работы с ней.

Назначение ОС - организация вычислительного процесса в вычислительной системе, рациональное распределение вычислительных ресурсов между отдельными решаемыми задачами; предоставление пользователям многочисленных сервисных средств, облегчающих процесс программирования и отладки задач. ОС в значительной степени формирует у пользователя представление о возможностях ВС, удобстве работы с ней, ее пропускной способности. Различные ОС на одних и тех же технических средствах могут предоставить пользователю различные возможности для организации вычислительного процесса или автоматизированной обработки данных.

В соответствии с условиями применения различают три режима ОС: пакетной обработки, разделения времени и реального времени. В режиме пакетной обработки ОС последовательно выполняет собранные в пакет задания. В этом режиме пользователь не имеет контакта с ЭВМ, получая лишь результаты вычислений. В режиме разделения времени ОС одновременно выполняет несколько задач, допуская обращение каждого пользователя к ЭВМ. В режиме реального времени ОС обеспечивает управление объектами в соответствии с принимаемыми входными сигналами. Время отклика ЭВМ с ОС реального времени на возмущающее воздействие должно быть минимальным.

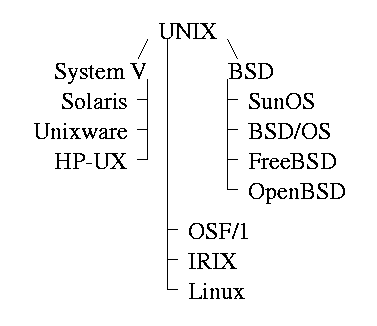
***3)Краткая историческая справка об ОС UNIX.***

ОС Unix была создана в 1969г. несколькими программистами Bell Labs фирмы AT&T "для себя", и работала на компьютере PDP-7. Через некоторое время большая ее часть была переписана с языка ассемблера на язык C, что дало возможность перенести ее на большое количество разных платформ. В настоящее время Unix работает на большинстве существующих архитектур, и для многих из них является основной ОС.

Дальнейшее развитие Unix, разработанного в AT&T и его производные называться System V (пятая версия), сокращенно SysV, иногда используется название "AT&T-версия Unix".

В середине семидесятых годов исходный код Unix попал в университет Беркли, где была создана своя версия Unix, получившая название "BSD Unix" (Berkeley Software Distribution).

В настоящее время большинство вариатнов Unix основаны или на SystemV, или на BSD.



Однако в последнее время предпринимаются усилия по выработке единого стандарта, и наиболее современные варианты Unix нельзя четко отнести ни к той, ни к другой ветви. В их число входят IRIX (компьютеры Silicon Graphics, пример в ИЯФ -- sky), Digital OSF/1 (компьютеры Alpha, пример в ИЯФ -- AxInp), и Linux.

В настоящее время UNIX используются в основном на [серверах](http://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80_%28%D0%B0%D0%BF%D0%BF%D0%B0%D1%80%D0%B0%D1%82%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5%29), а также как встроенные системы для различного оборудования. На рынке ОС для [рабочих станций](http://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B1%D0%BE%D1%87%D0%B0%D1%8F_%D1%81%D1%82%D0%B0%D0%BD%D1%86%D0%B8%D1%8F) и домашнего применения лидером является [Microsoft Windows](http://ru.wikipedia.org/wiki/Microsoft_Windows), UNIX занимает только второе ([Mac OS X](http://ru.wikipedia.org/wiki/Mac_OS_X" \o "Mac OS X)), третье ([GNU/Linux](http://ru.wikipedia.org/wiki/GNU/Linux)) и многие последующие места.

UNIX-системы имеют большую историческую важность, поскольку благодаря им распространились некоторые популярные сегодня концепции и подходы в области ОС и [программного обеспечения](http://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5). Также, в ходе разработки Unix-систем был создан язык [Си](http://ru.wikipedia.org/wiki/%D0%A1%D0%B8_%28%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F%29).

***4)Особенности ОС UNIX(режим работы, инструментальные и сервисные средства, язык С, TCP/P, Internet и пр.)***

* использование простых текстовых файлов для настройки и управления системой;
* широкое применение утилит, запускаемых в [командной строке](http://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BA%D0%BE%D0%BC%D0%B0%D0%BD%D0%B4%D0%BD%D0%BE%D0%B9_%D1%81%D1%82%D1%80%D0%BE%D0%BA%D0%B8);
* взаимодействие с пользователем посредством виртуального устройства — терминала;
* представление физических и виртуальных устройств и некоторых средств межпроцессового взаимодействия как [файлов](http://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB);
* использование [конвейеров](http://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D0%B5%D0%BD%D0%B0%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5_%28UNIX%29#.D0.9A.D0.BE.D0.BD.D0.B2.D0.B5.D0.B9.D0.B5.D1.80.D1.8B) из нескольких программ, каждая из которых выполняет одну задачу.
* Файловая система древовидная, чувствительная к регистру символов в именах, очень слабые ограничения на длину имён и пути.
* Нет поддержки структурированных файлов ядром ОС, на уровне системных вызовов файл есть поток байтов.
* Широкое использование текстовых файлов для хранения настроек, в отличие от двоичной базы данных настроек, как, например, в Windows.
* Широкое использование утилит обработки текста для выполнения повседневных задач под управлением скриптов.

Сетевые протоколы TCP/P

* Архитектура
* Протокольные модули
* Интерфейсы
* Адресация, DNS
* Маршрутизация,шлюзы

Основные службы Internet

* Почта(SMTP, POP, IMAP, UUCP)
* Передача файлов(FTP)
* Файловые серверы(NFS, SMB)
* Новостные группы(NNTP)
* Удаленные терминал и выполнение(telnet,rsh,ssh)

Сетевая безопасность

* Использование Firewall
* Защита inetd служб, tcpwrapper(tcpd)
* Proxy – серверы

Особая роль языка программирования Си состоит в том, что он, с одной стороны, позволяет писать для UNIX-систем практически столь же эффективный код, что и языки ассемблера, а с другой, является основным средством переноса программ между UNIX-системами. Можно сказать, что Си является машинно-независимым языком ассемблера для UNIX-систем. Это делает его основным средством написания эффективных и переносимых программ для этого класса вычислительных систем.

***5)Основы интерактивной работы в ОС UNIX.Начало и окончание сеанса.Доступ к оперативной документации.***

В операционной системе UNIX при наборе команд, имен файлов и каталогов, используется латинский алфавит, причем различаются заглавные и прописные буквы, т.е. команды и имена файлов должны вводится в том виде, в котором они приводятся в руководствах по UNIX [[5]](http://math.tsu.ru/cluster/other/cluster/literature.htm). Процесс общения с ОС UNIX осуществляется, в основном, при помощи клавиатуры и монитора. После входа в систему на экране можно увидеть одно из двух элементов интерфейса UNIX: командную строку или графическую оболочку.

Командная строка – традиционный способ общения в UNIX. Когда UNIX ожидает команду, на экране высвечивается приглашение – «$». Чтобы дать команду системе, нужно в командной строке набрать имя команды, а затем ее опции и аргументы (могут отсутствовать), разделенные пробелами. После набора ввод команды осуществляется нажатием клавиши ENTER. Команды UNIX имеют структуру

command option(s) argument(s)

Опции обычно сопровождаются знаком «минус» и модифицируют поведение команды. Аргументы – это обычно имена каталогов или файлов.

Заметим, что каталог пользователя находится в папке /home/user\_name/ корневого каталога системы. Названия файлов в ОС UNIX имеют структуру filename.ext, например, Euler.f; myCprog.c; a1.o; document.txt.

**НАЧАЛО СЕАНСА В СИСТЕМЕ UNIX**

**Пароль**

Когда вы впервые входили в систему, вам, наверное, пришлось ввести пароль, после того как вы вели свое входное имя. Однако системный администратор может и не установить пароль для вас, в этом случае приглашение password: не появится на экране.

Ваш пароль - средство безопасности; он защищает ваши файлы от доступа других пользователей. Если ваш системный администратор не установил пароль для вас, вы должны сделать это сами. Хорошо менять свой пароль примерно раз в месяц, даже если ваша система и не требует этого от вас.

Ниже приведен список правил, которым вы должны следовать, выбирая пароль. Пароль должен:

**-** иметь по меньшей мере шесть символов, но значение имеют только первые восемь.

**-** содержать по меньшей мере две буквы

**-** содержать по меньшей мере одну цифру или специальный знак (не используйте символы @ и #; они относятся к уничтожающим символам).

**-** отличаться от входных имен пользователей, отличаться от сдвинутых входных имен и не быть входным именем набранным в обратном порядке.

**-** новый пароль должен отличаться от старого по меньшей мере тремя символами.

Помните, что каждый может знать ваше входное имя, но только вы должны знать свой пароль. Не выбирайте пароль, о котором можно легко догадаться. Следует избегать использования в качестве пароля имен ваших супругов и детей. Также лучше иметь цифру где-нибудь в середине пароля, а не с краю.

Введите команду **passwd** для того чтобы изменить свой пароль или набрать новый.

**Конец сеанса**

Чтобы закончить сеанс, вы должны выйти из системы. Выйти можно двумя способами. Первый способ: нажать одновременно клавишу Control и букву **d**. Это называется **control d** и обозначается в справочнике **^d** или **<CTRL/d>**. После нажатия **^d** на терминале снова появится приглашение **login**:.

Альтернативный метод завершения сеанса - набрать слово **exit** и нажать <RET> после интерактивного символа $.

Если ваш терминал соединен посредством модема, вам может понадобится отключить телефон. Замечание: простое выключение терминала не заканчивает ваш сеанс в системе!

….

….

….

…..

***6)Команды ОС UNIX. Формат команд. Интерпретатор команд.***

**1.** **Простейшие действия ls** - выдать список файлов в текущем каталоге.

**cd [каталог]** -сменить текущий каталог. Если имя каталога не указывается, то текущим становится домашний каталог пользователя.

**cp <что\_копировать> <куда\_копировать>** -копировать файлы.

**mv <что\_перемещать> <куда\_перемещать>** -переместить или переименовать файл.

**ln -s <на\_что\_сделать\_ссылку> <имя\_ссылки>** -создать символическую ссылку.

**rm <файлы>** -удалить файлы.

**mkdir <каталог>** -создать новый каталог.

**rmdir <каталог>** -удалить пустой каталог.

**rm -r <файлы и/или каталоги> (рекурсивное удаление)** -удалить файлы или каталоги и их подкаталоги. ОСТОРОЖНЕЙ с этой командой, поскольку пока у Linux нет системы полного восстановления удаленных файлов (если вы не пользуетесь специальными программами для помещения удаленных файлов в специальный каталог, что-то вроде "корзинки" в ОС Windows).

**cat <имя\_файла>** -вывод содержимого файла на стандартный вывод (по умолчанию - на экран).

Можно записать вводимый на экран текст с помощью следующей последовательности действий:   
**cat > <имя\_файла>  
.   
.   
CTRL/d**

**more <имя\_файла>** -просмотр содержимого длинного текстового файла по страницам.

**less <имя\_файла>** -просмотр содержимого текстового файла с возможностью вернуться к предыдущим страницам. Нажмите q, когда захотите выйти из программы. "less" - аналог команды DOS "more", хотя очень часто "less" бывает более удобной чем "more".

**pico <имя\_файла>** -редактировать текстовый файл с помощью текстового редактора pico.

**lynx** -просмотр файла html или WWW ссылки с помощью текстового браузера Lynx.

**tar -zxvf <файл>** -распаковать архив tgz или tar.gz

**find <каталог>** -name имя\_файла -найти файл с именем "имя файла" и отобразить результат поиска на экране. Поиск начинается с каталога <каталог>; "имя\_файла" может содержать маску для поиска.

**pine** -хорошая текстово - ориентированная программа для чтения электронной почты.

**mc** -запустить программу управления файлами "Midnight Commander" (Выглядит как "Norton Commander", но по своим возможностям ближе к far).

**./Имя\_Программы** -запустить на исполнение исполняемый файл в текущем каталоге, если текущего каталога нет в списке каталогов, указанных в переменной окружения PATH.

**xterm (в X терминале)** -запустить простой терминал в графической оболочке X-windows. Для того чтобы выйти из него, наберите "exit".

**Стандартные команды и команды, дающие информацию по системе команд (всегда набираются в одной строке). Описание:**

**pwd** -вывести имя текущего каталога.

**whoami** -вывести имя под которым Вы зарегистрированы.

**date** -вывести дату и время.

**time <имя программы>** -выполнить программу и получить информацию о времени, нужном для ее выполнения. Не путайте эту команду с date. Например: Я могу определить выполнить команду ls и узнать, как много времени требуется для вывода списка файлов в каталоге, набрав последовательность: time ls

**who** -определить кто из пользователей работает на машине.

**rwho -a** -определение всех пользователей, подключившихся к вашей сети. Для выполнения этой команды требуется, чтобы был запущен процесс rwho.

**ruptime** -какие машины работают в сети, и какие остановлены.

**finger <имя\_пользователя>** -системная информация о зарегистрированном пользователе. Попробуйте: finger <ваш login-name>

**uptime** -количество времени, прошедшего с последней перезагрузки операционной системы.

**ps a** -вывести список текущих процессов в Вашем сеансе работы.

**top** -интерактивный список текущих процессов, отсортированных по использованию центрального процессора.

**uname -a** -вывести информацию о версии операционной системы.

**free** -вывести информацию по использованию памяти.

**df -h** -вывести информацию о свободном и используемом месте на дисках.

**du . -bh | more** -вывод на экран информации о размере файлов и каталогов, начиная с текущего каталога.

**set|more** -вывести текущие значения переменных окружения. (Не для всех shell. Для csh/tcsh - printenv | more, хотя set тоже покажет полезную информацию.)

**echo $PATH** -вывести значение переменной окружения "PATH" Команда echo может использоваться для вывода значений любых переменных окружения. Воспользуйтесь командами set или printenv для получения полного списка.

**Работа с сетью**

**ssh** -обеспечивает безопасное вхождение в удаленный сеанс работы с другой машиной, а также позволяет выполнить заданную команду на удаленной машине без вхождения в сеанс работы:   
**ssh [-l Ваше\_имя\_пользователя\_на\_удаленной\_машине] <имя\_удаленной\_машины>** -вхождение в сеанс на удаленной машине. Используйте имя машины или ее IP адрес. (Вы должны быть зарегистрированы на этой удаленной машине).   
(если Ваше\_имя\_пользователя одинаково на локальной и удаленной машинах, то его можно не набирать, т.е.:   
**ssh <имя\_удаленной\_машины>** -позволит Вам войти в сеанс на удаленной машине)   
**ssh <Ваше\_имя\_пользователя\_на удаленной машине@><имя\_удаленной\_машины> <команда>** -выполнит на удаленном компьютере заданную команду и передаст Вам на экран результат ее выполнения (При установлении соединения по ssh Вы должны будете ввести пароль, который Вы имеете на удаленном компьютере; при этом пароль будет передан по сети в зашифрованном виде, т.е. безопасным образом.)

**scp** -обеспечивает безопасное копирование файлов в сети:   
**scp <имя\_файла\_на\_локальном\_компьютере> <Ваше\_имя\_пользователя\_на удаленной машине>@<имя\_удаленной\_машины>:** -cкопирует файл с локального компьютера в Вашу корневую директорию на удаленном компьютере (наличие ":" в конце команды обязательно).

**telnet <имя\_удаленной\_машины>** -связаться по telnet с другой машиной. Войдите в сеанс работы после установления связи с помощью вашего пароля.

**ftp <имя\_удаленной\_машины>** -связаться по ftp с удаленным компьютером. Этот тип связи хорош для копирования файлов с/на удаленную машину.

Предпочтительно не пользоваться командами **telnet** и **ftp**, а использовать только ssh и scp, поскольку они обеспечивают безопасность сетевого соединения!!!

**hostname -i** -показывает IP адрес компьютера, на котором Вы работаете.

**Некоторые команды администрирования**

**alias ls="ls -Fskb --color"** -создать alias - псевдоним для того чтобы одной командой можно было запустить более сложную комбинацию команд. Поместите создание alias в файл /etc/bashrc если вы хотите, чтобы эти псевдонимы были доступны всем пользователям вашей системы.

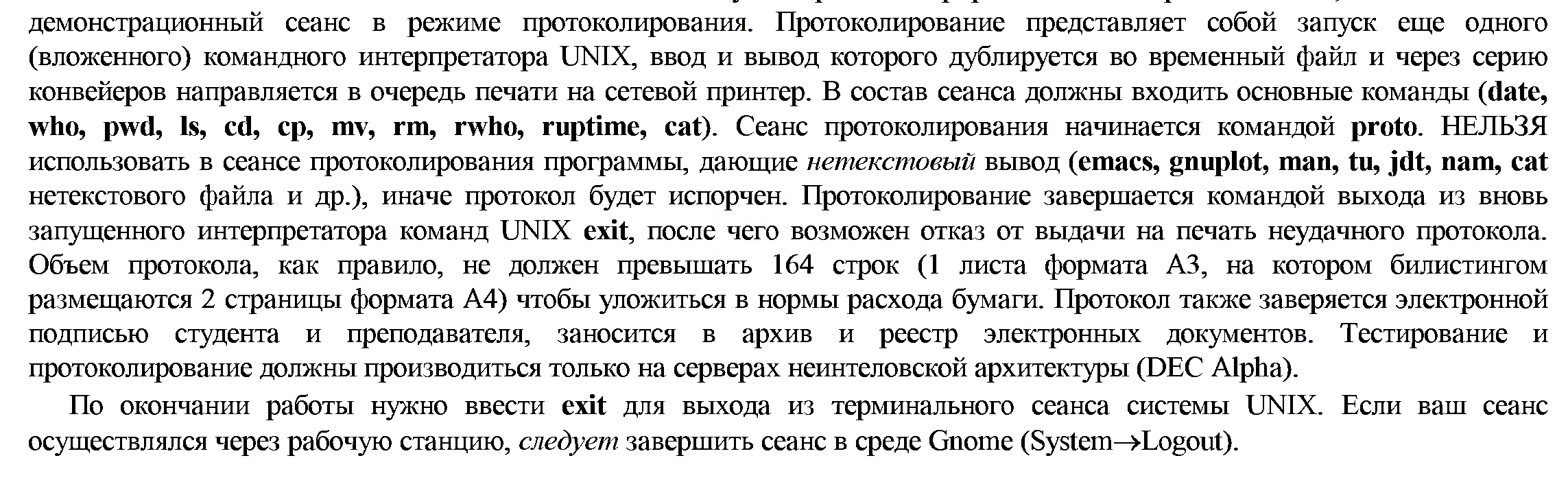
**Для tcsh** -формат определения alias другой:

**alias la 'ls -AF --color=none'**

**kapasswd** -команда для изменения пароля для доступа к файловой системе AFS. При работе на базовом Linux-кластере ЛИТ следует пользоваться только этой командой (а не командой passwd!) для изменения пароля для вхождения в кластер.

**passwd** -изменить свой пароль на каком-либо локальном компьютере.

***7)Протоколирование команд.***

 ***8)Файловая система ОС UNIX (структура, наименование, навигация)***

Файловая система в Unix организована в виде единого дерева директорий. Никаких букв дисков (a:, c:) здесь нет, а если используется несколько физических дисков (например, первый жесткий диск, второй, CD-ROM и т.д.) то они представляются отдельными ветвями на общем дереве.

Директории разделяются косой чертой "/" (а не обратной, как в Dos), например, "/home/inpunix/ivanov/" -- поддиректория "ivanov" в директории "inpunix" директории "home". Самая верхняя директория ("корневая") называется "/".

Какие разрешаются имена:

* Имя файла (и директории) может содержать практически любые символы, включая пробелы (и даже управляющие символы, типа перевода строки, если удастся их ввести). Примеры:

list.txt  
A file with spaces.txt  
Сводный отчет по трудозатратам за 1998й год.tex  
Manual [part 1 of 3].txt

* Символ "." не является особенным, и в имени может быть любое их количество, а может и не быть вовсе. Так, к примеру,

file.with.many.dots  
short  
file\_without\_extension  
.......

являются вполне нормальными именами.

* У программ не требуется какое-либо специальное расширение, типа .exe -- достаточно установить атрибут "x" (см. ниже).
* Заглавные и маленькие буквы различаются; так,

report.txt  
REPORT.TXT  
Report.txt  
RePorT.tXt

это четыре разных имени.

* В отличие от Dos/Windows, нет зарезервированных имен, таких, как con, com1 и т.д. Вместо них есть сответствующие файлы устройств в директории /dev:

|  |  |  |
| --- | --- | --- |
| **Файл в Dos** | **Устройство в Unix** | **Назначение** |
| con | /dev/tty | Консоль |
| nul | /dev/null | Пустота |
| com1 | /dev/ttyS0 | Первый последовательный порт |
| lpt1 | /dev/lp0 | Первый параллельный порт |

* У программ не требуется какое-либо специальное расширение, типа .exe -- достаточно установить атрибут "x" (см. ниже).
* Длина имени может быть до 255 символов; полный путь (включая директории) -- 1024 символа.

Хотя единственный символ, которого не может быть в имени, это "/" (поскольку это разделитель директорий), лучше не слишком увлекаться, и стараться не делать имен с пробелами, и не злоупотреблять различием в регистре букв -- это только внесет лишнюю путаницу. Также лучше не называть файлы по-русски.

В каждой директории есть поддиректория ".", являющаяся ссылкой на саму директорию и "..", являющаяся ссылками директорию уровнем выше.

Файлы, имена которых начинаются c точки (включая . и ..), считаются "неинтересными", и в листинге директории обычно не показываются.

***9)Метасимволы для генерации имен файлов.***

Метасимволы - символы, имеющие специальное значение для интерпретатора :

? \* ; & ( ) | ^ < > <пробел> <табуляция> <возврат\_каретки>

Однако каждый из этих символов может представлять самого себя, если перед ним стоит \. Все символы, заключенные между кавычками ' и ', представляют самих себя. Между двойными кавычками (") выполняются подстановки команд (см п. 2.2) и параметров (см. п. 2.3), а символы \, `," и $ могут экранироваться предшествующим символом \.

После всех подстановок в каждом слове команды ищутся символы \*,?, и [. Если находится хотя бы один из них, то это слово рассматривается как шаблон имен файлов и заменяется именами файлов, удовлетворяющих данному шаблону (в алфавитном порядке). Если ни одно имя файла не удовлетворяет шаблону, то он остается неизменным. Значения указанных символов:

\* - произвольная (возможно пустая) последовательность

символов;

? - один произвольный символ;

[...] - любой из символов, указанных в скобках перечислением

и/или с указанием диапазона;

cat f\* - выдаст все файлы каталога, начинающиеся с "f";

cat \*f\* - выдаст все файлы, содержащие "f";

cat program.? - выдаст файлы данного каталога с

однобуквенными расширениями, скажем "program.c" и "program.o", но

не выдаст "program.com";

cat [a-d]\* - выдаст файлы, которые начинаются с "a", "b",

"c", "d". Аналогичный эффект дадут и команды "cat [abcd]\*" и

"cat [bdac]\*".

***10)Управление файлами(создание, копирование, переименование, сцепление, удаление)***

|  |  |
| --- | --- |
| urbbul1d | file filename(s) - отображает тип файла (например ASCII текст, документ Word и др.) |
| urbbul1d | cat filename(s) - показывает содержание файлов (используется только для текстовых файлов!); |
| urbbul1d | more filename(s) - действует также, как и cat, но позволяет листовать страницы; |
| urbbul1d | head filename - показывает первые десять строк файла filename; |
| urbbul1d | tail filename - показывает последние десять строк файла filename; |
| urbbul1d | wc filename(s) - для каждого текстового файла показывает число байт, букв и линий; |
| urbbul1d | rm filename(s) - уничтожает файлы; |
| urbbul1d | cp filename newname - создает копии файлов с новыми именами; |
| urbbul1d | cp filename(s) dir копирует один или более файлов в другой каталог; |
| urbbul1d | mv oldname newname - изменяет имя файла или каталога; |
| urbbul1d | mv filename(s) dir - перемещает один или более файлов в другой каталог; |
| urbbul1d | find dir –name filename - пытается локализовать файл filename рекурсивно в подкаталоге dir; |

***11)Управление каталогами.***

|  |  |
| --- | --- |
| urbbul1d | pwd – показывает текущий рабочий каталог; |
| urbbul1d | cd dir – установит текущим рабочим каталогом катало с именем dir; |
| urbbul1d | cd – вернуться в каталог /home/user\_name/; |
| urbbul1d | mkdir subdir – создать новый подкаталог с именем subdir; |
| urbbul1d | rmdir subdir – переместить подкаталог с именем subdir, который должен быть пустым; |
| urbbul1d | ls – показать содержание текущего каталога; |
| urbbul1d | ls dir – показать содержание каталога dir; |
| urbbul1d | ls –A - показать все файлы, в том числе и скрытые; |
| urbbul1d | ls –l - показать аттрибуты (владельца, разрешение на доступ, размер файла и время последней модификации); |
| urbbul1d | mv oldname newname - изменить имя подкаталога или переместить его; |
| urbbul1d | cp dirname destination - копировать весь подкаталог dirname в другое место destination |

***12)Переадресация ввода и вывода в ОС UNIX.***

### Перенаправление ввода и вывода

В системе UNIX некоторые команды ожидают ввод только с клавиатуры (стандартный ввод) и большинство команд отображают свой вывод на экране терминала (стандартный вывод). Однако система UNIX позволяет вам перенаправлять ввод и вывод в файлы и программы, т.е. вы можете сказать shell:

* взять ввод из файла, а не с клавиатуры;
* послать вывод в файл, а не на терминал;
* использовать программу как исходные данные для другой программы.

#### 8.3.1. Перенаправить ввод: знак <

Чтобы перенаправить ввод, укажите в командной строке после знака "меньше чем" (<) имя файла:

command < имя\_файла<CR>

#### 8.3.2. Перенаправить вывод: знак >

Чтобы перенаправить вывод, укажите в командной строке после знака "больше чем" (>) имя файла:

command > имя\_файла<CR>

Примечание. Если вы перенаправите вывод в уже существующий файл, то вывод вашей команды заменит содержимое существующего файла.

Перед тем, как перенаправить вывод команды в конкретный файл убедитесь, что этот файл не существует. shell не предупреждает, что выполняет перезапись существующего файла.

Чтобы убедиться, что файл с запланированным именем не существует, воспользуйтесь командой ls с аргументом "имя\_файла". Если этот файл не существует, то ls выдаст сообщение, что файл не найден в текущем справочнике. Например, проверка существования файлов temp и junk даст следующий результат:

$ ls temp<CR>

temp

$ ls junk<CR>

junk: no such file or directiry

$

Это означает, что вы можете назвать свой файл junk, но не можете использовать в качестве имени temp, если не хотите потерять содержимое существующего файла.

#### 8.3.3. Добавить вывод в существующий файл

Чтобы добавить вывод в существующий файл и не разрушить его, вы можете воспользоваться символом >>:

command >> имя\_файла<CR>

В результате выполнения команды вывод будет добавлен в конец существующего файла. Если файл не существует, то он будет создан.

***13)Конвейеры команд***

Конвейер. Взять стандартный вывод одной программы и использовать как стандартный ввод другой. Это одна из самых богатых возможностей ОС UNIX; ее можно использовать для создания собственных команд. Предположим, Вам нужен список всех файлов, принадлежащих **edk**. Эту операцию нельзя выполнить с помощью ключей команды **ls**, но, используя конвейер, можно объединить команды **ls -l** и **grep**:

% ls -ls | grep "edk"

Команда ls -l выдает список всех файлов, включая информацию о владельцах, а grep извлекает все пункты, содержащие строку **edk**.

*Конвейеры* — это возможность нескольких программ работать совместно, когда выход одной программы непосредственно идет на вход другой без использования промежуточных временных файлов. Синтаксис:

команда1 | команда2

Выполняет команду1 используя её поток вывода как поток ввода при выполнении команды2, что равносильно использованию двух перенаправлений и временного файла:

команда1 > ВременныйФайл

команда2 < ВременныйФайл

rm ВременныйФайл

***14)Фоновые процессы: запуск, остановка, слежение. Приостановка и возобновление emacs.***

***……***

***15)Понятие о программировании на ИКЯ.***

***…….***

Редактирование текстов

***1)Понятие о редакторе текстов. Основные типы редакторов тектов.***

Для работы с текстами на компьютере используются программные средства, называемые текстовыми редакторами. Существует большое количество разнообразных текстовых редакторов, различающихся по своим возможностям  
Основное назначение текстовых редакторов – создавать текстовые файлы, редактировать тексты, просматривать их на экране, изменять формат текстового документа, распечатывать его на принтере.  
MS-DOS Editor и Блокнот предназначены для обработки несложных текстовых файлов   
Современные программы предусматривают множество дополнительных функций, позволяющих готовить текстовую часть документа на типографском уровне:

* форматирование символов и абзацев,
* оформление страниц,
* построение оглавлений и указателей,
* создание таблиц, проверка правописания и т. д.

Кроме того, современные программы позволяют включать в текст графические объекты: рисунки, диаграммы, фотографии.

**Современный текстовый процессор Microsoft Word** предназначен для создания, просмотра, модификации и печати текстовых документов, предусматривает выполнение операций над текстовой и графической информацией. С помощью Word можно быстро и с высоким качеством подготовить любой документ — от простой записки до оригинал-макета сложного издания.

Типы редакторов текстов:

* строчный процессор
* экранный процессор
* графический процессор
* пакетный процессор
* текстовый процессор

***2)Строчное редактирование. Концепция. Редактор Ed.***