

МУЛЬТИЗАДАЧНЫЕ ОПЕРАЦИОННЫЕ СИСТЕМЫ

Ознакомление с базисными понятиями командного интерфейса пользователя ОС UNIX и командами для работы с файловой системой.

Цели лабораторной работы

Первой целью данной работы является ознакомление с основными понятиями командного интерфейса пользователя ОС UNIX. Практическая цель – выполнение индивидуальных заданий в рамках сеанса работы в среде ОС UNIX, суть которых состоит в выполнении процедур начальной установки, входа в систему, проверки программного и системного окружения, освоении приемов переназначения ввода/вывода и конвейеризации, использовании утилит ОС UNIX.

Второй целью данной работы является ознакомление с возможностями, предоставляемыми командным интерфейсом пользователя ОС UNIX, для работы с файловой системой. Практическая цель – выполнение индивидуальных заданий в рамках сеанса работы в среде ОС UNIX, суть которых состоит в выполнении процедур копирования, перемещения, переименования и удаления файлов, создания, переименования и удаления директорий, использование процедур связи пользователей, а также использовании периферийных устройств.

1. ОСНОВНЫЕ ПОНЯТИЯ ОПЕРАЦИОННОЙ СИСТЕМЫ UNIX

Командный интерфейс пользователя. Основные сведения

Для всех пользователей, вот уже скоро сорок пять лет сталкивающихся с ОС UNIX, основной проблемой является отсутствие удобного и естественного, т.н. дружественного интерфейса пользователя. В семействах ОС типа UNIX в связи с чрезвычайно широким разнообразием видеотерминального оборудования и по историческим причинам господствует идеология строчного диалога с пользователем. При этом основной принцип построения пользовательского интерфейса – «в ОС типа UNIX: если что-то можно, то это можно везде и в любом количестве». Подобный подход означает достаточность возможностей даже простейших алфавитно-цифровых видеотерминалов для ввода любых UNIX-команд.

Итак, для ввода любой команды ОС UNIX правила командного интерфейса пользователя предоставляют следующий строчный формат:

```
$ <команда> [<ключи_вызова (модификаторы)>] [<параметры_вызова>] <ВВ>
```

, где

- \$ - системное приглашение или «подсказка» (prompt), выводимая в качестве признака готовности ОС UNIX воспринимать ввод команды пользователем;
- <команда> - любая из команд UNIX, либо утилита ОС;
- <ключи-вызова (модификаторы)> - последовательность однобуквенных символов — кодов режимов исполнения команды, префиксируемая символом "-" либо "--", либо "+" (ключи еще иногда называют флагами);
- <параметры_вызова> - аргументы команды;
- [...] - необязательное присутствие значения, заключенного между [и];
- <...> - нажатие клавиши терминала, обозначаемой "...";

- <BB> - нажатие клавиши "ENTER" ("RETURN", "CR/LF", "ввод" в зависимости от вида терминала).

Признаком завершения реакции системы на введенную команду является появление на экране видеотерминала пользователя системного приглашения.

Для входа в режим взаимодействия с ОС типа UNIX и получения доступа к командному интерфейсу пользователь один раз за сеанс работы должен пройти процедуру регистрации, т.е. сообщения системе своих индивидуальных, присвоенных системным администратором, параметров входа: login – имени и, может быть, пароля (при вводе на экране видеотерминала не отображается). Процедура регистрации может быть проиллюстрирована следующим фрагментом:

```
login: user1 <BB>
password: <....><BB>
$
```

где:

- login, password и \$ – реакция системы;
- user1 – пример login-имени пользователя;
- <....> – некоторая последовательность нажимаемых клавиш консоли, составляющая пароль.

В конце сеанса пользователю для правомочного завершения работы с системой необходимо отсоединиться от ОС с помощью команд logout, exit, halt либо reboot (если установлены права использования).

После регистрации в системе пользователь получает в распоряжение на время сеанса часть файловой системы ОС типа UNIX с собственной «домашней» директорией и копией командной оболочки. Командный интерфейс пользователя подразумевает для каждой команды наличие трех стандартных файлов:

- stdin – файла ввода информации,
- stdout – файла вывода информации,
- stderr – файла вывода сообщений об ошибках (файл диагностики).

По умолчанию в ОС типа UNIX для процесса исполнения UNIX-команды пользователя stdin по умолчанию назначен на клавиатуру, а stdout и stderr – по умолчанию на экран видеотерминала пользователя.

Для переназначения стандартных файлов ввода либо вывода существуют в ОС типа UNIX процедуры переадресации:

- > – переадресация вывода (запись в указанный файл)
- < – переадресация ввода (чтение из указанного файла)
- >> – переадресация вывода (добавление в конец указанного файла).

Кроме переадресации связь между процессами исполнения двух команд (UNIX-процессами в общем смысле) может быть осуществлена в виде т.н. конвейера – подачи данных с выхода одного UNIX-процесса на вход другого с помощью процедуры программного канала. Символом для обозначения программного канала является вертикальная черта |. Так например, в командной строке, вида

```
$ <команда-1>|<команда-2>
```

здесь выход процесса (а именно stdout), определяемый командой <команда-1>, является входом (stdin) процесса, определяемого командой <команда-2>.

Командный интерфейс пользователя, кроме того, предоставляет возможность UNIX-пользователю с помощью команд самого командного интерфейса изучать свойства команд этого интерфейса и настраивать, изменять, модифицировать комплекс программно-технических средств реализации этого интерфейса. Наиболее простым средством

информационного обеспечения командного интерфейса является команда `man`, которой в качестве параметра вызова передается имя команды. Не смущайтесь тем, что сообщения команды `man` не на русском языке – это язык, хотя и напоминающий английский, однако максимально адаптированный к международному жаргону программистов и потому должен быть понятен Вам.

Для настройки терминала служит команда `stty`. В стандартном, наиболее часто встречающемся варианте, пользователь для реализации интерфейса имеет в распоряжении виртуальный терминал, эмулирующий протокол общения с терминалом VT52/VT100, где для редактирования командной строки используются следующие последовательности:

- `<ctrl>+<c>` – принудительное завершение процесса (более точно, процессу посылается сигнал SIGINT, который по умолчанию приводит к завершению процесса, подробнее см. в лабораторной работе №4);
- `<ctrl>+<d>` – завершает текстовый ввод для программ;
- `<ctrl>+<h>` – для удаления последнего введенного символа в командной строке;
- `<ctrl>+<h>` – для удаления последнего введенного слова в командной строке;
- `<ctrl>+<z>` – приостанавливает текущий процесс и возвращает управление интерпретатору команд;
- `<@>` – удаление (смысловое) набранной, но не введенной строки командного интерфейса («сброс команды»);
- `` – завершение зависшего процесса и восстановление системного приглашения. CTRL-Z

Для адаптации скорости выводимой процессами на экран видеотерминала информации служат последовательности:

- `<ctrl>+<S>` – приостановка вывода;
- `<ctrl>+<Q>` – возобновление вывода.

2. ОСНОВНЫЕ СВЕДЕНИЯ О ПРОЦЕДУРАХ РАБОТЫ С ФАЙЛОВОЙ СТРУКТУРОЙ

В ОС UNIX единицей хранения является файл, причем доктрина, принятая отцами-основателями (Томпсон, Ритчи и примкнувший к ним Керниган), подразумевает, что объектом хранения в файле системы UNIX могут быть тексты, команды, а также каталоги и содержимое спецфайлов. Массив памяти, отведенный системой UNIX под файловое пространство, носит структурированный иерархический характер, где все файлы сгруппированы в директории (каталоги), а директории рекурсивно входят в корневую директорию. Корневая директория не является при этом ни чьим подкаталогом. Каждый пользователь имеет свою собственную точку входа в файловую систему – «домашнюю директорию». Именованье файлов (помните, что с точки зрения ОС UNIX директория – это тоже некоторый файл) определяется именованьем пути от какого-либо однозначно определенного в текущий момент каталога, т.е. перечислением в порядке спуска по иерархии файловой системы имен каталогов, где в качестве разделителя используется символ slash – «/». Корневая директория ОС UNIX – уникальна, существует всегда и однозначно определена на одно и то же место файловой системы для всех пользователей – определяется единственным символом slash. Текущая или рабочая (вспомните, что такое `pwd` для обозначения команды) директория пользователя – однозначно определена для каждого пользователя и может изменяться пользователем во время сеанса (независимо от других сеансов). Текущая директория для краткости именуется символом «точка» – независимо от реального имени, а директория, для которой текущая директория пользователя является подкаталогом (если таковая существует), именуется удвоенным символом «точка» – «..». Домашняя директория пользователя обозначается символом «~».

Командный интерфейс ОС UNIX позволяет пользователю выполнять следующие

действия в файловой системе:

- просмотр содержимого каталога `ls`;
- переход в заданный каталог `cd`;
- вывод полного имени текущего каталога `pwd`;
- определение типа файла по содержимому `file`;
- вывод содержимого файла в `stdout` (по умолчанию, на экран) `cat`;
- переименование или перемещение файла `mv`;
- копирование простого файла `cp`;
- удаление простого файла `rm`;
- создание нового каталога `mkdir`;
- удаление каталога `rmdir`;
- изменение атрибутов защиты файла `chmod`;
- выдача метаданных о файле, содержащейся в файловой системе `stat`;
- сортировку или слияние файлов `sort`;
- поиск в файле по шаблону `grep`;
- построчное сравнение файлов `diff` и `comm`.

ОС UNIX позволяет пользователю, если он имеет соответствующие права доступа, получить доступ через рабочую директорию к файлу из любой директории не путем копирования его в рабочий каталог, а путем занесения только ссылки на этот файл в свою рабочую директорию путем исполнения команды связывания («линкования») `ln`.

Принципиальной особенностью ОС UNIX является то, что все файлы находятся в едином дереве каталогов (в отличие, например, от ОС Windows, где каждому диску соответствует своё дерево каталогов). Работа данного дерева поддерживается благодаря механизму VFS – виртуальной файловой системы. Файловые системы физических носителей (например, жёстких дисков) могут быть подключены к дереву каталогов при помощи процедуры монтирования. Монтирование производится командой `mount` (от имени суперпользователя). Ввод команды `mount` без параметров позволяет получить список всех смонтированных файловых систем.

Идеология единого структурированного иерархического и допускающего множественность ссылок файлового пространства как файловой структуры позволяет в среде ОС UNIX относительно легко решать как проблему использования периферийных устройств, так и проблему информационного обмена между пользователями. Для выполнения, например, вывода файла (записи) с использованием периферийного устройства, определяемого спецфайлом `/dev/dev1` необходимо переназначить стандартный файл вывода на этот спецфайл. Для связи с пользователями можно воспользоваться дополнительно возможностями командного интерфейса ОС UNIX, который позволяет:

- копировать строки («посылать сообщение») с терминала одного пользователя на терминал другого пользователя непосредственно внутри одной файловой системы – `write`;
- управлять использованием с других терминалов своего терминала в качестве устройства - назначения («мишени») команды `write - mesg`.

Для построения более сложных коммуникационных процессов, например, для связи пользователей в разных файловых системах ОС UNIX или вообще в разных ОС служит система электронной почты (E-Mail). Для работы с реальными сетями электронной почты служит стандартизованная в ОС UNIX на уровне применения утилиты `mail` или `elm`, работа с которыми строится по принципу экранного меню и потому достаточно проста и не требует высокой квалификации. Для использования утилит электронной почты пользователь должен лишь иметь доступ к какому-либо серверу E-Mail и быть зарегистрированным в нем в качестве абонента.

ЗАДАНИЕ К ЛАБОРАТОРНОЙ РАБОТЕ

Домашняя подготовка

1. Изучить по прилагаемому списку литературы следующие вопросы и темы:

- системные ресурсы пользователя;
- вход в систему, выход из системы;
- рабочая директория, «домашняя» директория, текущая директория;
- терминал пользователя и его параметры.

2. Изучить по прилагаемой литературе функциональные особенности следующих команд командного интерфейса: login, logout (exit), who, pwd, date, time, cd, ls, stat, file, cat, more (less), ln, mkdir, rmdir, mknod, touch, mv, rm, ps, top, man, stty.

3. Подготовить ответы по вопросам коллоквиума.

Вопросы коллоквиума

- 1) Что такое многопользовательская операционная система?
- 2) Что такое мультизадачная операционная система?
- 3) Что такое мобильная операционная система?
- 4) Что такое ядро ОС типа UNIX?
- 5) Каким образом ОС типа UNIX различает пользователей системы?
- 6) Когда заканчивается регистрация пользователя в системе?
- 7) Кто устанавливает идентификатор и пароль пользователя в системе?
- 8) Есть ли различие между понятиями «идентификатор пользователя» и «имя терминала пользователя»?
- 9) Есть ли различие между понятиями «рабочая директория» и «домашняя директория» для пользователя?
- 10) Что такое «процесс» в ОС типа UNIX?
- 11) Какие команды могут информировать пользователя о процессах этого пользователя?
- 12) Информация, записываемая некоторым процессом в stdout составляет 25 строк текста. Какими средствами, Вам уже известными, можно прочесть эту информацию пользователю?
- 13) Каким образом (уже известным Вам) можно создать файл нулевой длины?
- 14) Каким образом можно сохранить информацию о времени входа в систему в виде файла?
- 15) Каким образом можно сохранить информацию об имени терминала пользователя в виде файла?
- 16) Каким образом (возможно, несколькими способами) можно в одном файле сохранить информацию о дате входа в систему, текущем пути к директории пользователя и дате, непосредственно предшествующей выходу из системы?
- 17) Каким образом узнать соответствие идентификаторов активных пользователей и имен терминалов?
- 18) Что такое блочориентированный специальный файл?
- 19) Что такое байториентированный специальный файл?
- 20) Как определить наличие в директории обычных файлов?
- 21) Как определить наличие в директории блочориентированных специальных файлов?
- 22) Как определить наличие в директории байториентированных специальных файлов?
- 23) Как определить наличие в директории поддиректории?
- 24) Что такое группа пользователей в системе UNIX?
- 25) Для чего служат специальные файлы в ОС типа UNIX?

26) Как определить, является ли какой либо файл в директории пользователя ОС типа UNIX исполняемым непосредственно?

27) С помощью какой из команд ОС типа UNIX можно определить наличие среди файлов пользователя stdin, stdout и stderr?

28) Какие категории пользователей в ОС типа UNIX Вы можете назвать?

29) Могут ли быть в ОС типа UNIX два и более пользователя с одинаковыми именами?

30) Могут ли существовать одновременно в ОС типа UNIX два и более процесса с одинаковыми именами пользователя и разными идентификаторами процесса?

31) Могут ли существовать одновременно в ОС типа UNIX два и более процесса с одинаковыми идентификаторами процесса и разными именами пользователя?

32) Из скольких полей должно состоять имя файла в ОС типа UNIX?

33) Из скольких полей может состоять имя файла в ОС типа UNIX?

34) К чему может привести применение команды cat к исполняемому файлу?

4. Просмотрите лабораторное задание, соответствующее Вашему варианту и обдумайте его выполнение.

Лабораторное задание

1) Перед выполнением лабораторного задания:

- ответьте на вопросы коллоквиума, заданные преподавателем;
- получите у администратора системы (преподавателя) регистрационные имя пользователя и пароль, а также право на использование терминала.

2) Зарегистрируйтесь в системе с терминала.

3) Определите следующие параметры среды:

- текущее время;
- имя Вашего терминала;
- имя Вашей рабочей директории, имя домашней директории Вашего пользователя;
- имена и соответствующие им терминалы всех активных пользователей;
- номера всех доступных для Вашего наблюдения активных процессов;
- параметры Вашего терминала.

4) Создайте пустой файл (файл, не содержащий никакой информации) в Вашей директории по имени first.snap;

5) В соответствии с номером в журнале учебной группы выполните пункт задания (см. таблицу 1).

Таблица 1

№ варианта	Содержание пункта задания
1	Создайте в Вашей домашней директории файл <code>contens.of.partent</code> и запишите в него имя и листинг каталога, подкаталогом которого является Ваша домашняя директория
2	Создайте в Вашей домашней директории файл <code>contens.of.par.sort</code> содержащий отсортированный по времени создания и кроме того еще отсортированный по времени последнего доступа листинг каталога, подкаталогом которого является Ваша домашняя директория
3	Создайте в Вашей директории файл <code>time.my</code> , содержащий текущее значение даты и времени, причем во второй строке файла зафиксируйте десятичный номер текущей недели в году, в третьей строке - десятичный номер года вместе с номером столетия, в четвертой - информация о часовом поясе
4	Создайте в Вашей директории файл <code>time.our</code> , содержащий кроме текущего значения даты и времени национальное представление полного названия дня недели, его десятичного номера и полного названия месяца.
5	Создайте в Вашей директории файл <code>users.of.session</code> имена и номера терминалов всех активных пользователей, текущее значение системного времени и имя Вашего терминала
6	Создайте в Вашей директории файл <code>process.of.session</code> содержащий список всех выполняющихся процессов, текущее значение системного времени и имя Вашего терминала

7	Создайте в Вашей директории файл <code>my.own</code> , содержащий ее имя, содержимое и имя Вашего терминала вместе с регистрационным именем
8	Создайте в Вашей директории файл <code>partent.of.my</code> , содержащий листинг каталога, подкаталогом которого является Ваша директория, а также информацию о текущем значении системного времени.
9	Создайте в Вашей директории файл <code>all.of.my.par</code> , содержащий рекурсивный вывод содержимого подкаталогов каталога, подкаталогом которого является Ваша директория
10	Создайте в Вашей директории файл <code>users.and.time</code> , создающий имена и номера терминалов всех активных пользователей, имя Вашей директории и имя каталога, подкаталогом которой является Ваша директория
11	Создайте в Вашей директории файл <code>cleven.example</code> , содержащий список всех выполняющихся процессов, имя Вашей директории и имя каталога, подкаталогом которой является Ваша директория
12	Создайте в Вашей директории файл <code>zwoelf.exam</code> , содержащий ее листинг, информацию о собственно директории а также ее листинг с указанием размера файлов директории в блоках
13	Создайте в Вашей директории файл <code>learn.no1</code> , содержащий время начала получения листинга текущей директории, сам листинг и время окончания выдачи листинга текущей директории (времена – с точностью до ввода кода командной строки).
14	Создайте в Вашей директории файл <code>example.sample</code> , содержащий время начала получения информации обо всех активных пользователях в системе, саму эту информацию и время окончания выдачи этой информации (время - приблизительно)
15	Создайте в Вашей директории файл <code>process.es</code> , содержащий (приблизительно) время начала получения информации о всех активных процессах в системе, саму эту информацию и (приблизительно) время окончания выдачи этой информации
16	Создайте в Вашей директории файл <code>column.partent.ls</code> , содержащий листинг каталога, подкаталогом которого является Ваша директория, сформированный в несколько колонок, а также имя этого каталога.

6) Убедитесь в том, что Ваша директория не пуста. Создайте файл `second.type`, содержащий информацию о типах (по содержимому) всех файлов Вашей директории, включая и файл `second.type`;

7) Создайте файл `second.rights`, содержащий информацию о правах доступа и владельцах файлов в Вашей директории;

8) Получите информацию о файле `second.type`, которая содержится в файловой системе;

9) Вызовите команду `man` для просмотра информации о командах:

`login, logout, exit, ls, pwd, who, date (time), ps, cd, file, cat, more, man, stty`.
Отметьте в создаваемом Вами файле `third.exeption` имена команд, по которым отсутствует информация при обращении к команде `man`;

8) Повторите пункт 3 лабораторного задания, сохраняя всю необходимую информацию в файле `first.snap` и используя в качестве разделителя данных о действующих параметрах последовательность символов типа `"-----...-----"` (или иных, по Вашему желанию).
Отметьте совпадения и несовпадения при выполнении команд пункта 3 и содержимого файла `first.snap`;

9) Попробуйте добиться того, чтобы вывод на Ваш терминал осуществлялся темными символами на светлом фоне;

10) Создайте в Вашей домашней директории поддиректорию `Mysecondir`, скопируйте Ваш файл `second.type` в директорию `Mysecondir`. Переместите Ваш файл `first.snap` в директорию `Mysecondir` с переименованием его в `second.snap`;

13). Установите для файла `second.snap` права доступа с абсолютным значением атрибута `777`;

14) Установите для директории права доступа внешнего пользователя такие, чтобы он мог копировать из Вашей директории;

15) Создайте из `Mysecondir` ссылку на файл, владельцем которого является другой пользователь. Что Вам мешает?

16). Создайте в поддиректории `Mysecondir` в Вашей домашней директории ссылку на

директорию, владельцем которой является другой пользователь. Что Вам мешает?

17) Выполните пункт задания в соответствии с Вашим номером в журнале учебной группы (см. таблицу 2)

Таблица 2

№ варианта	Содержание пункта задания
1	Выполните ещё раз пункт 4 задания, сохранив результат в файл second.snap. Выведите строки, отсутствующие в second.snap, но присутствующие в файле first.snap, полученном в пункте 4 задания. Результат поместить в файл differ в Mysecondir.
2	Выполните ещё раз пункт 4 задания, сохранив результат в файл second.snap. Выведите строки, присутствующие только в одном из файлов first.snap и second.snap, в файл differ в Mysecondir.
3	Выполните ещё раз пункт 4 задания, сохранив результат в файл second.snap. Выполните сравнение на совпадение с файлом first.snap, полученном в пункте 4 задания с выводом общих строк в файл common в Mysecondir.
4	Выполните два раза вариант 6 из таблицы 1 задания к лабораторной работе. Результаты сохраните в разные файлы. Выполните построчное сравнение этих файлов. Результат сохраните в файл compare Mysecondir.
5	Подсчитайте количество строк, содержащих имя Вашего терминала, которые содержатся в файлах, расположенных в Mysecondir. Сохраните эту информацию в файл compare Вашей домашней директории (попробуйте все это выполнить в одной командной строке).
6	Выполните два раза вариант 6 из таблицы 1 задания к лабораторной работе. Результаты сохраните в разные файлы. Подсчитайте количество слов среди несовпадающих строк полученных файлов, содержащих информацию о Вашем терминале, заноса эту информацию в файл compare в Вашей домашней директории. (попробуйте все это выполнить в одной командной строке).
7	Выведите все строки всех файлов, находящихся в Вашей домашней директории (включая поддиректории), содержащих имя Вашего терминала, заноса эту информацию в файл compare Вашей домашней директории (Попробуйте все это выполнить в одной командной строке).
8	Создайте в домашней директории директорию Mythirdir. Выполните Ваше задание пункта 5) лабораторной работы, при условии, что рабочей директорией будет Mythirdir. Сравните директории Mysecondir и Mythirdir, заноса получаемую информацию в файл compare в домашней директории.
9	Найдите в Вашей домашней директории (включая поддиректории) файлы, содержащие в имени символ «d».
10	Создайте в Mysecondir файл sorted, содержащий в первой своей части построчную информацию о единицах хранения в корневой директории, отсортированную по именам файлов, во второй части – ту же информацию, но отсортированную по размерам файлов и в третьей части – отсортированную по правилам доступа, подсчитайте число строк в этом файле.
11	Создайте в Mysecondir файл sorted, содержащий в первой своей части построчную информацию о всех активных пользователях, отсортированную по именам пользователей, во второй части – ту же информацию, но отсортированную по именам терминалов, в третьей – отсортированную по временам входа в систему, подсчитайте число строк в этом файле.
12	Создайте в Mysecondir файл sorted, содержащий в первой своей части построчную информацию о всех активных процессах, отсортированную по именам пользователей, во второй части – ту же информацию, отсортированную в соответствии с номером процесса – по именам терминалов, в третьей – отсортированную по времени инициализации процесса, подсчитайте число строк в файле sorted.
13	Создайте в домашней директории директорию Mythirdir. В Mythirdir создайте ссылки на все Ваши существующие файлы, а также файл sorted, содержащий в каждой своей строке полную информацию (т.е. права доступа, размер, срок создания, владелец, имя и т.д.) о всех Ваших файлах, включая информацию о возможном содержимом файла, отсортированный по последнему.
14	Выполните дважды (бесконфликтно, например, в разных директориях) Ваше задание пункта 5) лабораторной работы. Из первого из полученных файлов получите строки, содержащие текст ttyAnn, где A – некоторая (любая) буква латиницы, nn – десятичный номер, а из второго – строки, не содержащие этот текст. Подсчитайте количество строк в первом и во втором случае, записав эту информацию в файл example.2 в Mysecondir.
15	Выполните дважды (бесконфликтно, например, в разных директориях) Ваше задание пункта 5) лабораторной работы. Из первого из полученных файлов получите строки, содержащие информацию о процессах с номерами от 100 до 999, из второго – строки, не соответствующие терминалам пользователей. Полученную информацию запишите в файл process.2 с подсчетом числа строк в нем самом.

16	Выполните вариант 9 из таблицы 1 задания к лабораторной работе. Выведите количество строк, слов и символов в полученном файле в файл lengths.
----	---

- 19) Включите Ваш терминал для приема сообщений от других пользователей, использующих команду write.
- 20) Зарегистрируйтесь с другого терминала и перешлите на терминал, использовавшийся в предыдущем пункте задания, сообщение вида "Hello, I am a user from ttyxx".
- 21). Выберите любой из разделов команды man и перешлите его между терминалами с припиской в конце о том, кто Вы с точки зрения ОС UNIX. Продемонстрируйте этот пункт работы преподавателю.
- 22) Завершите сеанс работы с ОС.

Контрольные вопросы:

- 1) Могут ли одновременно в ОС типа UNIX существовать два пользователя с одним и тем же именем, но работающие на разных терминалах?
- 2) Могут ли одновременно в ОС типа UNIX существовать два обычных разноименных пользователя, запускающие свои задания с одного и того же терминала?
- 3) Могут ли в одной и той же директории работать два различных пользователя одновременно?
- 4) Может ли один и тот же пользователь одновременно выполнять более одного процесса и при этом каждый процесс в своей директории?
- 5) Как можно выяснить связь между процессом с известным идентификатором и командой или программой, соответствующей этому процессу?
- 6) Какие файлы позволяют бесконфликтно применять к себе команду cat?
- 7) Какие команды из изученных в данной лабораторной работе возможно объединить через конвейер?
- 8) Какие команды из изученных в данной лабораторной работе допускают переназначение ввода и каким образом?
- 9) Какие команды из изученных в данной лабораторной работе допускают переназначение вывода и каким образом?
- 10) Какие команды из изученных в данной лабораторной работе допускают переназначение диагностики и каким образом?
- 11) Какие команды из изученных в данной лабораторной работе в результате верного вызова формируют ненулевой stderr и нулевой stdout?
- 12) Каким образом можно установить последовательность регистраций всех активных пользователей?
- 13) Может ли в ОС типа UNIX одновременно исполняться одинаковые процессы с одинаковыми именами у разных пользователей?
- 14) Может ли в ОС типа UNIX пользователь, перемещаясь по иерархической файловой структуре, попадать в директории, принадлежащие другим пользователям?
- 15) Какой последовательностью может считаться конец файла в ОС типа UNIX?
- 16) Поддиректорией какой директории является Ваша рабочая директория и поддиректорией какой директории является директория, поддиректорией которой является Ваша рабочая директория.
- 17) К чему может привести выполнение команды, удаляющей файл, на который другими пользователями установлена ссылка?
- 18) Есть ли различие в используемом объеме памяти при копировании файла в директорию и "линкования" его туда же?
- 19) Может ли домашняя директория содержать в качестве поддиректории корневую директорию?
- 20) Попробуйте написать команду перехода в директорию, одной из поддиректорий которой

является директория, одной из поддиректорий которой является директория, являющаяся рабочей для этой команды?

21) К чему могло бы привести выполнение команды `rm -rf ..`?

22) К чему могло бы привести выполнение команды `mv . ..`?

23) К чему могло бы привести выполнение команды `chmod u-rwx .` ?

24) К чему приведет выполнение команды:

- `mkdir dire?tory` ?

- `mkdir dire*tory` ?

- `mkdir dire[ckCK]tory` ?

25) При каких необходимых и достаточных условиях пользователь в рабочей директории может исполнить команду `mkdir` и команду `rmdir`?

26) Что делает (и делает ли) команда

`grep Los Angeles>file`

и что делает (и делает ли) команда

`grep 'Los Angeles'<file`?

27) Может ли директория иметь то же имя, что и директория, поддиректорией которой она является? То же имя, что и директория, в поддиректории которой она находится?

28) Может ли директория иметь несколько родительских каталогов? Несколько установленных ссылок на каталоги более высоких уровней?

29) Объясните значение полей строки команды `ls -la` для файла `/dev/dev1`.

30) Как получить распечатку файла `/usr/mydir/myfile.text`?

31) Что происходит при выполнении команды `write user1 tty5 < file.text` и что происходит при выполнении команды `cat file.text > /dev/tty5` ?

32) Как узнать, разрешено ли в данный момент поступление на Ваш терминал сообщений от других пользователей?

33) Подумайте, как можно передать информацию на другой терминал, не используя команду `write` (подсказка: терминалу соответствует файл устройства)?