

Московский Государственный Технический Университет
имени Н. Э. Баумана
Факультет «Информатика и системы управления»
Кафедра «Компьютерные системы и сети»

УТВЕРЖДАЮ

Зав. кафедрой ИУ6

д.т.н., проф. _____ Сюзев В.В.

" ____ " _____ 2013 г.

Исследование среды управления и методов защиты
Методические указания по выполнению лабораторной работы
по дисциплине "Операционные системы"

Часть 1.

Исследование среды управления Linux.

МОСКВА 2013

Часть 1. Исследование среды управления Linux.

В повседневной работе системный администратор часто использует служебные программы для конфигурирования учетных записей пользователей, модификации ПО и параметров служб, установки нового оборудования и т. д. Для управления ОС можно использовать средства, которые позволяют настраивать и приспосабливать различные утилиты под конкретные потребности. Например, можно встретить консоль управления, которая предоставляет стандартный интерфейс для одного или нескольких приложений, называемых *оснастками* (snap-in), которые применяются для конфигурирования элементов вашей среды. Эти оснастки приспособлены для решения конкретных задач, их можно упорядочивать и группировать в рамках консоли согласно вашим предпочтениям

Цель 1-й части - исследование среды управления Linux.

Введение.

В данной работе на проработку выносятся следующее:

- Окружения рабочего стола (DE) Linux (GNOME, KDE, Xfce, LXDE);
- Настройка Openbox;
 - Установка и настройка панелей;
 - Настройка autostart;
 - Настройка рабочего стола;
 - Отображение иконок;
 - Установка и настройка conky;
- Управление с помощью консоли;
 - Основные команды и утилиты;
 - Основные утилиты;
 - Объединение команд;
 - Более сложные утилиты;
 - top, ps, pstree;
 - ping, netstat, ip;
 - Написание скриптов на bash.

Окружения рабочего стола (DE) Linux

Для UNIX-подобных систем существует множество окружений рабочего стола (Desktop Environments, DE). Некоторые UNIX-подобные системы запрещают смену DE, но большинство позволяют пользователю самому выбирать подходящее рабочее окружение. Обычно выбор окружения рабочего стола происходит при выборе дистрибутива. Например, Debian предлагает для скачивания образы с GNOME, KDE, Xfce и LXDE. Другой вариант – загрузить дистрибутив только с консольным интерфейсом и установить DE самостоятельно.

Необходимо отметить, что зачастую понятия DE и WM используются неправильно. DE включает в себя WM (менеджер окон), отвечающий за такие действия, как, например, отображение рамок вокруг окон, поддержание одних окон поверх других и т.д., но DE также включает в себя и дополнительные утилиты, например, системный трей.

В данной части лабораторной работы будут приведены теоретические части по самым популярным из существующих окружений рабочего стола.

GNOME

Среда GNOME (GNU Network Object Environment) разрабатывается с 1997 года как полностью свободная рабочая среда, ориентированная на обычного пользователя.

На данный момент GNOME основан на следующих принципах:

- Открытость – исходные коды полностью доступны.
- Организация и поддержка – стабильные релизы выпускаются по жесткому циклу разработки, поддержка производится сообществом.
- Интернационализация – пакеты GNOME активно переводятся на множество языков.
- Доступность – предполагается, что среда должна быть понятна любому пользователю независимо от его уровня владения компьютером, а также предлагать специальные функции для людей с ограниченными возможностями.

- Дружественность к разработчикам – этот принцип вытекает из нескольких предыдущих, он означает, что разработчик прикладного ПО для среды (или самой среды) должен иметь полный доступ к документации и инструментам разработки. Это позволяет разрабатывать ПО на различных языках программирования, при этом легко интегрируя различные компоненты в среду.

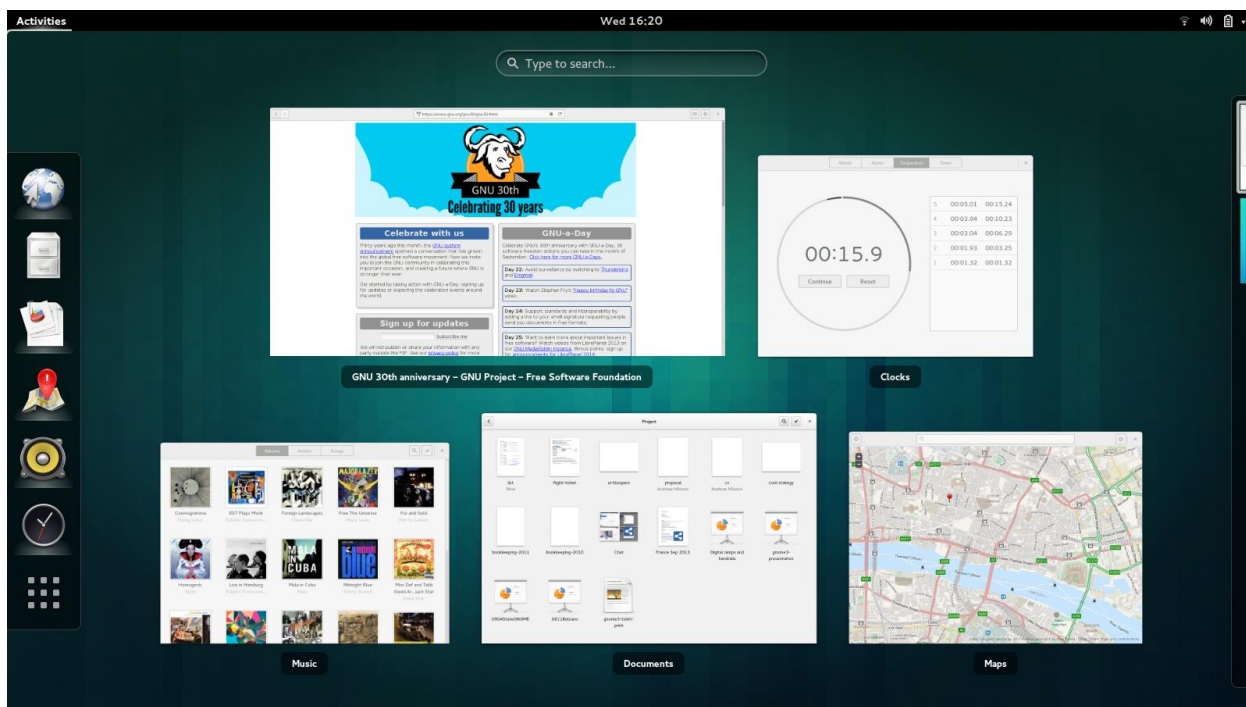


Рисунок 1 – GNOME 3

GNOME можно назвать оптимальным выбором для новичка, он лёгок в освоении и запускается на многих машинах, поддерживает мини-приложения («апплеты»). Начиная с версии GNOME 3, разработчики стали гораздо больше внимания уделять визуальным эффектам, что не лучшим образом сказалось на требованиях к ресурсам.

KDE

KDE (K Desktop Environment с 2010 года – KDE SC, KDE Software Compilation) – окружение рабочего стола и набор программ, использует Qt. Проект разрабатывается с 1996 года, ранее критиковался за использование несвободного ПО (библиотеки Qt).

С версии 4 использует фреймворк Plasma, позволяющий писать приложения с визуальными эффектами (прозрачность, тени и т.д).

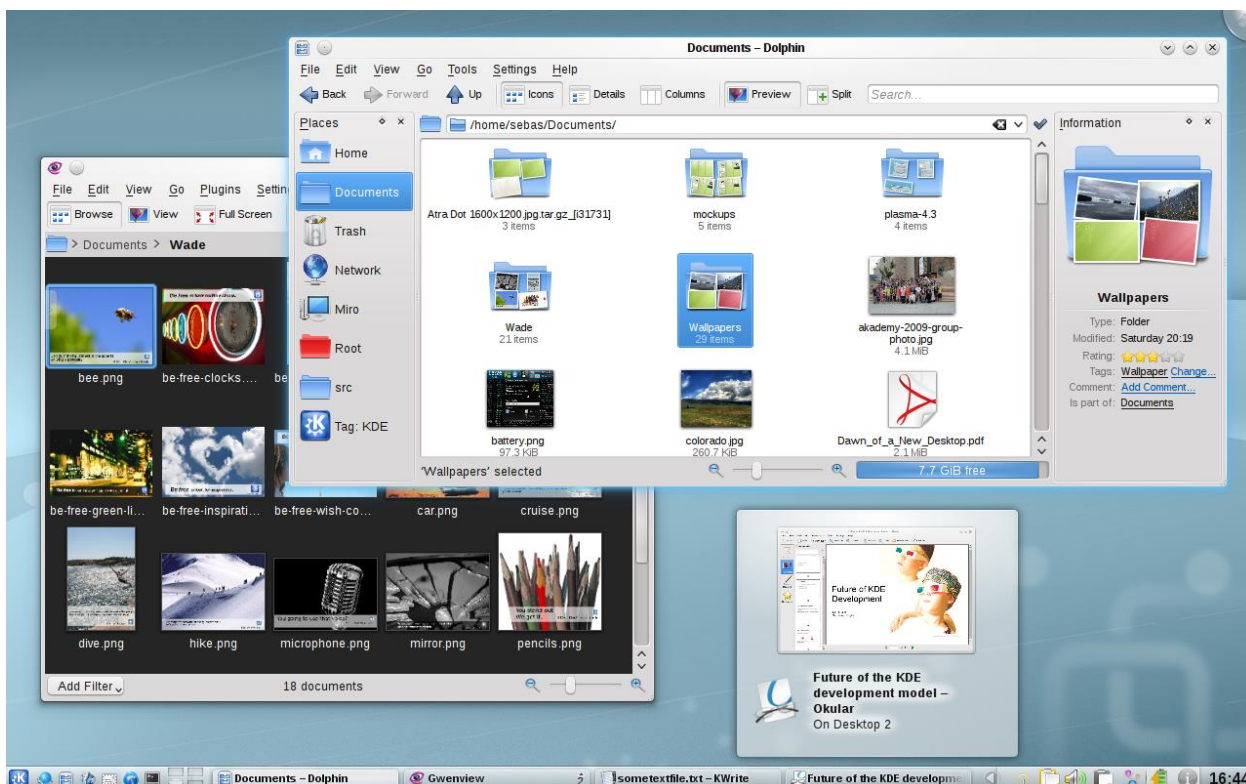


Рисунок 2- KDE с Plasma

Также с версии 4 появились мини-приложения (виджеты) «плазмоиды».

KDE можно считать оптимальным выбором для начинающего пользователя с достаточно мощным компьютером: хотя по умолчанию KDE потребляет не так уж много ресурсов, чрезмерное «увлечение» неопытного пользователя «плазмоидами» и визуальными эффектами может замедлить работу системы. Бытует мнение, что KDE потребляет очень много ресурсов, однако в руках опытных пользователей данная среда работает достаточно стабильно.

Xfce

Xfce является лёгким и быстрым окружением рабочего стола, отвечающим требованию модульности. Является очень гибкой средой – с помощью стандартных настроек может использоваться как окружение рабочего стола с упрощённой графикой, так и различными эффектами. Благодаря использованию библиотек GTK 2, поддерживает апплеты GNOME.

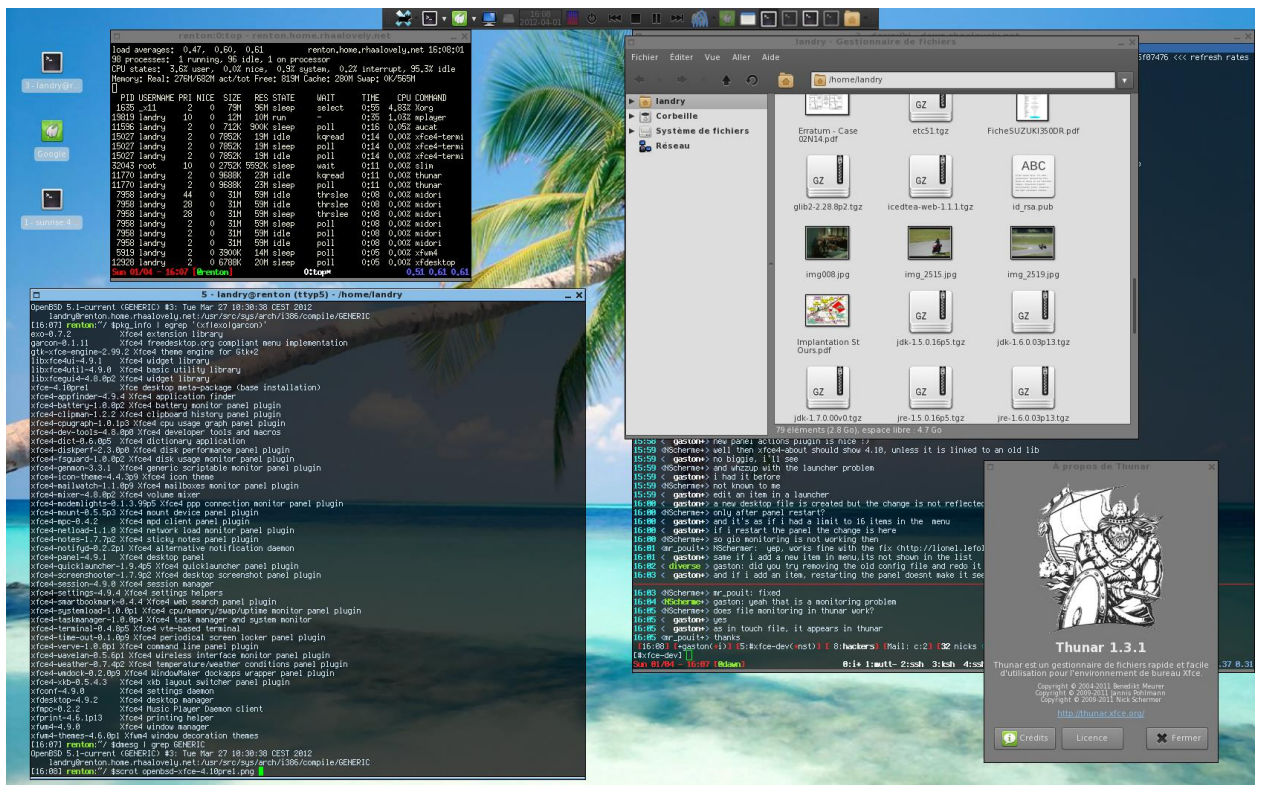


Рисунок 3 – Xfce 4.10

LXDE

LXDE (Lightweight X11 Desktop Environment) – легковесная среда X11, разрабатывается с 2006 года. Позиционируется как энергоэффективное и нетребовательное к ресурсам решение для широкого круга пользователей. Используется как на x86-компьютерах, так и на ARM-устройствах (была доступна для Maemo).

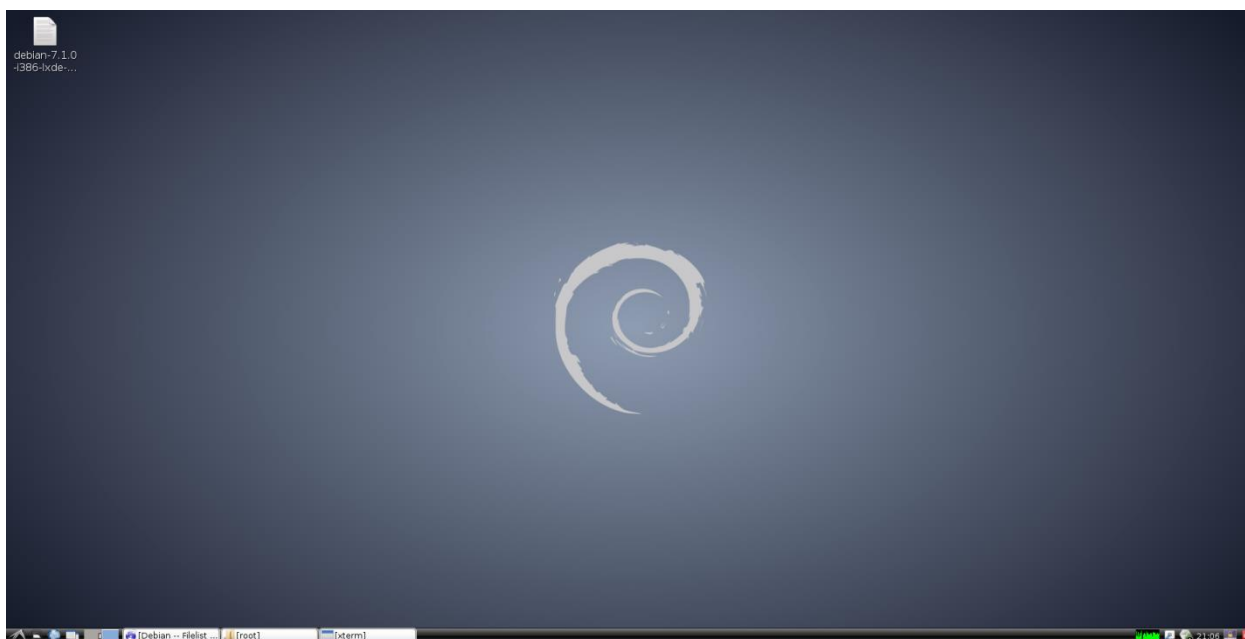


Рисунок 4 – LXDE в Debian 7.0

Настройка Openbox

Openbox – популярный оконный менеджер, который используется как с DE, так и отдельно. В данной части лабораторной работы будет проведена настройка Openbox и некоторых прикладных программ, которые позволяют организовать рабочее пространство. *Курсивом* выделены пункты, необязательные для выполнения.

Для выполнения данной ЛР необходимо войти в систему как суперпользователь (root).

Установка и настройка панелей

По умолчанию Openbox предоставляет пользователю лишь рабочий стол с контекстным меню. Для получения более привычного способа управления нужно установить панели. Среди самых популярных панелей для Openbox можно выделить xfce4-panel, lxpanel и tint2. Первые две входят в состав Xfce и LXDE соответственно.

Для установки xfce4-panel необходимо выполнить команду:

```
apt-get install xfce4-panel
```


Другие перечисленные панели также доступны для установки из репозитория, добавленного в ходе ЛР 2.

Для запуска панели необходимо набрать в терминале:

```
xfce4-panel
```

После этого будет показан диалог первого запуска, необходимо выбрать «Использовать настройки по умолчанию». Результат показан на рисунке 5.

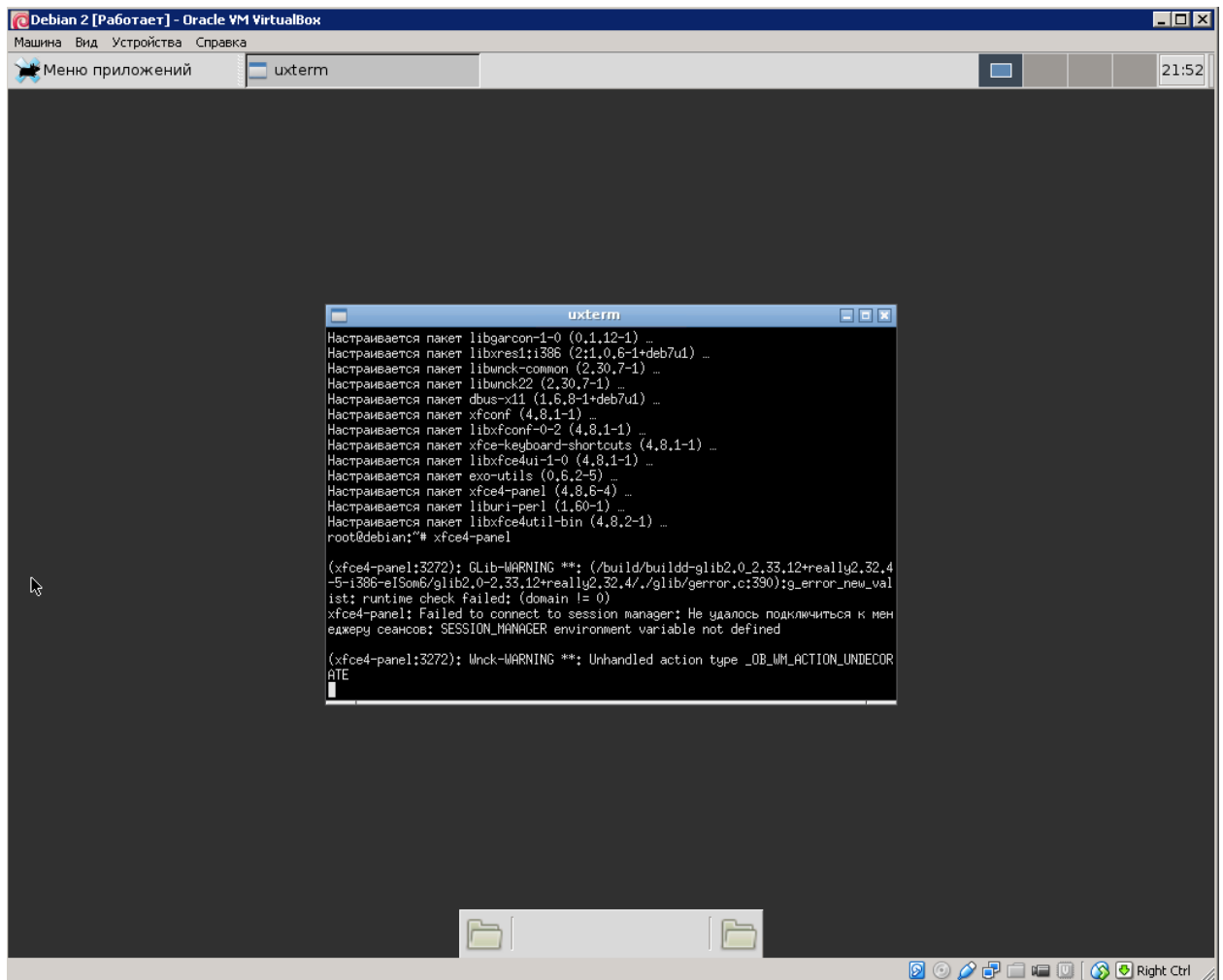


Рисунок 5 – Панель xfce4 в Openbox

Однако при закрытии окна консоли панель пропадёт. Она пропала из-за того, что родителем процесса панели являлся процесс окна консоли. Этот недостаток будет исправлен в следующем пункте.

На верхней панели располагаются следующие элементы (слева направо): меню приложений, меню системных действий (иконка отсутствует, однако при наведении будет отображаться подсказка), кнопки окон, переключатель рабочих мест, часы.

Переключатель рабочих мест позволяет переключаться между виртуальными рабочими столами. Как видно из рисунка 5, он также отображает положение окон на рабочем столе, если окно будет развёрнуто на весь экран, переключатель окон отобразит его иконку.

На нижней панели присутствуют две иконки, ещё 4 – не определены. Это связано с тем, что в нашем дистрибутиве не заданы некоторые параметры (например, браузер по умолчанию).

Настроим верхнюю панель. Для этого нужно щёлкнуть правой клавишей мыши по панели, после чего появится меню, показанное на рисунке 6.

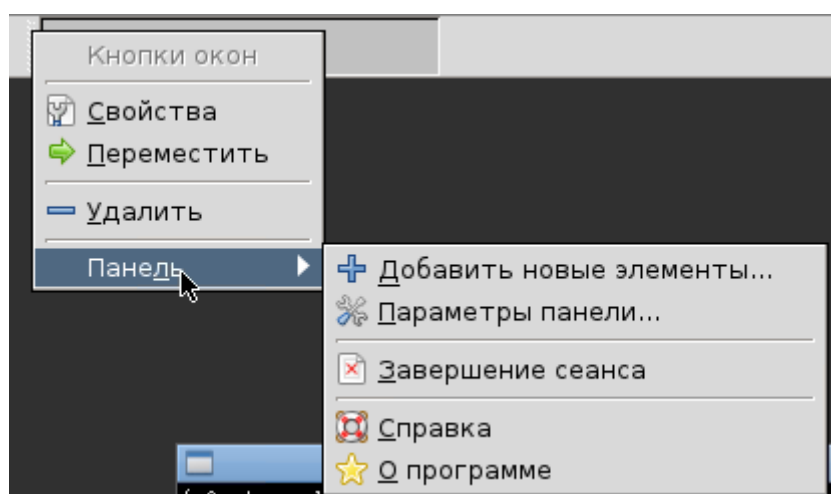


Рисунок 6 – Контекстное меню элемента

В этом меню нужно выбрать пункт «Параметры панели», после чего вокруг панели появится выделение и откроется окно настроек (рисунок 7).

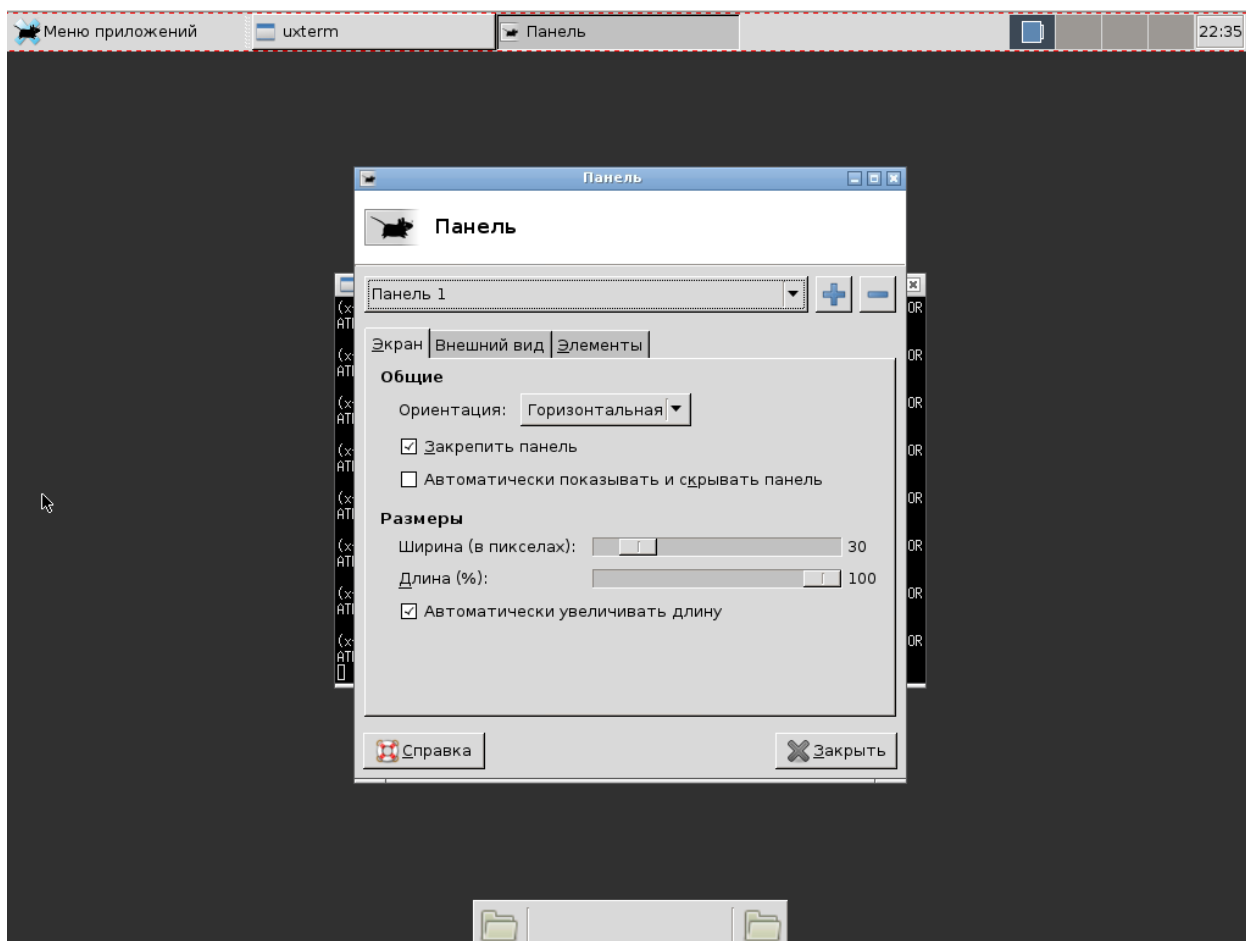


Рисунок 7 – Параметры панели

Над вкладками находится переключатель панелей (раскрывающийся список) и кнопки добавления/удаления панелей.

Вкладка «Экран» позволяет настроить такие опции, как ширину панели, возможность перетаскивать панель, ширину панели, автоматическое скрывание и т.д.

Вкладка «Внешний вид» содержит всего одну опцию, позволяющую выбрать стиль панели: системный, цвет или изображение.

Вкладка «Элементы» содержит список элементов, размещённых на панели. Их можно перемещать (клавиши с зелёными стрелками), настраивать, удалять и добавлять.

Настроим нижнюю панель следующим образом: постоянная ширина, возможность перетаскивания по экрану, автоскрывание.

Для настройки нижней панели нужно выбрать «Панель 2» в переключателе панелей. На вкладке «Экран» убрать отметку с пункта

«Закрепить панель» и поставить в «Автоматически показывать и скрывать панель», ширину установить 45 пикселей и длину 25%. На вкладке «Элементы» убрать все элементы, кроме «Показать рабочий стол», разделитель и «Каталог». Разделитель настроить следующим образом: стиль – прозрачный, установить отметку в пункте «Расширять». После закрытия окна настройки панель скроется, при наведении на неё курсора она развернётся. Теперь панель можно перетаскивать по всему экрану (захватив курсором одно из мест по краю панели, отмеченных точками), пример показан на рисунке 8.

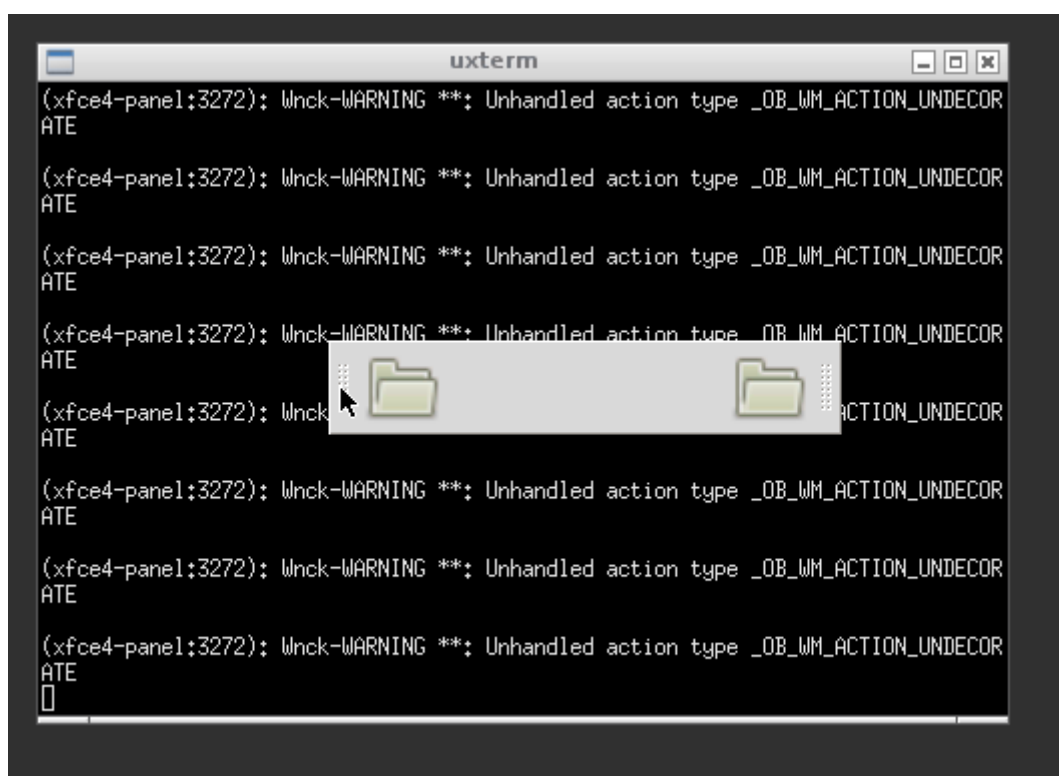


Рисунок 8 – Панель поверх окна терминала

Окно настроек панели также можно вызвать через меню: «Меню приложений» - «Настройки» - «Панель».

Настройка autostart

Как было отмечено, панель закрывается при закрытии окна терминала, из которого она была вызвана. Это можно исправить с помощью настроек автозапуска.

В этом пункте работы будет, как и в ЛР 2, использоваться системный скрипт autostart. Openbox может быть настроен для каждого пользователя отдельно, тогда файлы autostart, environment, rc.xml и menu.xml должны содержаться в домашнем каталоге каждого пользователя (~/.config/openbox/). При отсутствии этих файлов Openbox обращается к системным, хранящимся в папке /etc/xdg/openbox/. Эти файлы лучше оставлять неизменными или исправлять очень аккуратно, так как ошибка в них может вызвать негативные последствия. Особенно этот факт должны иметь в виду системные администраторы, многие программы работают по такому принципу.

Добавим автозапуск панели и терминала для всех пользователей. Откроем в leafpad файл autostart:

```
leafpad /etc/xdg/openbox/autostart
```

В конце файла нужно добавить 2 строки:

```
xfce4-panel &  
uxterm
```

Скрипт autostart срабатывает не всегда, а лишь в случае запуска сессии openbox (равносильно вызову openbox-session, команда openbox не должна запускать этот скрипт).

После этого необходимо выйти из Openbox (через контекстное меню) и снова запустить x-сервер командой startx через консоль.

Настройка рабочего стола

Для организации привычного пользователю Windows окружения не хватает, как минимум, одного важного элемента: иконок на рабочем столе. Для отображения иконок могут использоваться отдельные программы. В данной ЛР предлагается использовать idesk. Вторым элементом, который будет добавлен на рабочий стол, будет менеджер системных ресурсов conky.

Отображение иконок

В данном пункте работы понадобится файловый менеджер, для этого установим PCManFM:

```
apt-get install pcmanfm
```

Выполнение этой команды может занять достаточно много времени.

Файловый менеджер вызывается командой `pcmanfm`. Результат запуска показан на рисунке 9.

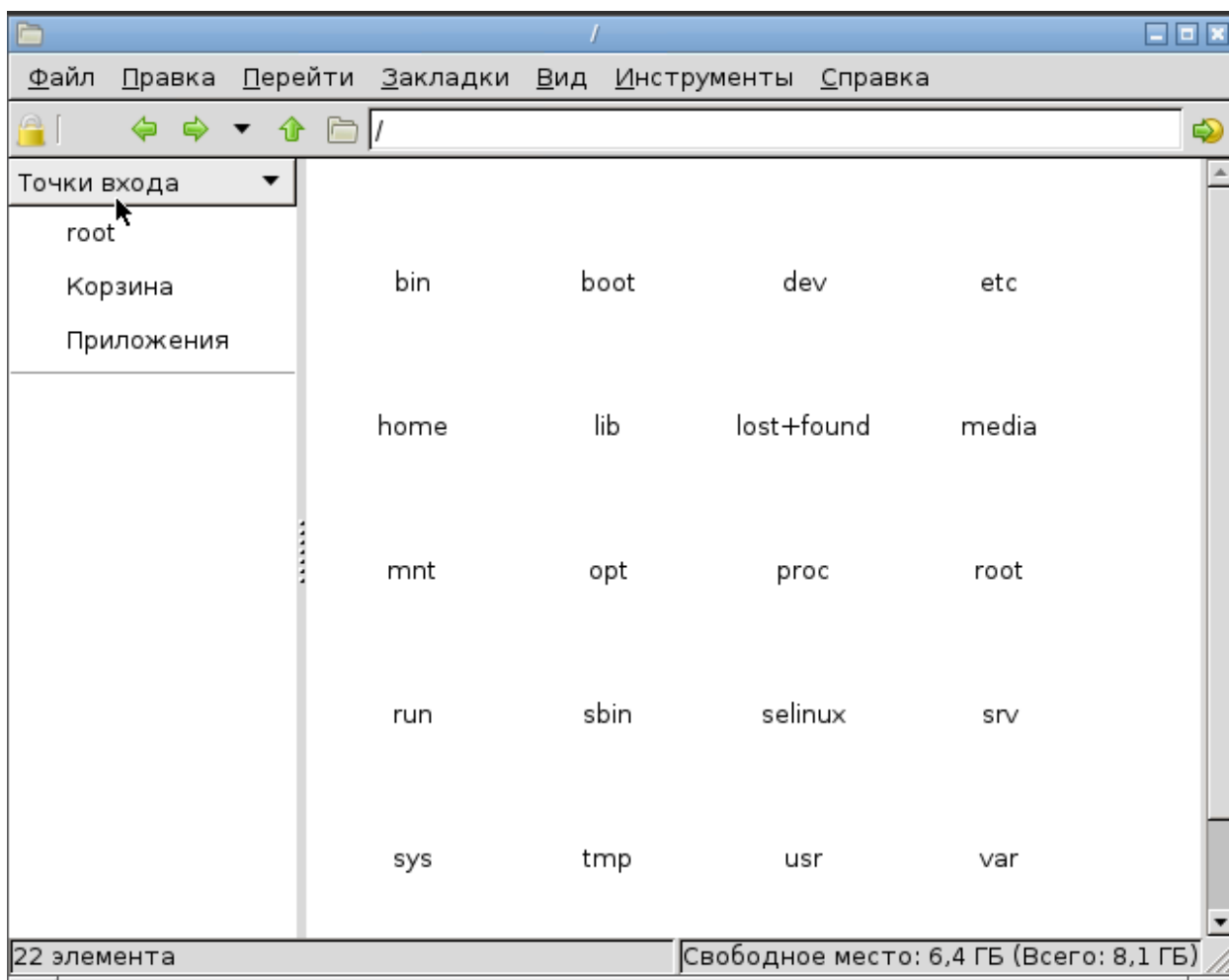


Рисунок 9 – PCManFM без иконок

Как можно заметить, в программе отсутствуют некоторые иконки. Эта проблема будет решена позднее. Обратите внимание, как изменилось меню Openbox (в меню «Debian»-«Приложения» появился пункт «Управление файлами»), «Меню приложений» панели xfce также изменилось.

Для отображения иконок нужно установить `idesk`:

```
apt-get install idesk
```

При попытке вызова будет выдана ошибка. Она объясняется отсутствием необходимой папки, её можно создать следующей командой:

```
mkdir ~/.idesktop
```

Результат повторного запуска idesk показан на рисунке 10.

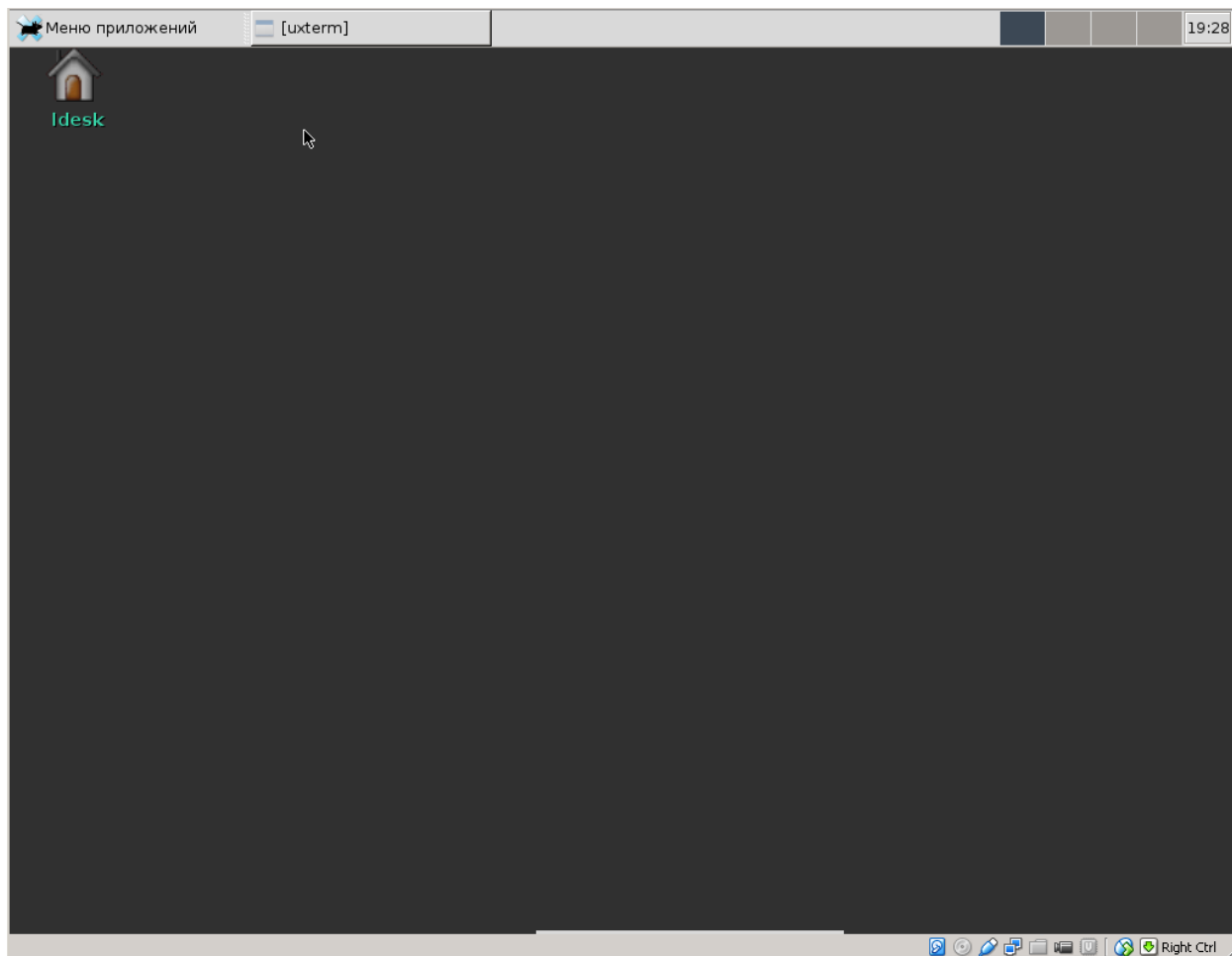


Рисунок 10 – Стандартная конфигурация idesk

Если закрыть окно терминала, из которого был запущен idesk, то он закроется. Необходимо исправить скрипт autorun. Теперь он должен выглядеть так:

```
xfce4-panel &  
uxterm &  
idesk
```

Теперь необходимо попробовать создать ярлык на рабочем столе. Ярлыки в idesk хранятся в виде файлов lnk. Для создания ярлыка (и открытия его для редактирования) для leafpad нужно выполнить следующую команду:

```
leafpad ~/.idesktop/leafpad.lnk
```

В данной команде используется особенность большинства редакторов в UNIX-системах: если файл существует, он открывается, если нет – он создаётся.

В файле необходимо набрать следующее:

```
table Icon
    Caption: Leafpad
    Command: /usr/bin/leafpad
    Icon: /usr/share/pixmaps/leafpad.png
    Width: 36
    Height: 36
    X: 100
    Y: 100
end
```

Это описание ярлыка размером 36x36, использующего иконку Leafpad, располагающегося на позиции (100;100). Для отображения иконки надо перезапустить idesk, выполнив в терминале:

```
killall idesk
idesk
```

Результат показан на рисунке 11. Следует заметить, что в таком случае родителем idesk снова станет окно терминала, потому более правильный путь – завершить x-сессию Openbox и перезапустить x-сервер.

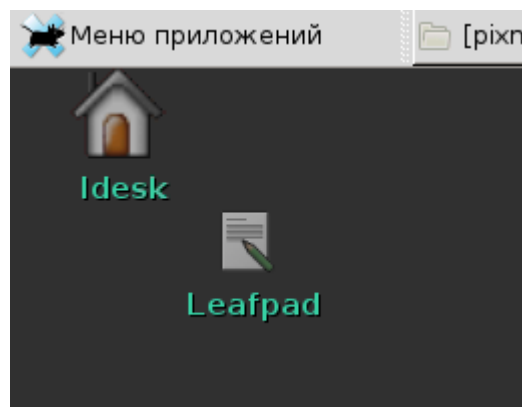


Рисунок 11 – Иконка Leafpad на рабочем столе

Преимуществом хранения ярлыков в таком формате является гибкая настройка. Возможно настроить не только размер и положение, но и реакцию на отдельные кнопки мыши. Так, `Command[0]` отвечает за реакцию на двойное нажатие левой кнопки мыши, `Command[1]` – за двойное нажатие правой. Поменяйте название ярлыка на `TextEditor` и сделайте так, чтобы при двойном щелчке левой клавишей мыши открывался `Leafpad`, при двойном щелчке правой – `vi` в новом окне `ixterm`.

Ярлыки по рабочему столу можно перемещать простым перетаскиванием, при этом их конфигурационные файлы изменяются автоматически.

Теперь необходимо настроить отображение иконок в `PCManFM`. Он использует библиотеку `GTK`, следующая команда скачает и установит иконки, которые будут использоваться всеми `GTK`-приложениями:

```
apt-get install tango-icon-theme
```

Для того, чтобы именно эти иконки использовались в `GTK`-приложениях, необходимо создать для них конфигурационный файл:

```
leafpad ~/.gtkrc-2.0
```

В этом файле необходимо прописать следующую строку:

```
gtk-icon-theme-name="Tango"
```

Теперь `PCManFM` должен выглядеть так, как показано на рисунке 12.

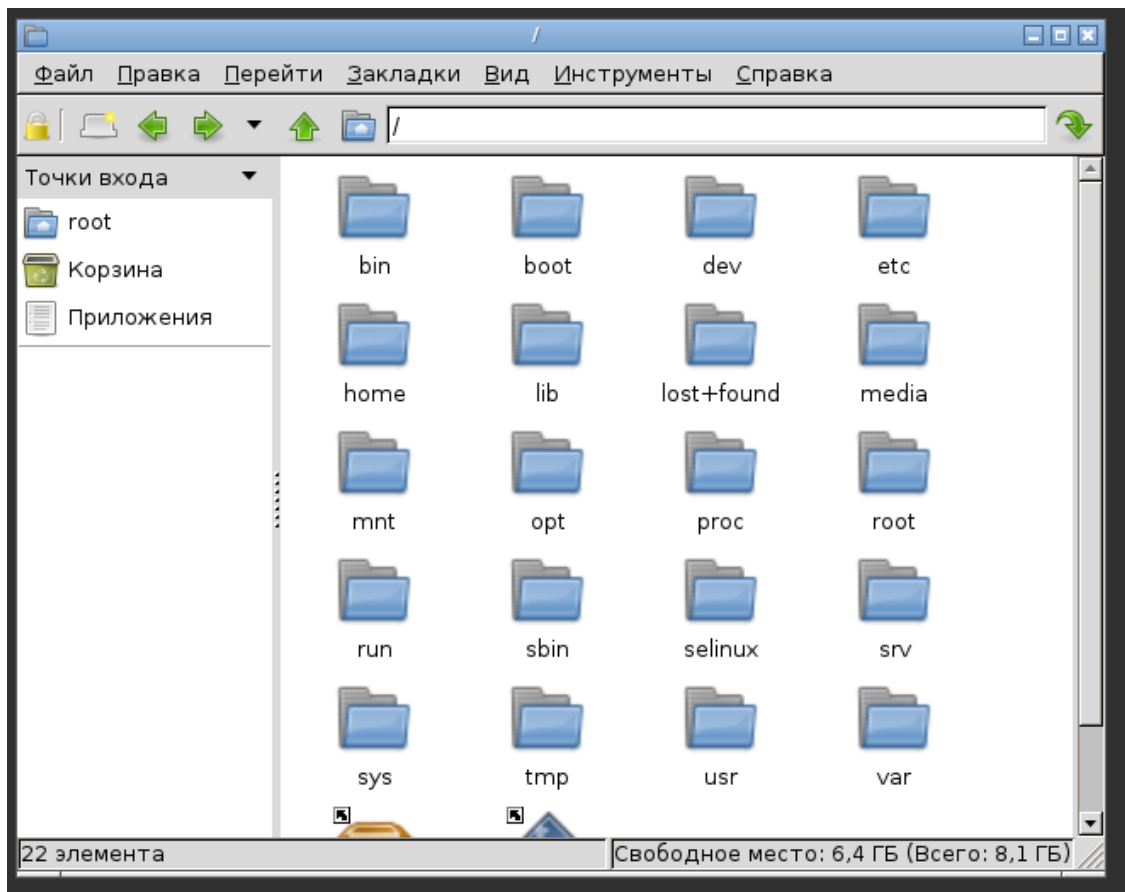


Рисунок 12 – PCManFM с иконками Tango

Также можно обратить внимание на то, как изменились некоторые иконки в панели (например, «Скрыть окна»).

Установка и настройка conky

conky представляет собой менеджер системных ресурсов. Это гибкая утилита, позволяющая отслеживать такие параметры, как использование ОЗУ и диска, нагрузку на ЦП. Однако conky может отображать даже новости и погоду.

Для установки conky надо выполнить:

```
apt-get install conky
```

После установки и запуска (командой `conky`) на экране отобразится окно conky, перекрывающее иконки на рабочем столе (рисунок 13).

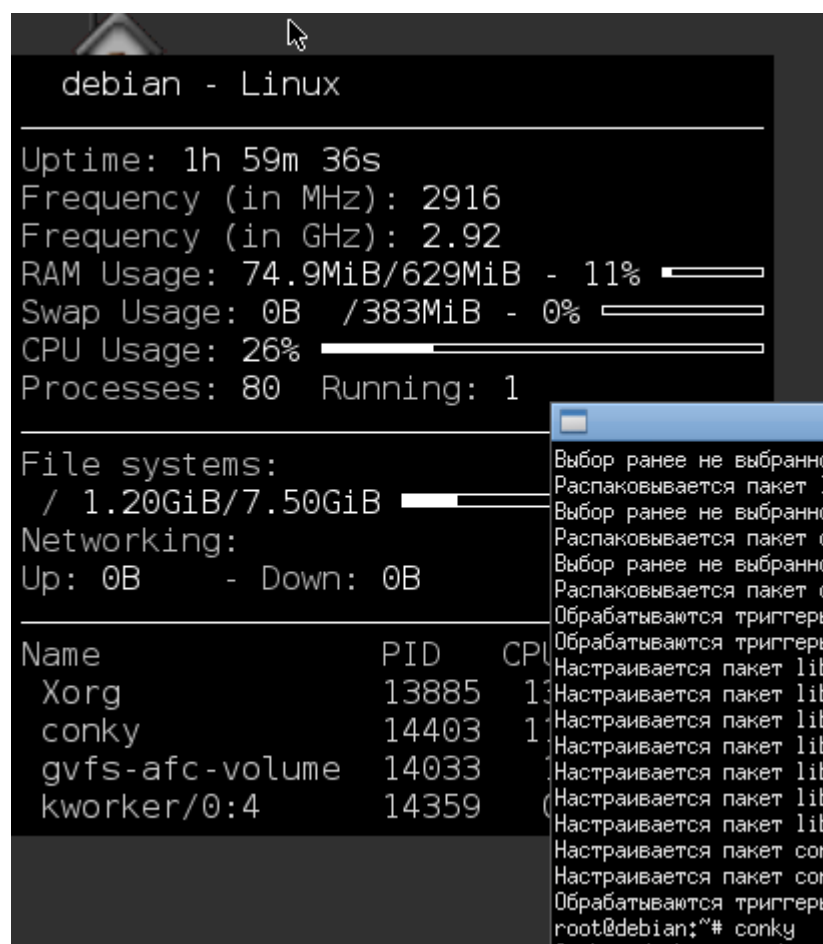


Рисунок 14 – conky

Обратите внимание, что при закрытии окна терминала, из которого был вызван conky, сам conky не закроется. Это нужно сделать с помощью `killall conky` или аналогичным способом.

Для настройки conky необходимо отредактировать его конфигурационный файл:

```
leafpad /etc/conky/conky.conf
```

Как и в случае с настройкой Openbox, это общий файл. Пользовательские файлы настроек conky должны лежать по адресу ~/.conkyrc.

Необходимо изменить положение conky. Для этого замените строку

```
alignment top_left
```

на

```
alignment top_right
```

В секции Text отображается текст, выводимый на панель conky. Следующая строка добавит в conky отображение имени пользователя в системе:

```
{color grey}Current user: $USER
```

На рисунке 15 показан фрагмент конфигурационного файла, отображающий, куда необходимо вставить эту строку.

```
TEXT
${scroll 16 $nodename - $sysname $kernel on $machine | }
${color grey}Current user: $USER
$hr
```

Рисунок 15 – Редактирование конфигурации conky

После настройки снова запустите conky, результат показан на рисунке 16.

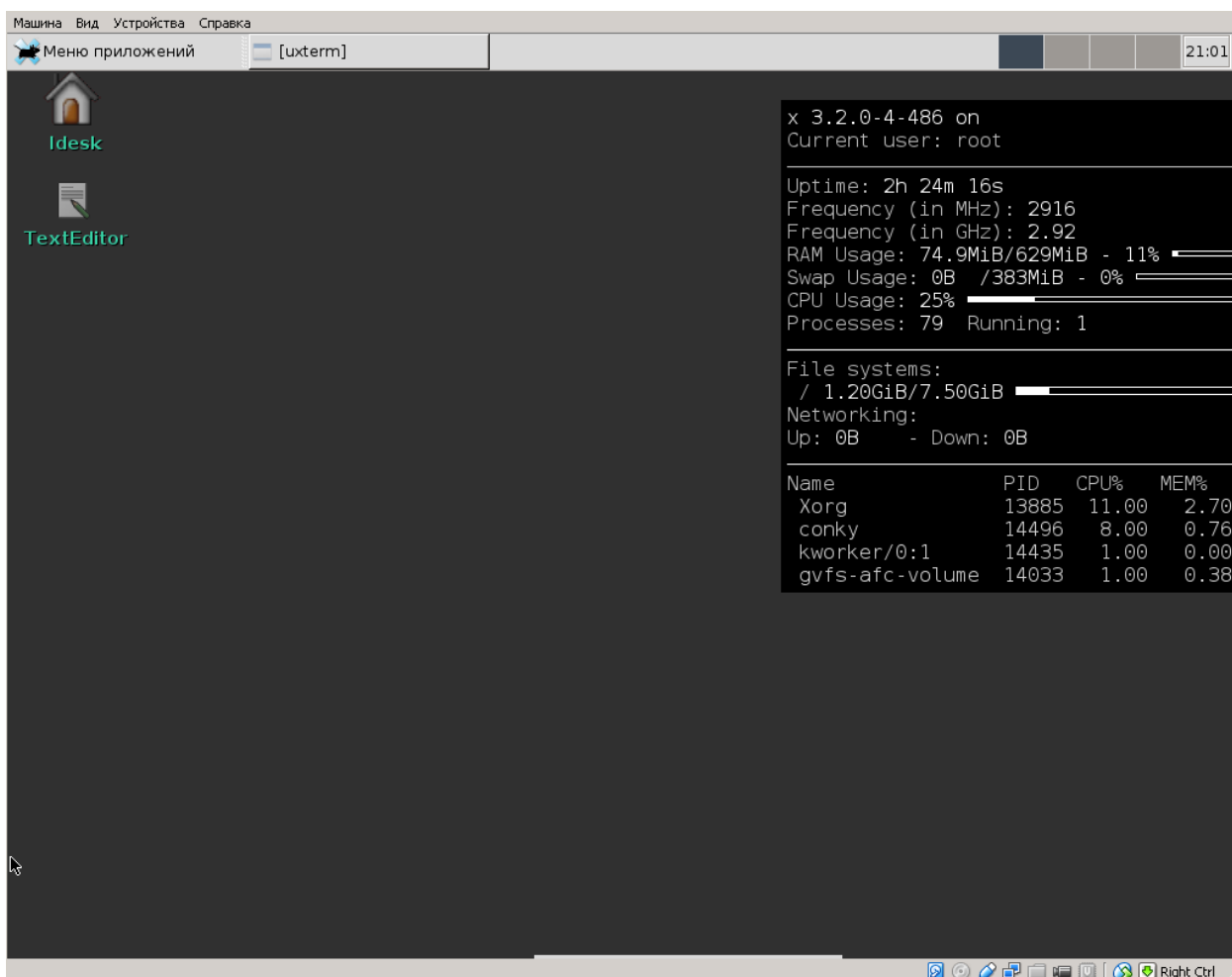


Рисунок 16 – Openbox, xfce4-panel, idesk и conky

Наконец, необходимо отредактировать скрипт автозапуска, добавив в него conky, теперь скрипт должен выглядеть следующим образом:

```
xfce4-panel &  
  
uxterm &  
  
idesk &  
  
conky
```

Можно настроить отображение фоновой картинки в качестве обоев. Для этого вместе с Openbox часто используют программу feh. Установить её можно следующей командой:

```
apt-get install feh
```

Для файла bmstu.jpeg установка изображения в качестве фона может выглядеть следующим образом:

```
feh --bg-scale ~/bmstu.jpeg
```

Результат показан на рисунке 17.

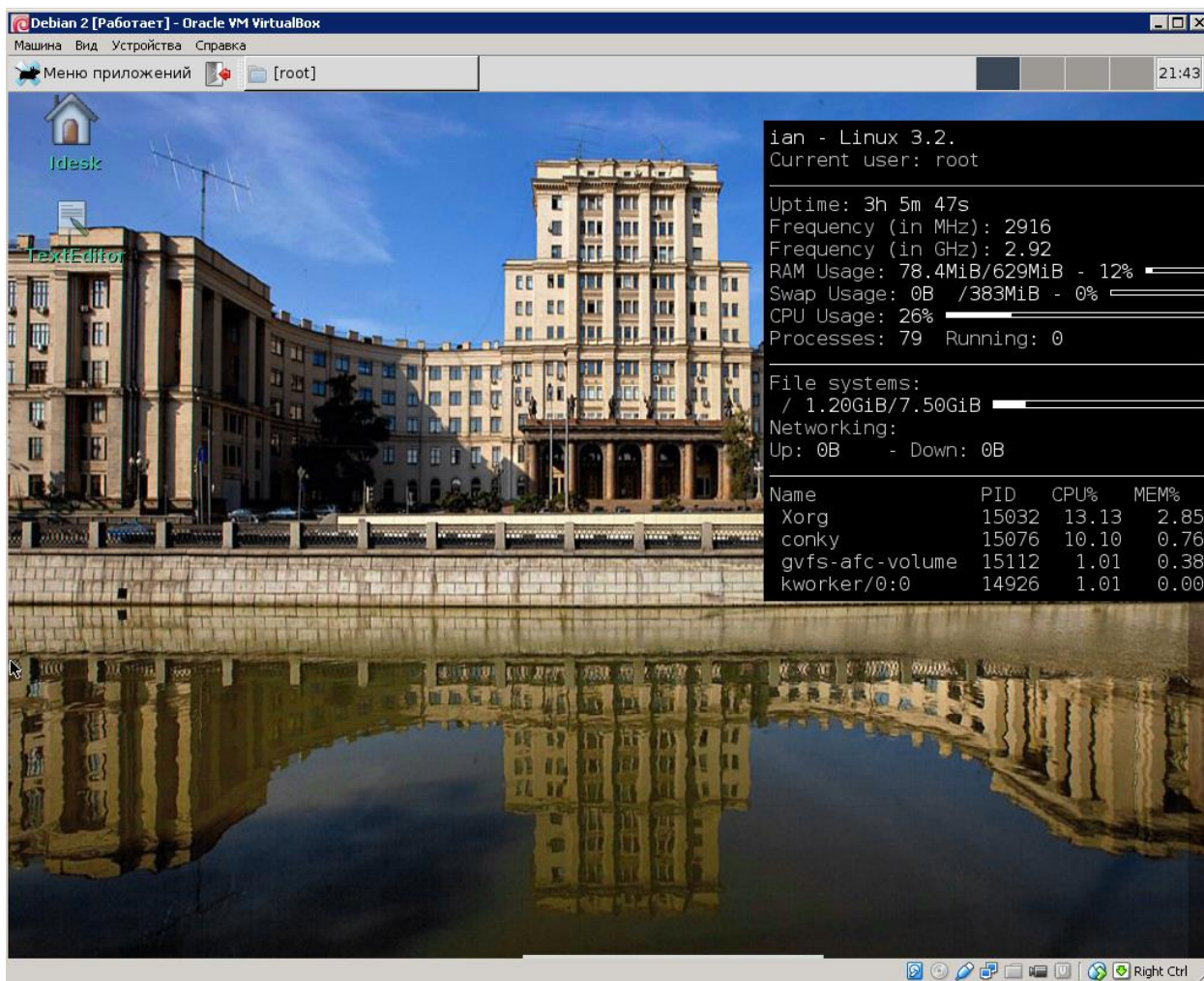


Рисунок 17 – Фоновое изображение в Openbox

Полную инструкцию по `feh` можно получить с помощью `man`.

Установите в свой фоновый рисунок и отредактируйте скрипт `autostart` так, чтобы фоновое изображение устанавливалось каждый раз.

Возможно изменение разрешения экрана без перезапуска X-сервера. Для просмотра поддерживаемых разрешений необходимо выполнить следующую команду:

```
xrandr
```

Чтобы изменить разрешение, нужно выполнить команду:

```
xrandr -s WxH
```

Где *W* – ширина, *H* – высота в пикселах.

Управление с помощью консоли

В данной (и предыдущих) ЛР активно использовались консольные команды и утилиты. Консоль в UNIX-подобных системах, как правило, играет заметную роль. Считается, что в хорошо продуманной UNIX-подобной системе все основные действия должны быть выполнимы через консоль, без запуска графической оболочки. В данной части ЛР будут описаны основные команды утилиты, способы их соединения друг с другом, будут даны основы для самостоятельного написания bash-скриптов.

Основные команды и утилиты

В этой секции будут приведены команды и утилиты, присутствующие во всех POSIX-совместимых системах и в основанных на Debian дистрибутивах. К каждой команде будут даны комментарии. Следует отметить, что употребление терминов «команды», «утилиты» и «программы» весьма условно и чётко не определено: так, кто-то может называть `grep` командой, кто-то – консольной утилитой.

Основные утилиты

Ниже приведены основные утилиты, которые могут быть использованы в данной ЛР, и основная информация об используемых ключах. Более подробную информацию можно получить с помощью `man`.

`touch` – создание пустого файла. Принимает в качестве параметра путь и имя создаваемого файла, например:

```
touch /root/file.txt
```

`grep` – гибкая утилита поиска. Используется совместно с регулярными выражениями. Следующий пример показывает вывод из файла строк, в которых встречаются слова и сочетания слов, начинающиеся на «х» и оканчивающиеся на «е» (т.е. если в строке есть «е» после «х»):

```
grep x.*e /etc/xdg/openbox/autostart
```


Регулярные выражения – большая тема, в данной ЛР они использоваться не будут.

`mkdir` – создать новую папку.

`cp` – копирование файла из одного каталога в другой.

`find` – поиск файлов по имени, поддерживает регулярные выражения.

`wc` – подсчёт слов. Может напечатать количество слов (`-w`), длину максимальной строки (`-L`), количество символов (`-m`).

`ls` – вывод списка каталогов.

`man` – страница справочного руководства.

`clear` – очистка окна консоли.

`echo` – вывод на консоль.

`apt-get install` – установка пакета (Debian).

`apt-get install -y` – установка пакета, автоматический ответ на все вопросы «да» (Debian).

`apt-get install -d` – только загрузка пакета (Debian).

`apt-get update` – обновление индекса пакетов (Debian).

`apt-get upgrade` – обновление пакетов (Debian).

`apt-get dist-upgrade` – обновление всего дистрибутива (Debian).

Зачастую команда аналогична предыдущей, различие – в поведении при изменении отношений между установленными пакетами.

`apt-get remove` – удаление пакета (Debian).

`apt-cache search` – поиск пакета (Debian).

Объединение команд

Возможно несколько способов объединения команд:

- По логическому «И» (`&&`). Вторая команда выполняется только в случае успешного выполнения первой и т.д. В Debian часто используют следующее сочетание:

`apt-get update && apt-get upgrade`

- По логическому «ИЛИ» (||). Вторая команда выполняется в случае, если первая не завершилась успешно.

- Последовательно (;). Команды исполняются в любом случае.

- Вторая команда использует результаты вывода первой с помощью канала (|). Например, следующая команда посчитает, сколько папок каталога / начинаются с l:

```
ls / | grep ^l | wc -l
```

Более сложные утилиты

В данной части работы будут представлены несколько утилит, которые могут быть полезны при администрировании Linux.

top, ps, pstree

Перечисленные в заголовке утилиты являются мониторами процессов.

ps – по умолчанию выводит только запущенные пользователем процессы. Некоторые ключи, используемые этой утилитой:

- a – процессы, связанные с конкретным терминалом (но без системных).

- e, -A – вывод всех процессов в системе.

- f – расширенный вывод.

- T – процессы на определённом терминале.

- r – только активные процессы.

- o – выводить только определённые столбцы.

По умолчанию выводятся следующие столбцы:

PID – идентификатор процесса (process id).

TTY – терминал, с которым связан процесс.

TIME – сколько процессорного времени занимает процесс.

CMD – команда, запустившая процесс.

С использованием ключа -f выводятся следующие поля:

UID – идентификатор пользователя, вызвавшего процесс.

PID – см.выше.

PPID – PID родителя (parent PID).

C – приоритет, используемый планировщиком для данного процесса.

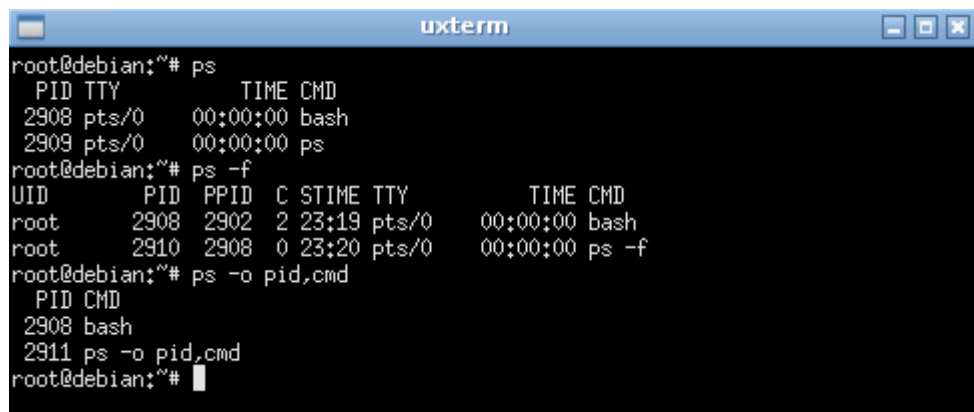
STIME – время старта процесса.

TTY – см.выше.

TIME – см.выше.

CMD – см.выше.

На рисунке 18 показаны примеры использования утилиты ps.



```
root@debian:~# ps
  PID TTY          TIME CMD
 2908 pts/0        00:00:00 bash
 2909 pts/0        00:00:00 ps
root@debian:~# ps -f
UID          PID  PPID  C  STIME TTY          TIME CMD
root         2908   2902  2  23:19 pts/0        00:00:00 bash
root         2910   2908  0  23:20 pts/0        00:00:00 ps -f
root@debian:~# ps -o pid,cmd
  PID CMD
 2908 bash
 2911 ps -o pid,cmd
root@debian:~#
```

Рисунок 18 – Вывод утилиты ps

pstree – очень похожая на ps утилита. В отличие от ps, pstree отображает процессы в виде дерева. Некоторые ключи, используемые с pstree:

-h – подсвечивает текущий процесс и его предков. Не все терминалы поддерживают подсветку.

-l – показывать длинные строки.

-n – сортировать процессы с одинаковым предком по PID (числовая сортировка), а не по имени.

-u – отображать UID.

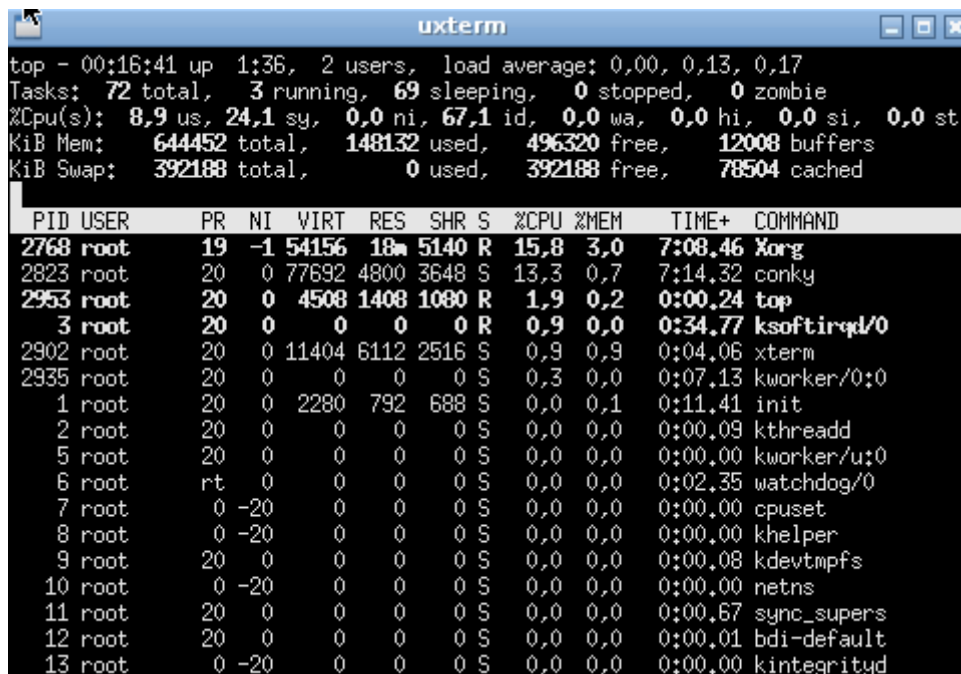
На рисунке 19 показан пример использования утилиты pstree.

%CPU – процент использования ЦП.

%MEM – процент использования ОЗУ.

TIME+ - общее время активности процесса.

На рисунке 20 показан пример работы top.



```
top - 00:16:41 up 1:36, 2 users, load average: 0,00, 0,13, 0,17
Tasks: 72 total, 3 running, 69 sleeping, 0 stopped, 0 zombie
%Cpu(s): 8,9 us, 24,1 sy, 0,0 ni, 67,1 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
KiB Mem: 644452 total, 148132 used, 496320 free, 12008 buffers
KiB Swap: 392188 total, 0 used, 392188 free, 78504 cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2768	root	19	-1	54156	18m	5140	R	15,8	3,0	7:08.46	Xorg
2823	root	20	0	77692	4800	3648	S	13,3	0,7	7:14.32	conky
2953	root	20	0	4508	1408	1080	R	1,9	0,2	0:00.24	top
3	root	20	0	0	0	0	R	0,9	0,0	0:34.77	ksoftirqd/0
2902	root	20	0	11404	6112	2516	S	0,9	0,9	0:04.06	xterm
2935	root	20	0	0	0	0	S	0,3	0,0	0:07.13	kworker/0:0
1	root	20	0	2280	792	688	S	0,0	0,1	0:11.41	init
2	root	20	0	0	0	0	S	0,0	0,0	0:00.09	kthreadd
5	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kworker/u:0
6	root	rt	0	0	0	0	S	0,0	0,0	0:02.35	watchdog/0
7	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	cpuset
8	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	khelper
9	root	20	0	0	0	0	S	0,0	0,0	0:00.08	kdevtmpfs
10	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	netns
11	root	20	0	0	0	0	S	0,0	0,0	0:00.67	sync_supers
12	root	20	0	0	0	0	S	0,0	0,0	0:00.01	bdi-default
13	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	kintegrityd

Рисунок 20 – Утилита top

Перед таблицей процессов выводится системная информация. Если запустить top в маленьком окне консоли, а затем развернуть его на весь экран, top заполнит собой весь экран.

ping, netstat, ip

ping – простейшая утилита, позволяющая проверить соединение в TCP/IP сетях. Некоторые ключи:

- i – интервал (секунд).
- c – количество передаваемых запросов.
- w – завершить работу по истечении времени (секунд).
- s – размер передаваемых пакетов (байт).

netstat – утилита, предназначенная для просмотра подробного состояния портов и связанной с ними информации. Некоторые ключи:

- a – вывод информации обо всех сокетах.

-r – показать статистику маршрутизации.

-p – показывать имя процесса для сокета.

ip – показ и управление маршрутизацией и сетевыми интерфейсами

Linux. Имеет следующий синтаксис:

```
ip [ОПЦИИ] ОБЪЕКТ {КОМАНДА | help}
```

Где ОБЪЕКТ (неполный список):

link- сетевой интерфейс.

address- адрес на интерфейсе в соответствии с протоколом.

route- запись таблицы маршрутизации.

rule- правило маршрутизации.

tunnel- туннель IP.

Написание скриптов на bash

Как было показано ранее, консоль Linux – мощное средство для администрирования. В целях автоматизации были созданы скрипты. Скрипт теоретически может быть написан на любом языке, для которого в системе существует интерпретатор. Обычно скрипты пишут на bash или awk.

В виду краткости курса в данной части ЛР будут даны только основы bash, необходимая дополнительная литература находится в свободном доступе в сети Интернет.

Простейший скрипт – просто набор команд для исполнения.

Скрипты могут начинаться со следующей строчки:

```
#!/usr/bin/env bash
```

Эта строка показывает, какая программа должна выполнить скрипт.

Скрипт либо можно сделать исполняемым (чтобы его можно было исполнять из консоли как отдельную команду), либо запускать его с помощью команды bash.

Пример скрипта:

```
#!/usr/bin/env bash
```

```
echo "Hello! My parent is:" $PPID
```

Это очень простой скрипт, он выводит приветствие и PID своего родителя (\$PPID – встроенная переменная). Наберите текст этого скрипта и сохраните его под именем `myscript` в каталоге `/root`. Попробуйте запустить его:

```
bash /root/myscript
```

Или просто:

```
bash myscript
```

Чтобы скрипт сделать запускаемым без команды `bash`, необходимо изменить его свойства:

```
chmod +rx myscript
```

Эта команда позволяет всем пользователям читать и исполнять файл скрипта. Теперь го запуск может выглядеть так:

```
/root/myscript
```

`bash` является достаточно мощным языком программирования, в нём есть массивы, циклы, порождение процессов, перенаправление потоков ввода/вывода и другие возможности. Цель этой краткой секции – показать дополнительные возможности работы с консолью.

Выводы

Для UNIX-подобных систем существует множество окружений рабочего стола. Окружение рабочего стола дружелюбно к пользователю и содержит множество дополнительных программ. Как правило, эти оболочки достаточно гибки в настройке, некоторые элементы одних могут использоваться в других. DE часто имеют различную идеологию: некоторые из них предлагают базовый (но комфортный для работы) набор программ, минимум визуальных эффектов, в то время как другие предлагают максимум программ (мультимедиа, офис, Интернет). Каждый пользователь, попробовавший несколько DE и дистрибутивов, как правило, самостоятельно определяет для себя самый подходящий вариант.

Некоторые пользователи предпочитают облегчённый интерфейс, тогда они ставят не DE, а только WM, дополняя его необходимыми компонентами. В ходе данной ЛР была описана схема настройки Openbox и дополнительных программ, позволивших получить рабочее место в привычном для пользователя Windows стиле. Полученные знания так же помогают лучше понять состав DE.

В последней части работы были получены теоретические сведения о некоторых утилитах Linux. Данные утилиты могут быть полезны для начала работы с Linux в качестве администратора, однако формат и направленность лабораторной работы не позволяет описывать их подробно. Подробную информацию о работе предложенных утилит можно получить с помощью man или из открытых источников. Были изложены общие сведения по скриптовым языкам в общем и bash в частности. Этих сведений недостаточно, чтобы начать разработку сложных скриптов, но достаточно для общего представления о возможностях bash.

Порядок выполнения работы.

1. Ознакомиться с теоретическими сведениями по среде управления.
2. Изучить и использовать консоль, утилиты и команды среды управления.
3. Пройти тестирование по теоретическому материалу.

Отчет должен включать:

- название работы и ее цель;
- описание команд и результаты работы;