

Пирогова М.А.

Сборник лабораторных работ
по курсу
ГРАФИЧЕСКИЕ СИСТЕМЫ

Оглавление

Лабораторная работа № 1. Основные понятия операционной системы (ОС) UNIX. Командная оболочка ОС UNIX. Графическая оболочка оконного менеджера ОС	1
Часть I. Командный интерфейс пользователя ОС UNIX. Основные сведения	2
Часть II. Файловая структура ОС UNIX	5
Часть III. Экранный визуальный редактор vi	8
Лабораторная работа № 2. X Window System. Ресурсы рабочей станции. Эмулятор терминала. Управление ресурсами и возможности пользователя по локализации приложений.	13
Лабораторная работа № 3. X Window System. Ресурсы. Управление ресурсами стандартных X-клиентов в X11	19

Лабораторная работа № 1. Основные понятия операционной системы (ОС) UNIX. Командная оболочка ОС UNIX. Графическая оболочка оконного менеджера ОС

Часть I. Командный интерфейс пользователя ОС UNIX. Основные сведения

Цель лабораторной работы

Целью данной работы является ознакомление с базисными понятиями **командного интерфейса** пользователя ОС UNIX. Основная 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 сервисных средств ОС UNIX для самообучения.

Командная оболочка ОС UNIX

Для всех пользователей, вот уже скоро двадцать 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>] [<параметры_вызова>] <BB>

где

- \$ - системный промптер («подсказка», выводимая в качестве признака готовности ОС UNIX воспринимать ввод команды пользователем);
- <команда> - любая из команд UNIX, либо утилита ОС;
- <ключи-вызова> - 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 промптера. 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: user <BB>

[password: <....><BB>]

\$>_

где

- **login**, **password** и **\$** - реакция системы;
- **user** - 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 и получения системного промптера 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** - на экран видеотерминала пользователя.

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

> - переадресация вывода (запись не в **stdout** по умолчанию)

< - переадресация ввода (чтение не из **stdin** по умолчанию)

>> - переадресация ввода с добавлением.

Кроме переадресации связь между процессами исполнения двух команд (UNIX-процессами в общем смысле) может быть осуществлена в виде т.н. конвейера - закорачивания 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>

где <команда-1> имеет смысл, определенный выше; выход процесса, определяемый командой <команда-1>, является входом процесса, определяемого командой <команда-2>.

Командный интерфейс пользователя кроме того предоставляет возможность UNIX-пользователю с помощью команд самого командного интерфейса изучать свойства наиболее употребительных 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, а для настройки служит команда **stty**. В стандартном, наиболее часто встречающемся варианте, пользователь для реализации интерфейса имеет в распоряжении терминал типа VT52/VT100, где для редактирования командной строки используются следующие последовательности:

- **<ctrl>+<h>** - для удаления отдельного символа в командной строке;
- **<@>** - удаление (смысловое) набранной, но не введенной строки командного интерфейса («сброс команды»),
- **** - завершение зависнувшего процесса и восстановление системного промптера, а для адаптации скорости выводимой процессами на экран видеотерминала информации служат последовательности:
- **<ctrl>+<S>** - приостановка вывода,
- **<ctrl>+<Q>** - возобновление вывода.

Задание к лабораторной работе

Домашняя подготовка

Выполнение пунктов лабораторной работы в этой части потребует изучения следующих вопросов и тем:

- системные ресурсы пользователя;
- вход в систему, выход из системы;
- рабочая директория, «домашняя» директория; Текущая директория;
- терминал пользователя и его параметры.
- функциональные особенности следующих команд командного интерфейса:
login, logout (exit), ls, ln, unlink, pwd, who (whoami), date, ps, cd, file, cat, more, man, stty.

Просмотрите лабораторное задание, соответствующее вашему варианту и обдумайте его выполнение.

Лабораторное Задание

1. Зарегистрируйтесь в системе с терминала.
2. Определите следующие параметры среды:
 - текущее время;
 - имя вашего терминала;
 - место вашей рабочей («домашней») директории;
 - имена и соответствующие им терминалы всех активных пользователей;
 - номера всех доступных для вашего наблюдения активных процессов;
 - параметры вашего терминала,
3. Создайте пустой файл (файл, не содержащий никакой информации) в вашей директории по имени **first.snap**;
4. Убедитесь в том, что ваша директория не пуста. Создайте файл **second.type**, содержащий информацию о типах всех файлов вашей директории, включая и файл **second.type**.
5. Вызовите команду **man** для просмотра информации о командах: **login, logout (exit, halt), ls, ln, pwd, who, date (time), ps, cd, ln, file, cat, more, man, stty**. Отметьте в создаваемом Вами файле **third.exeption** имена команд, по которым отсутствует информация при обращении к команде **man**. 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.
6. Повторите пункт 2 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 2 и содержимого файла **first.snap**.
7. Попробуйте добиться того, чтобы вывод на ваш терминал осуществлялся темными символами на светлом фоне.
8. Завершите сеанс работы с ОС.

Часть II. Файловая структура ОС UNIX

Цель лабораторной работы

Целью данной работы является ознакомление с возможностями, предоставляемыми командным интерфейсом пользователя ОС UNIX, для работы с файловой системой. Основная 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.

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, являющуюся точкой входа в файловую систему суперпользователя. Корневая директория не является при этом ничьим подкаталогом.

Обычный пользователь имеет свою собственную точку входа в файловую систему – «домашнюю директорию», с которой Вы уже познакомились на практике во время выполнения первой части лабораторной работы. Именование файлов (помните, что с точки зрения ОС UNIX директория - это тоже некоторый файл) определяется именованием 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 - (/)**.

Вершина иерархии файловой системы для каждой инсталляции операционной системы UNIX – это корневой каталог или корневая директория.

Корневая директория ОС UNIX - уникальна, существует всегда и однозначно определена на одно и то же место файловой системы для всех пользователей. Корневая директория не имеет имени, но при именовании пути от корневого каталога сам этот корневой каталог определяется единственным символом **slash (/)** в самой первой позиции имени пути. Пояснение:

путь aaa/ddd и путь /aaa/ddd

указывают на два разных каталога с именем ddd. Первый каталог расположен в подкаталоге aaa текущего каталога пользователя, а второй каталог указан с абсолютным именем и располагается в подкаталоге aaa корневого каталога ОС.

Эти два каталога могут совпадать только в том случае, если текущим каталогом пользователя (первый путь) является (стал) корневой каталог.

Любая другая директория, кроме корневой, текущая или рабочая (вспомните, что такое **rwd** для обозначения команды) директория пользователя, однозначно определены каждая для своего пользователя, может меняться пользователем во время сеанса. 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» - «.» - такая директория еще может называться «родительским каталогом».

Командный интерфейс ОС UNIX дает возможность пользователю выполнять следующие действия в файловой структуре, кроме тех, которые уже изучены в первой части лабораторной работе (см. **ls, file, cat, pwd** и, конечно же **cd**) :

- переименование файла **mv;**
- перемещение простого файла **mv;**
- копирование простого файла **cp;**

- | | |
|------------------------------------|---------------|
| • удаление простого файла | rm; |
| • создание нового каталога | mkdir; |
| • удаление каталога | rmdir; |
| • изменение атрибутов защиты файла | chmod; |
| • сортировку или слияние файлов | sort; |
| • поиск в файле по шаблону | grep; |
| • построчное сравнение файлов-I | diff; |
| • построчное сравнение файлов-II | comm. |

Отличительной особенностью ОС UNIX является то, что пользователь, если он имеет соответствующие права доступа, может 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**.

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 (например, BITNET, INTERNET) 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

Выполнение пунктов лабораторной работы в этой части потребует изучения следующих вопросов и тем:

- файловая структура и права доступа пользователя;

- использование периферийных устройств вообще и НГМД и АЦПУ в частности;
- общение пользователей в и вне ОС UNIX.
- функциональные особенности команд командного интерфейса ОС UNIX, отвечающих за:
- навигацию в файловой структуре: **ls, pwd, cd, file, cat, wc, mv, cp, rm, mkdir, rmdir, chmod, sort, grep, diff, comm, ln, diff** (факультативно: **cmp, grep, egrep, fgrep** - по заданию преподавателя);
- коммуникацию пользователей: **write, msg**, (факультативно: **elm, mail, mailk** и т.п. - по заданию преподавателя).

Лабораторное задание

1. Создайте в вашей домашней директории поддиректорию **Mysecondir**;
2. Скопируйте ваш файл **second.type** в директорию **Mysecondir**;
3. Переместите ваш файл **first.snap** в директорию **Mysecondir** с переименованием его в **second.snap**;
4. Установите для файла **second.snap** права доступа с абсолютным значением атрибута **777**;
5. Переустановите для директории права доступа внешнего пользователя такие, чтобы он сначала мог копировать из вашей директории, а потом не мог, вернитесь к исходному состоянию.
6. Создайте из **Mysecondir** ссылку на файл **first.snap** пользователя, определяемого по списку вашей группы по формуле $(n+1) \bmod(k)$, где n - ваш номер в списке группы, k - число студентов в вашей группе. Что Вам мешает?
7. Создайте директорию **Mysecondir** в вашей домашней директории и создайте в ней ссылку на директорию **Mysecondir** пользователя с номером $(n+2) \bmod(k)$. Что Вам мешает?
8. Вызовите команду **man** для просмотра информации о командах **mv, cp, rm, mkdir, chmod, sort, grep (Grep, egrep, fgrep), diff, comm, temp**. Добавьте в ваш файл **third.exeption** имена команд, по которым отсутствует информация при обращении к команде **man**. Не правда ли, «английский», на котором написаны разделы, читаемые **man**, ненамного сложнее диалекта Э. Шукиной и потому должен быть понятен Вам.
9. Включите ваш терминал для приема сообщений от других пользователей, использующих команду **write**.
10. Перешлите пользователю с номером $(n+3) \bmod(k)$ сообщение, типа .
«Hello, I am a user from ttyxx».
11. Выберите любой из разделов команды **man** и перешлите его пользователю с номером $(n+4) \bmod(k)$ с припиской в конце о том, кто Вы с точки зрения ОС UNIX. Продемонстрируйте этот пункт работы преподавателю.
12. Завершите сеанс работы с ОС.

Часть III. Экранный визуальный редактор **vi**

Цель лабораторной работы

Целью данной работы является ознакомление с основными возможностями стандартизированной на уровне применения расширенной версии базового редактора

текстовых файлов ОС типа UNIX. Основная практическая цель - выполнение индивидуальных заданий в рамках сеанса работы в среде vi, суть которых состоит в выполнении ввода текста, выполнении изменений введенного текста в файле, удаление фрагментов текста в файле, контекстного поиска/замены в текстовом файле, выполнении сервисных процедур настройки среды редактора vi.

Визуальный экранно-ориентированный редактор vi

«Генетически» визуальный редактор vi является идеологическим потомком стандартного для системы UNIX строчного редактора ed. Для работы в редакторах такого типа необходимо знать, в каком из режимов - командном или текстовом - находится пользователь, запросивший ресурсы редактора. В командном режиме объектом работы редактора является файл в целом, в текстовом – отдельная строка (строки) файла. Собственно работа с редактором vi начинается с вызова

\$ vi [<имя_редактируемого_файла>]

где <имя редактируемого файла> - имя текстового файла, по отношению к которому выполняются модифицирующие этот текст процедуры редактора vi; при отсутствии в системе файла с таким именем он будет создан для начала на время сеанса с vi, а при наличии файла с таким именем до вызова - vi отобразит первые строки этого файла на экране видеотерминала; в любом случае после вызова редактора он переходит в командный режим.

Для пользователя, находящегося в командном режиме визуального редактора vi, возможно выполнение следующих, необходимых для редактирования, действий:

- перемещение в редактируемом файле (перемещение курсора или маркера);
- контекстный поиск и замена в файле;
- выход из редактора с сохранением или без сохранения изменений в файле;
- ввод команд редактирования в позиционированной области файла.

Для возврата из текстового режима в командный пользователь, находящемуся в текстовом режиме, достаточно хотя бы один раз нажать клавишу <ESC>. Для пользователя, находящегося в текстовом режиме визуального редактора vi, доступны следующие, необходимые для редактирования, действия:

- ввод (вставка) текста с позиции/после позиции, отмеченной маркером;
- ввод (вставка) текста во вновь образованную пустую строку над строкой файла, отмеченной маркером.

После перехода в командный режим позиционирования маркера производится путем нажатия алфавитных клавиш **H** (влево), **J** (вниз), **K** (вверх), **L** (вправо), расположенных на стандартной QWERT - клавиатуре подряд. Возможно нажатие комбинации nn**H**, nn**J**, nn**K**, nn**L**, вызывающее сдвиг маркера в выбранном направлении сразу на nn позиций (nn – это число, вводимое путём нажатия на цифровые клавиши

стандартной алфавитно-цифровой клавиатуры: так, чтобы переместиться в файле на двенадцать строк вниз, нажмите последовательно три клавиши – **1**, **2** и **J**). Если заранее известен номер в файле позиционируемой строки, то для перемещения в эту строку можно воспользоваться комбинацией нажимаемых клавиш **xxG**, где **xx** - номер строки. Простое нажатие клавиши **G** перемещает указатель на самую последнюю строку файла, а нажатие **<CTRL>G** выводит на экран номер текущей строки (той, где позиционирован маркер).

Контекстный поиск, т.е. поиск некоторой последовательности символов, например **abed**, производится путем набора символа slash (**</>**), сразу за которым набирается последовательность контекста (**abed**), после чего нажимается клавиша **<BB>**. В результате курсор передвинется вперед по файлу к ближайшей последовательности **abed**, если такая существует. Переход к следующей последовательности контекста - по нажатию клавиши **N** - то же, но назад по файлу. Возможен вариант этой команды с поиском назад по файлу с заменой символа slash на **?**, тогда нажатие клавиши **G** вызовет переход к следующему контексту назад по файлу и клавиши **N** -вперед по файлу. Для замены слова, в котором позиционирован курсор, на любое другое необходимо ввести команду, задаваемую комбинацией клавиш **CW**, после чего, до нажатия клавиши **<ESC>**, редактор переходит в текстовый режим ввода заменяющей последовательности.

Для перевода редактора **vi** в режим ввода/вставки текста сразу за позицией курсора в редактируемом файле служит команда **a**, а для выполнения ввода/вставки текста в позиции курсора служит команда **i**. После ввода любой из этих команд редактор переходит в текстовый режим до нажатия клавиши **<ESC>**, возвращающей его снова в командный режим. В текстовом режиме пользователь имеет возможность непосредственного ввода с клавиатуры символов текста, пользоваться клавишей «Забой» (**<Backspace>** или **<-->**) в зависимости от типа клавиатуры) и клавишами табуляции. Для перевода редактора **vi** из командного режима в режим ввода текста во вновь образуемую пустую строку над позиционированной служит команда **O**, а для тех же действий над позиционированной строкой - команда **o**. В обоих случаях будет вначале образована пустая строка, курсор будет перемещен в первую позицию этой новой строки и **vi** переключится в текстовый режим до нажатия **<ESC>**.

Для одновременного выполнения контекстного поиска и замены служат более сложные командные последовательности. Эти командные последовательности вводятся в зоне команд редактора **vi**. В этой же зоне вводятся команды выхода из редактора. Для помещения курсора в зону команд (обычно это - самая нижняя из видимых строк видеотерминала, она не является частью редактируемого файла) необходимо в командном режиме набрать символ (нажать клавишу) двоеточие (**<:>**). Так, для замены **<старой__последовательности__символов>** вперед по файлу и замены только этой первой найденной последовательности служит последовательность

: n, Ns/<старая__последовательность_символов>/<новая__последовательность_символов>/S <BB>

То же, но с изменениями направления просмотра – назад по файлу:

:n, Ns?<старая__последовательность_символов>**?<новая__последовательность_символов>?S <BB>**

Выполнение в зоне команд действий по записи изменений и выхода из редактора кодируется последовательностью:

: wq

а для отказа от записи изменений, сделанных во время редактирования, используется последовательность:

: q!

В режиме использования зоны команд возможно обращение непосредственно из визуального редактора **vi** к некоторым командам ОС UNIX с возвратом в командный режим **vi** путем нажатия либо **<BB>** (нормальное завершение вызова команды ОС UNIX), либо **<CTRL>D** (анормальное завершение и возврат в **vi** при любых условиях). Так, для получения списка работающих в системе можно не выходя из **vi** выполнить следующее:

- перейти в командный режим;
- набрать последовательность **:! who <BB>**

В общем случае формат вызова следующий **:!<команда> <BB>**

Командный интерфейс визуального редактора **vi** гораздо богаче рассмотренных здесь случаев, однако уже этих примеров достаточно, чтобы осознать гибкость этого программного средства.

Задание к лабораторной работе.

Домашняя подготовка

Выполнение пунктов лабораторной работы в этой части потребует изучения следующих вопросов и тем:

- расширенные команды позиционирования **vi**;
- добавление нового текста в **vi**;
- изменение текста в **vi**;
- смещение текста в **vi**;
- удаление текста в **vi**;
- поиск/замена текста в **vi**.
- функциональные особенности вызова следующих команд визуального редактора **vi**:
(, {, }, [,], O, \$, H, M, a, A, I, I, o, O, cW, cw, >, <, », «, dw, dW, t, T, ct, cT, dt, dT, f, F, cf, cF, df, dF, r, R, s, S, y, p, P, dd, e, n, set

Лабораторное задание

1. Создайте в вашей домашней директории поддиректорию **Mytherdir**;
2. Получите у преподавателя ваш вариант текста, занесите его в файл **mytext.1.txt** с помощью редактора **vi**.
3. В редакторе **vi** в начале текста файла **mytext.1.txt** добавьте шифр своей группы и свой номер по журналу группы. Результат сохранить в файле.
4. В редакторе **vi** в конце текста файла **mytext.1.txt** добавьте текст что-нибудь, типа «Your most humble obedient servant, ..., student»), результат сохранить.
5. Выполните задания пунктов 3) и 4) в новых строках, создаваемых в средней части текста файла **mytext.1.txt**.
6. По заданию преподавателя произведите поиск подстроки вперед/назад по тексту файла **mytext.1.txt** с контекстной заменой на новую подстроку.
7. Скопируйте **mytext.1.txt** в **mytext.2.txt**.
8. В файле **mytext.1.txt** поменяйте местами первую и последнюю строки, а в этих строках - первые и последние слова, сохраните результат.
9. В режиме редактирования **mytext.1.txt** создайте файл **mytext.3.txt**, содержащий текущую дату, список активных пользователей и процессов, а также текст файла **mytext.2.txt**. Файл **mytext.2.txt** удалите, а текст файла **mytext.3.txt** поместите в файл **mytext.1.txt** начиная со второй строки, сохраните результат.
10. Завершите сеанс работы с ОС.

Контрольные вопросы

1. Системные ресурсы пользователя.
2. Функциональные особенности следующих команд командного интерфейса: **login, logout (exit), ls, ln, unlink, pwd, who (whoami), date, ps, cd, file, cat, more, man, sty** (по заданию преподавателя).
3. Файловая структура и права доступа пользователя.
4. Функциональные особенности команд командного интерфейса ОС UNIX, отвечающих за:
 - навигацию в файловой структуре: **ls, pwd, cd, file, cat, wc, mv, cp, rm, mkdir, rmdir, chmod, sort, grep, diff, comm, ln, diff** (факультативно: **cmp, grep, egrep, fgrep** - по заданию преподавателя);
 - коммуникацию пользователей: **write, msg**, (факультативно: **elm, mail, mailk** и т.п. - по заданию преподавателя).
5. Расширенные команды позиционирования **vi**.
6. Добавление нового текста в **vi**.
7. Поиск/замена текста в **vi**.
8. Функциональные особенности вызова команд визуального редактора **vi** – по указанию преподавателя.

Лабораторная работа № 2. X Window System. Ресурсы рабочей станции. Эмулятор терминала. Управление ресурсами и возможности пользователя по локализации приложений.

Цель лабораторной работы

Целью данной работы является ознакомление с понятием «ресурсы Рабочей Станции (РС)» на примере командного интерфейса X Window System (X11) для настройки стандартными способами X-Клиента эмулятора терминала **xterm**.

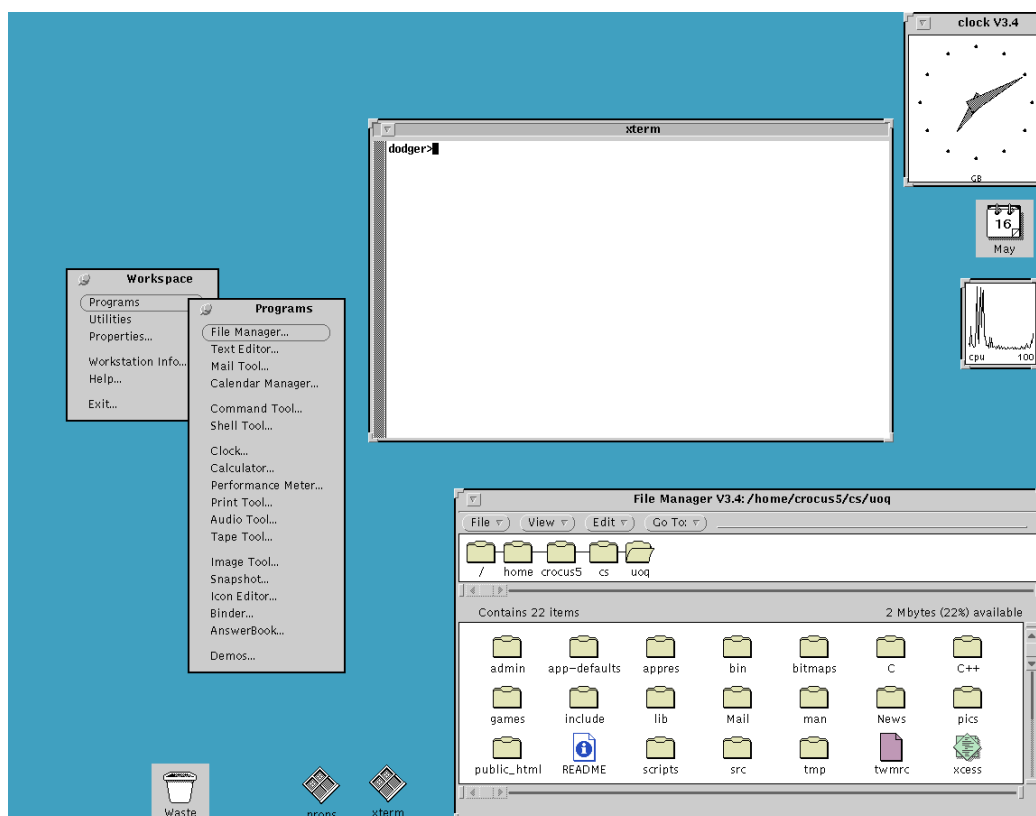
Пояснения к лабораторному заданию

Эмулятор терминала, приложение терминала, term или **tty**, для краткости — это программа, которая эмулирует терминал компьютера внутри некоторой другой архитектуры вывода данных на экран.

xterm является стандартным эмулятором терминала для среды **X Window System** в Unix. Пользователь имеет возможность работать с несколькими **xterm** терминалами, запущенными в одно и то же время на одном и том же дисплее. Каждый из виртуальных терминалов предоставляет независимый ввод/вывод для процессов, запущенных в каждом из них (обычно это **Unix shell** процессы).

В то же время **xterm** является одним из первых X-Клиентов, разработанных в рамках проекта создания стандарта **X Window System (X11)**.

В процессе выполнения Лабораторной работы студенты изучают стандартные способы локализации Графического пользовательского интерфейса приложений, относящихся к классу X-Клиентов, т.е. разработанных с помощью Инструментального пакета высокого уровня (ИПВУ) **Xt**, а также возможности пользователя по доступу к таким графическим ресурсам РС, как база цветов и фонтов.



Домашняя подготовка

Выполнение пунктов лабораторной работы потребует изучения теоретических вопросов, находящихся в документе [Lecture to LAB02_03.pdf](#).

Лабораторное задание

1. Зарегистрироваться в системе. Стартовать X-сессию известными вам способами. Рекомендуется работать в стандартном графическом окружении **Window Manager'a X11 - OPEN LOOK**:

\$openwin

2. Проверить состояние ресурсов - найти и просмотреть состояние базы данных цветов - файл **rgb.txt**. Для реализации поиска использовать утилиту **find**.

3. Создать файл, содержащий все доступные X-сессии фонты. Для создания этого файла воспользоваться утилитой **X11 xlsfonts**. Результат выполнения этой утилиты (стандартный поток вывода - **stdout**) перенаправить в файл **allfonts.txt**:

\$xlsfonts > allfonts.txt

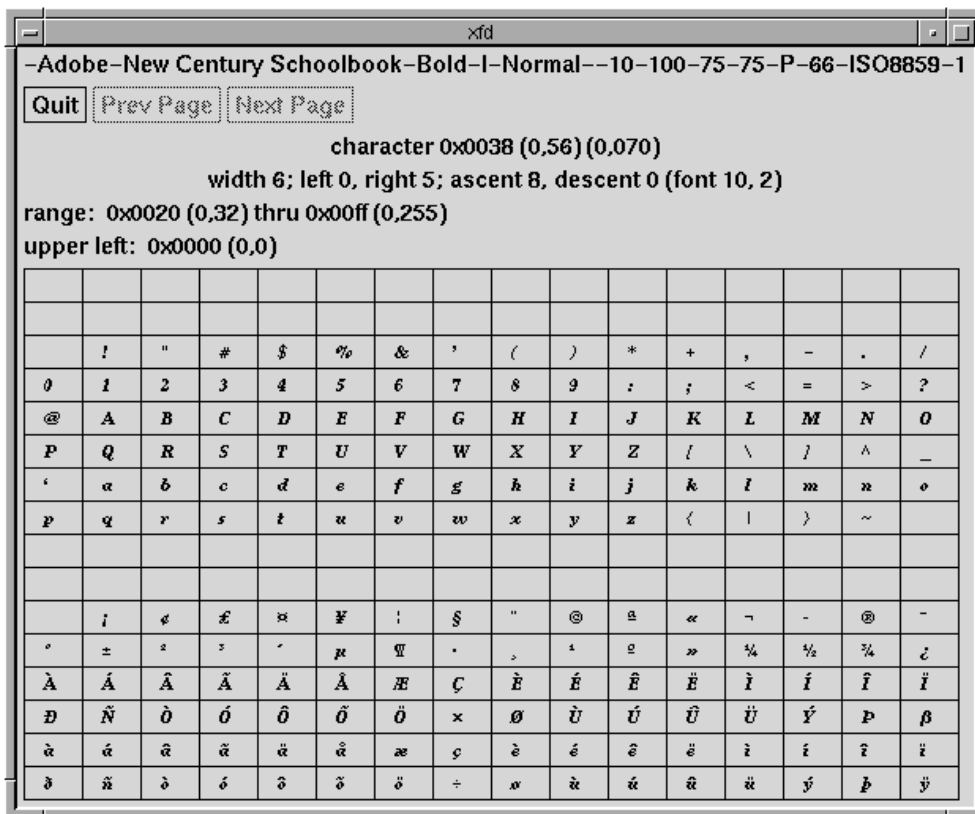
4. Определить количество доступных цветов и фонтов. Для определения доступного количества цветов и фонтов необходимо использовать соответственно утилиты **X11 xlsfonts** и **showrgb** (к примеру; однако, если эта утилита отсутствует, следует получить количество доступных цветов другим известным вам способом), перенаправив их выходные потоки в файлы (см. пункт 3 данной лабораторной работы), а затем подсчитать в этих файлах количество строк с помощью утилиты **UNIX wc** с ключом **-l**:

\$xlsfonts > allfonts.txt \$showrgb > allcolors.txt \$wc -l allfonts.txt \$wc -l allcolors.txt

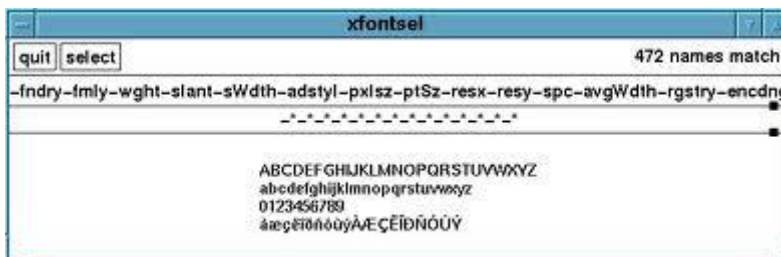
5. С помощью команды **xfd** просмотреть особенности семейства фонтов. Особенности требуемого семейства фонтов (выбор осуществляется по номеру бригады в соответствии с заданием на ЛР) просмотреть с помощью утилиты **X11 xfd** (например, для семейства **times**):

\$xfd *times*

- 1 бр. - foundry (adobe)
- 2 бр. - foundry (b&h)
- 3 бр. - foundry (misc)
- 4 бр. - family (courier)
- 5 бр. - family (times)
- 6 бр. - family (lucida)
- 7 бр. - family (helvetica)
- 8 бр. - все фонты с высотой (pixels) - 14



6. С помощью команды **xfontsel** выбрать какой-либо шрифт. Эта утилита X11 позволяет осуществлять выбор в интерактивном режиме в открывающемся после вызова утилиты окне.



7. Стартовать **xterm** в фоновом режиме. Вывести в это окно **список цветов (для четных бригад)/фонтов (для нечетных бригад), доступных** данной X-сессии. Попытаться просмотреть весь список.

Для выполнения этого пункта ознакомиться с ресурсами и способами их задания из командной строки при запуске **х-клиента xterm** с помощью команды ОС UNIX **man**.

Стартовать **xterm** в **фоновом режиме** с параметрами (ресурсами) по умолчанию:

\$xterm &

Вывести в стартованное окно список цветов/фонтов, доступных для данной X-сессии, например:

\$xlsfonts | more

Попытка просмотреть сразу весь список не удалась, т.к. буфер для запоминания такого количества строк слишком мал.

8. Стартовать **xterm** в фоновом режиме с буфером необходимой длины для просмотра и с добавлением режима построчной прокрутки **-scrolling**.

Результат продемонстрировать преподавателю.

Выполнение этого пункта возможно с использованием командного интерфейса X-утилит - изменение ресурсов **x-клиента (окна) xterm** с помощью опций командной строки вызова. Необходимо поменять величину буфера и добавить полосу прокрутки для просмотра файлов, длина которых укладывается в заданную глубину буфера.

Для этого используем ресурсы **-sb (scrollbar - прокрутка)** и **-si (буфер)** заданной длины, например - 1500 строк:

\$xterm-sb-sl 1500 &

9. Стартовать **xterm** в фоновом режиме с полным цветовым оформлением. Стартуем **xterm** в фоновом режиме с полным цветовым оформлением, т.е. задаем следующие ресурсы: цвет текста – «передника» в окне (**foreground, ключ -fg**), цвет «задника» - фона в окне (**background, ключ -bg**), цвет рамки окна - ключ **-bd**:

\$xterm -fg White -bg Black -bd Red &

Можно попытаться изменить цвета курсора (**-cr**) и пойнтера (**-ms**).

10. Стартовать **xterm** в фоновом режиме с полным фоновым оформлением. Стартуем **xterm** в фоновом режиме с полным фоновым оформлением. Для реализации этого пункта задания использовать утилиты **xfd** и **xfontsel**.

Ключ для задания соответствующего ресурса **x-клиента xterm -fn (fontname)**, например:

\$xterm -fn "-adobe-helvetica-*" &

(попробовать различные фонты и их описание, манипулируя символом *)

11. Стартовать **xterm** в фоновом режиме, изменив стандартный заголовок окна на пользовательский - **Ф.И. членов бригады**, и наименование пиктограммы - на N бригады. Стартуем **xterm** в фоновом режиме, изменив стандартный заголовок окна на пользовательский - **Ф.И.О. членов бригады**, и наименование иконки (окно в свернутом состоянии) - на № бригады. Для этого использовать ресурсы **Title (ключ -T)** и **-iconic** (при использовании этого ключа в строке запуска окно **xterm** откроется сразу же как иконка) с именем name (ключ **-n**):

\$xterm -T "Ф.И.О членов бригады" -n "№ бригады" &

12. Стартовать **xterm** в фоновом режиме по пунктам 6-11 сразу как икону. Раскрыть.

13. Стартовать **xterm** в фоновом режиме с заданием полей в n пикселей, где $n=4 \times (N \text{ бригады})$.

\$xterm -b n &

14. Стартовать **xterm** в фоновом режиме. Реализовать старт **xterm** с заранее определенными параметрами - фон, цвет, поля, величина буфера, прокрутка, иконизация, изменение заголовка окна и иконы. Оформить в виде командного файла. Результат продемонстрировать преподавателю.

Вариант задания, т.е. тип оболочки и конкретных параметров для скриптов, выбрать в соответствии с номером бригады из Таблицы:

Для бригад №. 1,3,5,7,9 - **/bin/sh**;

Для бригад №. 2,4,6,8 - **/bin/csh**

Номер бригады	Параметр	Значение параметра
1,2,3	фон	выбрать третьим из списка имен foundry
1,4,7	цвет	background - n -й из списка всех цветов foreground — m -й "-----". Текст должен быть различим
1,5,9	величина буфера	a+b , где a - по выбору преподавателя, но д.б. положительным, b - квадрат числа a
4,5,6	фон	выбрать n -ым из списка фонов, отсортированного по возрастанию размеров
2,5,8	цвет	background - 3-й из списка цветов, содержащих символ "с" первым в имени цвета, foreground — 3-й из списка цветов, содержащих символ "g" первым в имени цвета
2,6,8	величина буфера	a+b , где $a = a1 - a2 $, $b = a1 \cdot a2$, где a1 и a2 - по выбору преподавателя - целые положительные числа
7,8,9	фон	выбрать n -ым из списка всех courier
3,6,9	цвет	background - по выбору из всех цветов foreground — отстоящим на двадцать пять позиций в списке всех цветов от background
3,7,4	величина буфера	a+b , где a равно числу файлов в каталоге /etc , b равно числу каталогов в каталоге /dev

15. Закрывать X-сессию, выйти из системы.

Контрольные вопросы

1. Ресурсы Рабочей станции. База данных цветов и фонты. Способы доступа.
2. Основные X-Клиенты и X-Утилиты реализаций X11 для UNIX-подобных операционных систем.
3. Эмулятор терминала. Запуск в фоновом режиме.
4. Ресурсы X-Клиентов. Настройка окон с помощью опций командной строки.
5. Механизм конфигурирования ресурсов в X11.
6. Понятия ресурсной базы данных и менеджера ресурсов в X11.

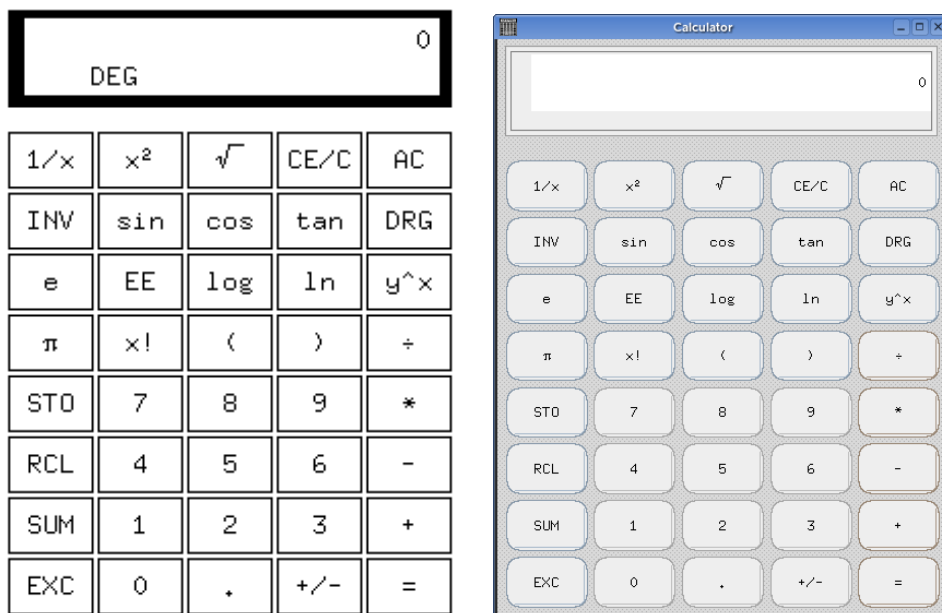
Лабораторная работа № 3. X Window System. Ресурсы. Управление ресурсами стандартных X-клиентов в X11.

Цель лабораторной работы

Целью данной работы является дальнейшее изучение стандартных для X11 способов загрузки в ресурсную базу данных X-сервера ресурсов, необходимых для воспроизведения окна X-Клиента в стартованной X-сессии. Изучение имеющихся в распоряжении пользователя способов в порядке их приоритетности осуществляется на примере одного из самых популярных X-Клиентов **X Window System**, реализованных для большинства популярных ОС и для различных **WM, xcalc**.

Пояснения к лабораторному заданию

Примеры GUI **xcalc** для различных наборов управляющих элементов (**widgets**) оконных менеджеров приведены ниже:



Выбор данного X-Клиента обусловлен тем, что все ресурсы управляющих элементов, образующих GUI этого приложения, вынесены в конфигурационный файл **Xcalc application class resource file**, что позволяет осуществлять настройку приложения в соответствии с требованиями пользователя путем простого редактирования текстового ресурсного файла.

Домашняя подготовка

Выполнение пунктов лабораторной работы потребует изучения теоретических вопросов, находящихся в документе [Lecture to LAB02_03.pdf](#).

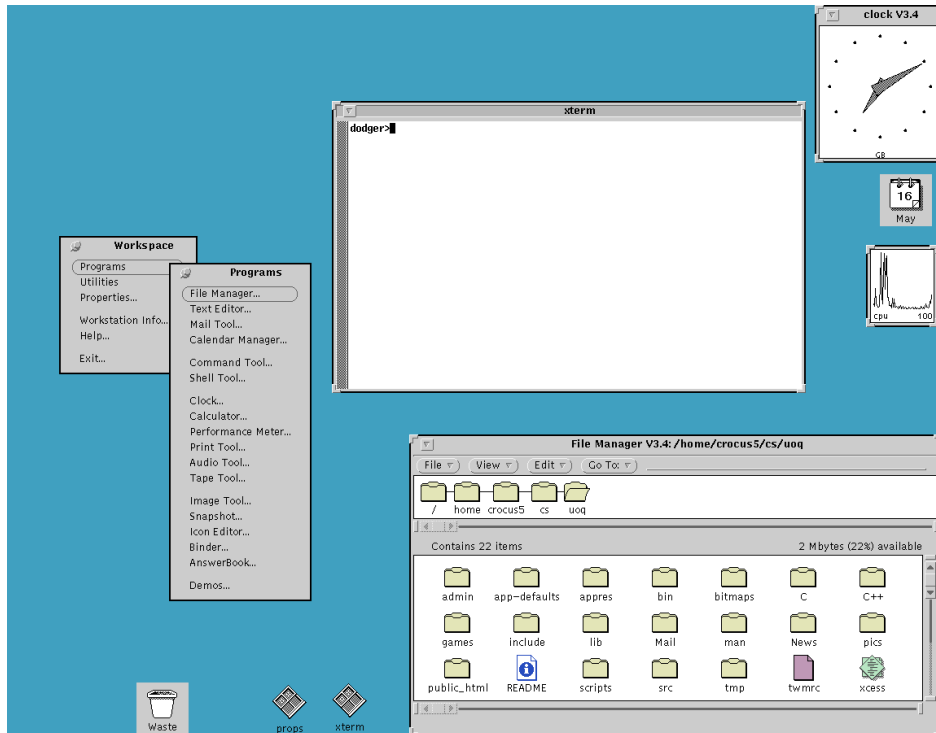
Лабораторное задание

1. Зарегистрироваться в системе. Стартовать X-сессию известными вам способами.
2. Выполнить заполнение файла **HELP.X11.SOME.XCLIENTS** текстами инструкций по работе со следующими базовыми X-Клиентами X11, воспользовавшись командой

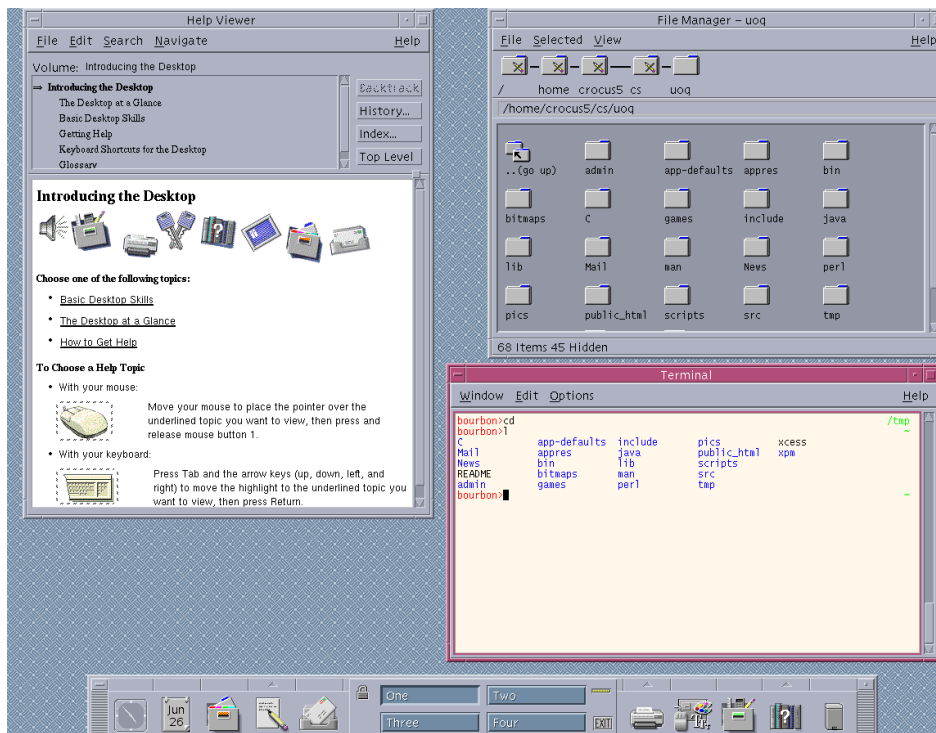
man: bitmap, oclock, xclock, xterm, xlogo, xload, xcalc, xsetroot.

3. В соответствии с инструкциями из файла **HELP.X11.SOME.XCLIENTS** вызвать х-клиентов из пункта 2. Ниже приведены примеры экранов с корневым окном X11 и несколькими окнами вызванных базовых X-Клиентов для различных наборов управляющих элементов популярных оконных менеджеров (WM), например:

- для **OpenLook Window Manager:**



- для **CDE**, стартованного из под OC Solaris:



- для Motif Window Manager



4. Установить **root-окно** с новым цветом/узором пользователя, выбрав его из списка загруженных, а затем создав самостоятельно.

Стандартный интерфейс для данного X-Клиента в среде **Linux** выглядит следующим образом:

Синтаксис:

```
xsetroot [-def] [-gray|grey] [-mod x y] [-solid цвет]
```

Описание:

Установка свойств корневого окна, включая цвет и режимы заполнения. В современных графических интерфейсах типа **GNOME** и **KDE** используются собственные механизмы управления свойствами корневого окна (фона рабочего стола).

Опции:

-def	Установка режимов по умолчанию
-gray grey	Задать серый фон
-mod x y	Установить заполнение сеткой. Величины x и y могут лежать в пределах от 1 до 16
-solid цвет	Заполнить фон цветом'ом равномерно

Примеры:

```
xsetroot -mod 12 12 -bg darkviolet -fg darkblue
```

Для выбора узора задника корневого окна можно воспользоваться общим видом командной строки с дополнительными опциями:

Синтаксис:

```
xsetroot [-help] [-def] [-display display] [-cursor cursor-  
file maskfile] [-cursor_name cursorname] [-bitmap filename]  
[-mod x y] [-gray] [-grey] [-fg color] [-bg color] [-rv] [-  
solid color] [-name string]
```

Для выбора любого другого, кроме сетки, узора задника корневого окна из списка доступных необходимо использовать опцию `-bitmap`, которая вызывает приложение X11, позволяющее вызывать узор из списка доступных, редактировать его, или создавать пользовательские образцы узоров.

Синтаксис:

```
bitmap [ -options ... ] [ filename ] [ basename ]
```

5. Для перечисленных в пункте 3 х-клиентов переустановить следующие ресурсы с помощью опций командной строки :

- сменить **foreground**,
- сменить **background**,
- сменить **font**,
- изменить ресурс «геометрия» (геометрические размеры и начало координат) окна вывода любого из базовых клиентов X11.

6. Составить собственный файл ресурсов для:

- установки ресурсов **xterm** - задать пользовательские **font**, **background**, **Foreground**, **scrollbar**,
- изменить стандартные **foreground** и **background** для х-клиентов **xclock**, **xlogo**, **xload**;

Для выполнения этого пункта работы использовать два способа настройки стандартных Х-Клиентов – с помощью опций Х-утилиты **xrdb**, и отредактировав файл с зарезервированным именем **/.Xdefaults**, размещенного в домашней директории пользователя. Объяснить результат применения этого файла.

Примечание. Пример ресурсного настроечного файла для некоторых базовых Х-клиентов, который может быть подгружен в ресурсную базу данных локального Х-Сервера с помощью утилиты **xrdb**, приведен в файле [XclientResFile.pdf](#). Если подобному ресурсному файлу присвоить зарезервированное имя **/.Xdefaults** и разместить его в домашней директории пользователя, то при каждом очередном старте

X-сессии, он автоматически будет подгружаться в ресурсную базу данных X-Сервера, меняя значения прописанных в нем ресурсов с заданных на пользовательские.

7. С помощью утилиты **xrdb** настроить клиента **xcalc** в соответствии с индивидуальным заданием.

- Получить индивидуальное задание у преподавателя. Задание заключается в изменении GUI приложения **xcalc** (например, изменить форму кнопок, их взаимное расположение, обозначения функций на них, цвет задника и передника на отдельных кнопках или для всего GUI, и т.д.);
- Для выполнения задания составить ресурсный настроечный файл, который позволил бы изменить соответствующие значения ресурсов **xcalc** по умолчанию на пользовательские;
- Применить составленный ресурсный файл любым из изученных в пункте 6. способом изменения содержания ресурсной базы данных X-Сервера.

Примечания:

1. Примеры ресурсных настроечных файлов для изменения значений ресурсов по умолчанию для базового X-Клиента X11 – **xcalc** приведены в файле [XcalcResFile.pdf](#). Подобный файл может быть подгружен в ресурсную базу данных локального X-Сервера с помощью утилиты **xrdb**. Если подобному ресурсному файлу присвоить зарезервированное имя **/.Xdefaults** и разместить его в домашней директории пользователя, то при каждом очередном старте X-сессии, он автоматически будет подгружаться в ресурсную базу данных X-Сервера, меняя значения прописанных в нем ресурсов с заданных на пользовательские.
 2. Особенностью базового X-Клиента X11 **xcalc** является то, что все его ресурсы, как те, что отвечают за внешний вид его GUI, так и те, что определяют модель поведения пользователя, работающего с этим приложением, собраны в специальном ресурсном файле типа **Xcalc application class resource file**, расположенном в специальной директории файловой системы ОС (**/usr/lib/X11/app-defaults/<classname>**). Это текстовый файл, доступный для изучения и редактирования (в зависимости от прописанных системным администратором прав привилегированных и обычных пользователей). Настройка ресурсов GUI **xcalc** при старте этого приложения производится в соответствии с правилами создания ресурсной базы данных клиентов (см. [Lecture to LAB02_03.pdf](#)), Шаг № 5. Для изучения способов задания значений ресурсов X-Клиента X11 **xcalc** см. [XcalcAppDefFile.pdf](#)
8. Закрывать X-сессию, выйти из системы.

Контрольные вопросы

1. Концепция ресурсов. Для чего используется механизм ресурсов в **X Window System**?
2. Механизм конфигурирования ресурсов. Формат строки конфигурационных файлов.
3. Формирование ресурсной базы данных X-Сервера. Механизмы наследования.
4. Откуда управляющий элемент GUI может взять значение ресурса, необходимого ему для воспроизведения своего окна в корневом окне PC?
5. Стандартные способы формирования ресурсной базы данных X-Сервера. Приоритетность этих способов.
6. Стандартные конфигурационные ресурсы X-Клиентов X11.
7. Утилита **xrdb**. Опции, позволяющие осуществлять настройку X-Клиентов в соответствии с требованиями конечного пользователя.
8. Дополнительные X-Утилиты для анализа окружения окон после старта X-Сессии.
9. Иерархия управляющих элементов X-Клиента **xcalc**. Способы настройки и изменения ресурсов по умолчанию. Сравнить с другими базовыми X-Клиентами.