Федеральное государственное бюджетное образовательное учреждение

высшего образования

|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | ***«Московский государственный технический университет  имени Н.Э. Баумана (национальный исследовательский университет)»***  ***(МГТУ им. Н. Э. Баумана)*** |

Факультет ***Информатика и системы управления***

Кафедра ***Компьютерные системы и сети (ИУ6)***

**Отчет**

**по лабораторной работе №3**

**Дисциплина: Операционные системы**

**Название лабораторной работы: Исследование среды управления Linux**

Студент гр. ИУ6 - 54 **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_**А.А. Шумаков**\_\_\_\_**

(Подпись, дата) (И.О. Фамилия)

Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_**В.Ю.Мельников\_**\_**

(Подпись, дата) (И.О. Фамилия)

Москва, 2018

***Цель лабораторной работы:*** исследование среды управления Linux.

**Выполнение работы**

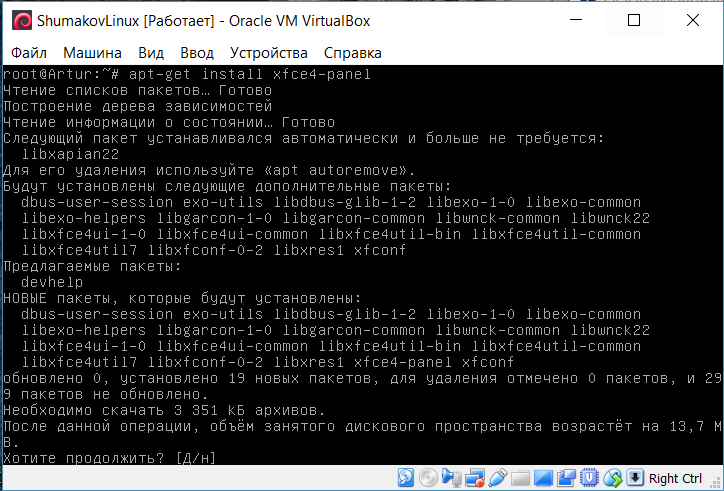
**Настройка Openbox**

*Openbox* – популярный оконный менеджер, который используется как с *DE* (*Desktop Environments*), так и отдельно. В данной части лабораторной работы произведем настройку *Openbox* и некоторых прикладных программ, которые позволяют организовать рабочее пространство. Для начала работы войдем в систему как суперпользователь (логин: *root*).

**Установка и настройка панелей**

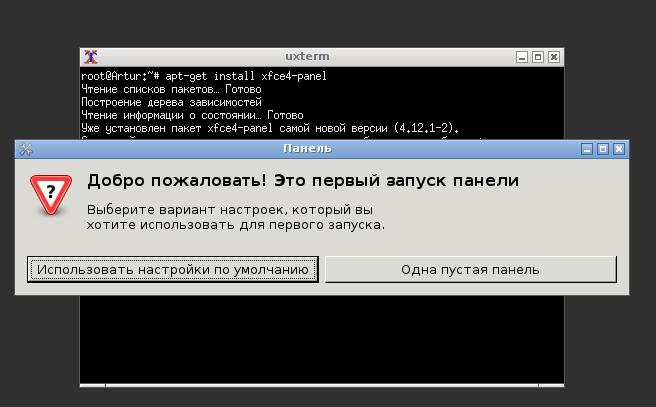
По умолчанию *Openbox* предоставляет пользователю лишь рабочий стол с контекстным меню. Для получения более привычного способа управления нужно установить панели. Среди самых популярных панелей для *Openbox* можно выделить *xfce4-panel*, *lxpanel* и *tint2*. Первые две входят в состав *Xfce* и *LXDE*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

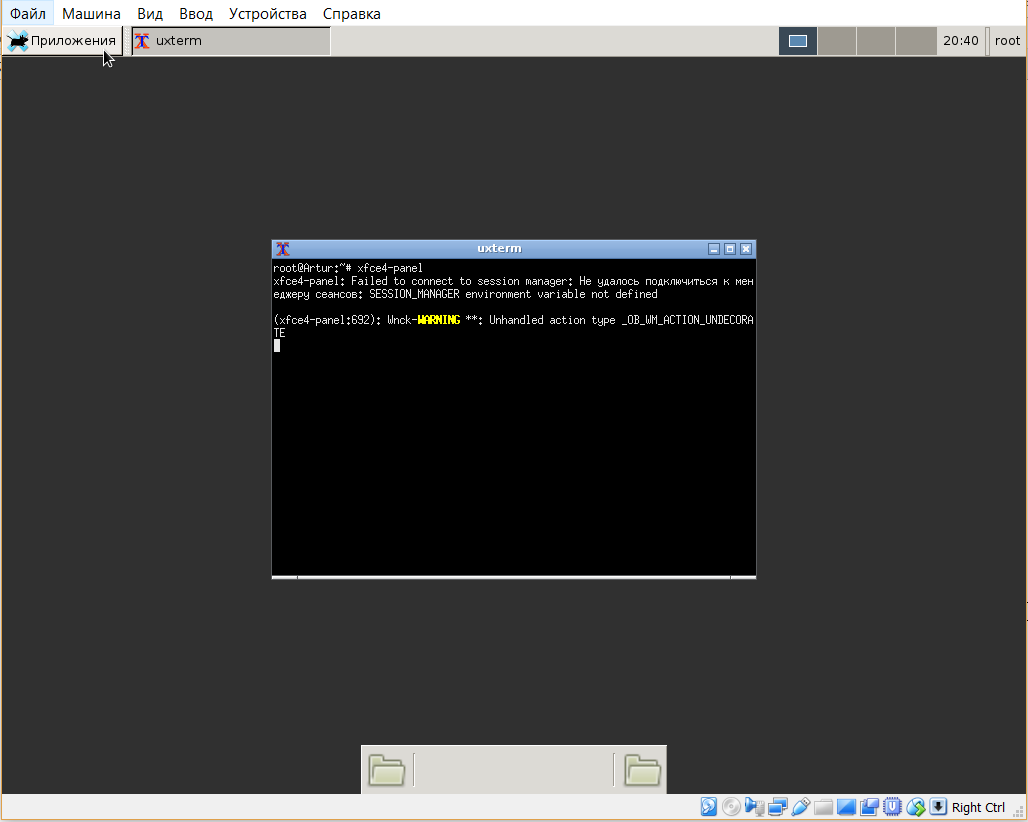
Для установки *xfce4-panel* выполним в терминале ОС команду:   
*apt-get install xfce4-panel*



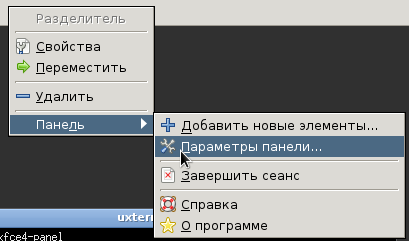
Для запуска панели наберем в терминале: *xfce4-panel*

При первом запуске панели будет предложен выбор варианта настройки. Для простоты выбираем *Использовать настройки по умолчанию*. В результате в верхней части экрана появилась панель.

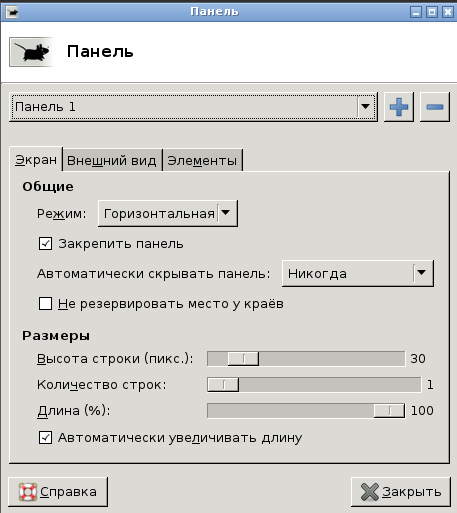




Теперь приступим к настройке панели для чего нажмем *ПКМ* по панели и выбирем пункт меню *Панель->Параметры панели*

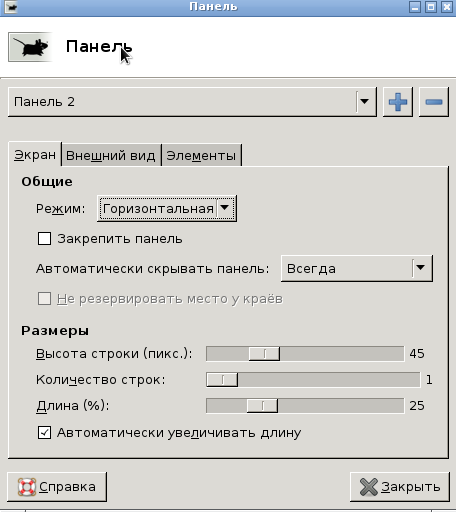


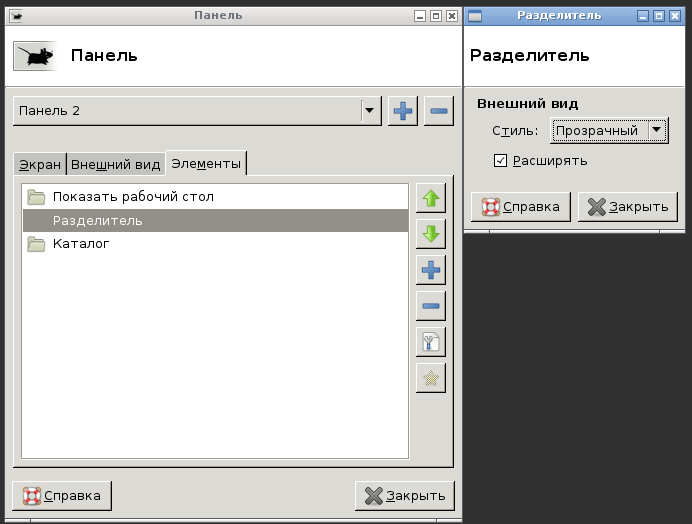
Открылось окно настройки панели.

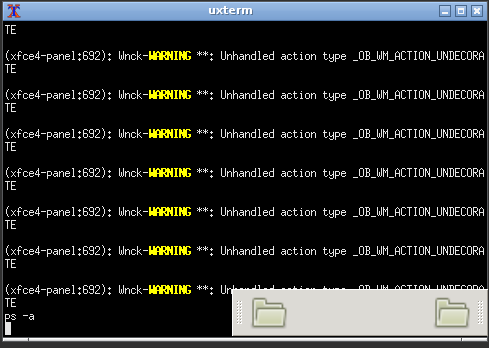


Настроим нижнюю панель следующим образом: постоянная ширина, возможность перетаскивания по экрану, автоскрытие.

Для настройки нижней панели выбираем «Панель 2» в переключателе панелей. На вкладке «Экран» уберем отметку с пункта «Закрепить панель» и поставим в «Автоматически показывать и скрывать панель», ширину установим 45 пикселей и длину 25%. На вкладке «Элементы» уберем все элементы, кроме «Показать рабочий стол», разделитель и «Каталог». Разделитель настроим следующим образом: стиль – прозрачный, установим 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).







**Настройка autostart**

Если мы закроем окно терминала, из которого мы вызвали панель, то панель закроется. Это можно исправить с помощью настроек автозапуска.

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

*leafpad /etc/xdg/openbox/autostart*

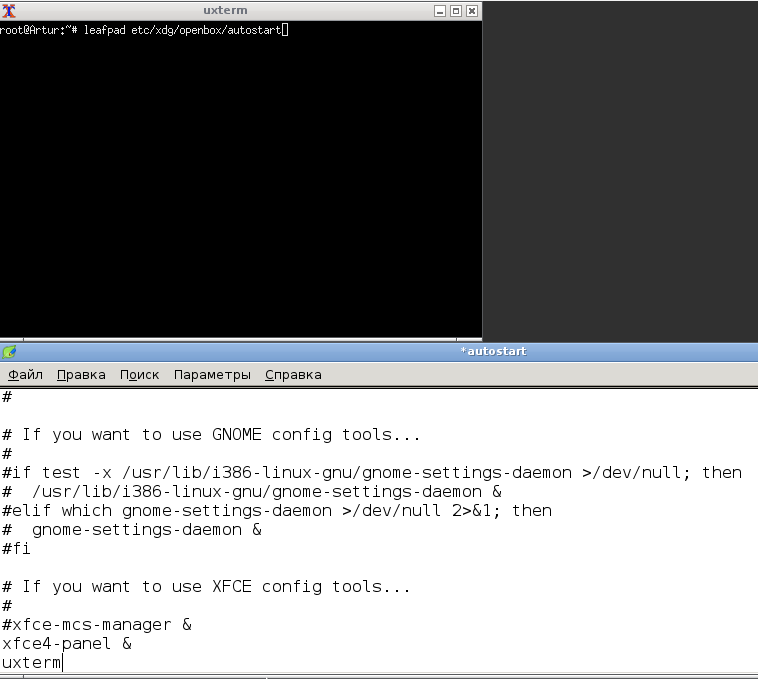
В конце файла нужно добавить 2 строки:

*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 выйдем из *Openbox* (через контекстное меню) и снова запустим x-сервер командой *startx* через консоль.

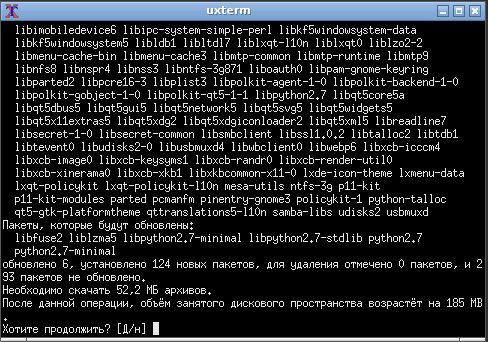


**Настройка рабочего стола**

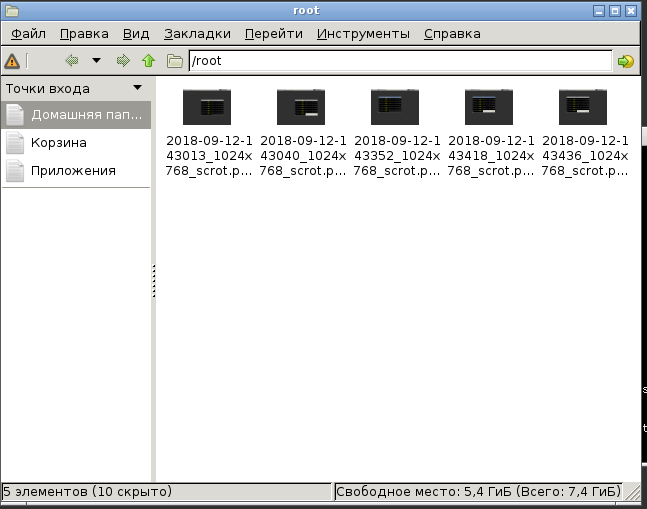
Для организации привычного пользователю *Windows* окружения не хватает, как минимум, одного важного элемента: иконок на рабочем столе. Для отображения иконок могут использоваться отдельные программы.

**Отображение иконок**

Для отображения иконок требуется установить файловый менеджер *PCManFM* командой: *apt-get install pcmanfm*

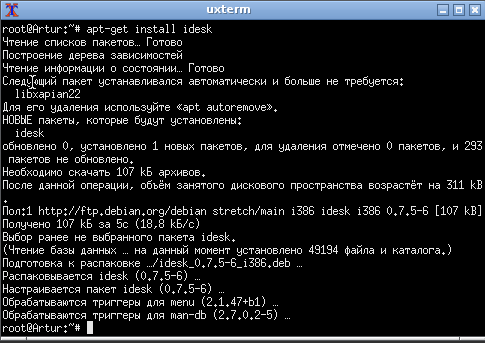


Файловый менеджер вызывается командой *pcmanfm*.



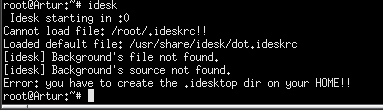
Как можно заметить, в программе отсутствуют некоторые иконки. Эта проблема будет решена позднее. Обратите внимание, как изменилось меню *Openbox* (в меню *«Debian»-«Приложения»* появился пункт *«Управление файлами»*), *«Меню приложений»* панели *xfce* также изменилось.

Для отображения иконок установим *idesk* : *apt-get install idesk*

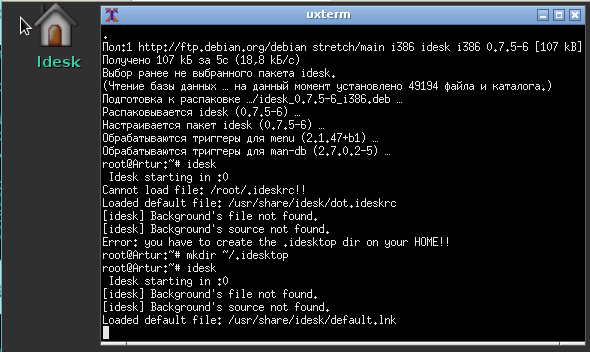


При попытке вызова будет выдана ошибка. Она объясняется отсутствием необходимой папки, её можно создать следующей командой:

*mkdir ~/.idesktop*



В результате повторного ввода команды *idesk* появится иконка на рабочем столе.

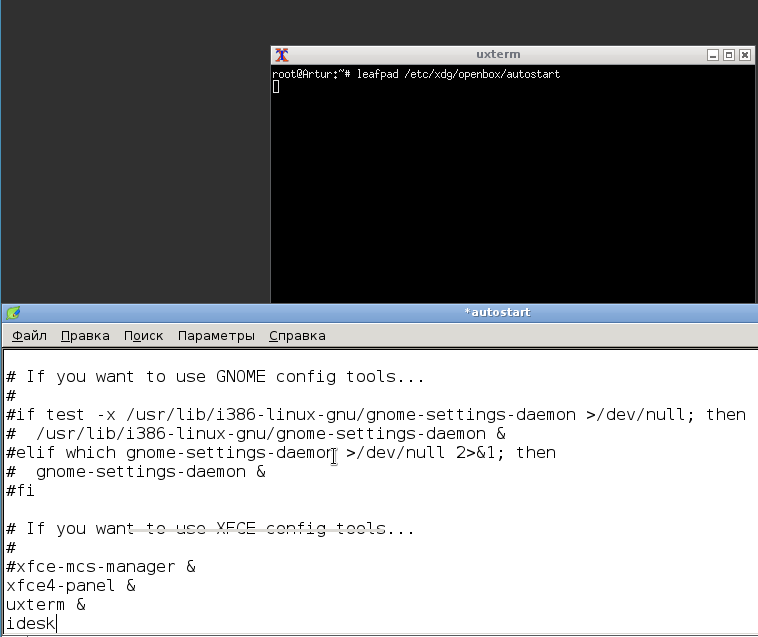


Если закрыть окно терминала, из которого был запущен *idesk*, то он закроется. Необходимо исправить скрипт *autorun*. Теперь он должен выглядеть так:

*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 набираем следующее:

*table Icon*

*Caption: Leafpad*

*Command: /usr/bin/leafpad*

*Icon: /usr/share/pixmaps/leafpad.png*

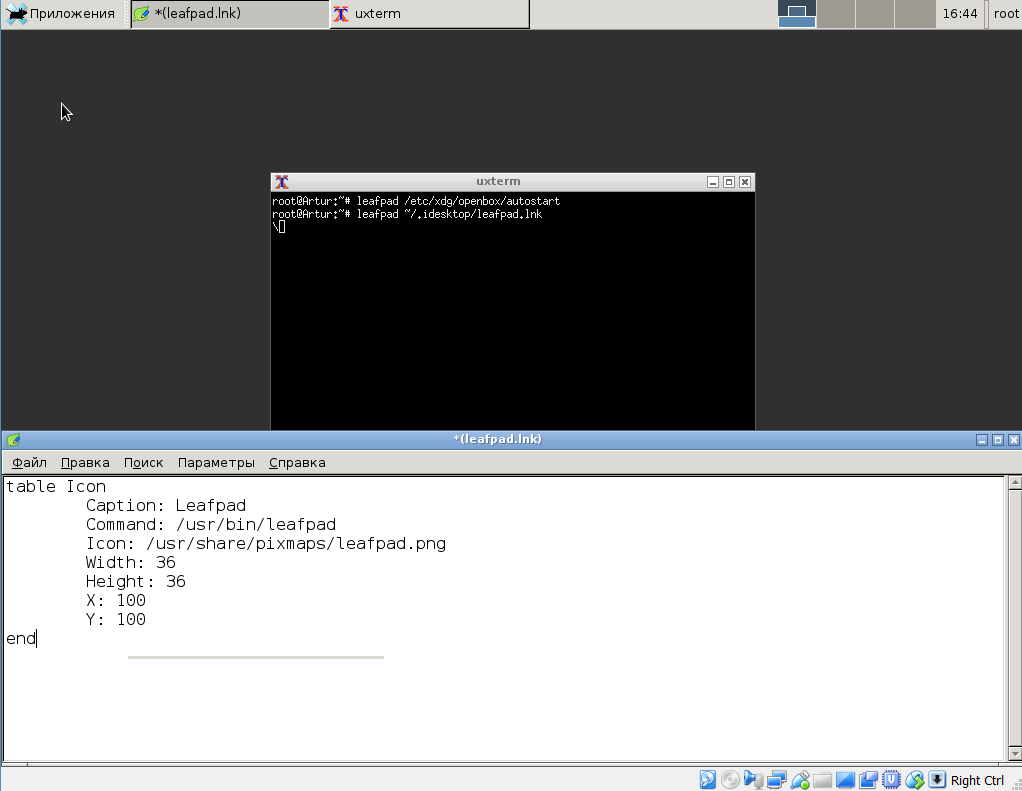
*Width: 36*

*Height: 36*

*X: 100*

*Y: 100*

*end*



Это описание ярлыка размером 36х36, использующего иконку *Leafpad*, располагающегося на позиции (100;100). Для отображения иконки надо перезапустить *idesk*, выполнив в терминале:

*killall idesk*

*idesk*



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. Поменяем название ярлыка на *TextEditor* и сделаем так, чтобы при двойном щелчке левой клавишей мыши открывался *Leafpad*, при двойном щелчке правой – *vi* в новом окне *uxterm*.

Для этого выполняем последовательно следующие действия:

Редактируем ярлык *Leafpad*, выполнив команду: *leafpad ~/.idesktop/leafpad.lnk*

В файле набираем следующее:

*Table Icon*

*Caption: TextEditor*

*Icon: /usr/share/pixmaps/leafpad.png*

*Width: 36*

*Height: 36*

*X: 100*

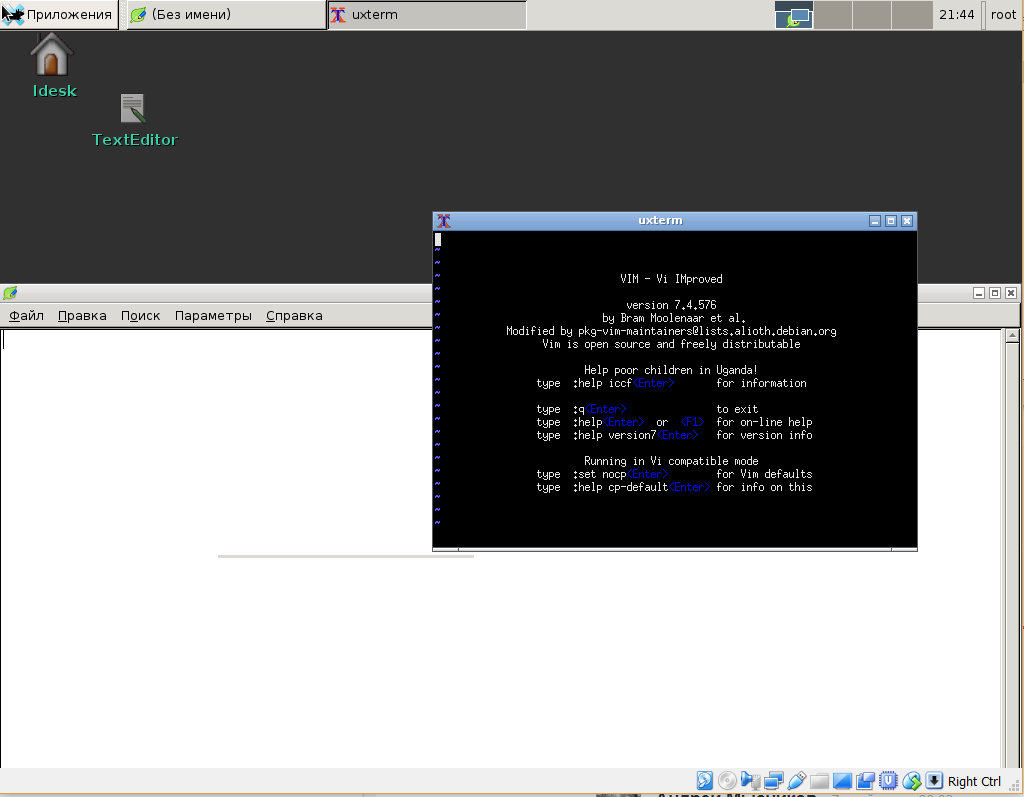
*Y: 100*

*Command[0]: /usr/bin/leafpad*

*Command[1]: uxterm /usr/bin/vi*

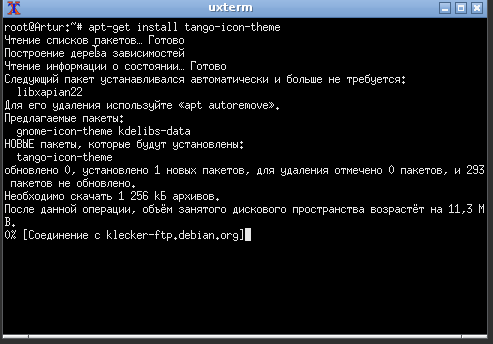
*end*

После выполнения этих несложных действий при двойном щелчке по ярлыку ЛКМ будет открываться *leafpad*, а при двойном щелчке ПКМ — текстовый редактор *vi*.



Теперь настроим 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*



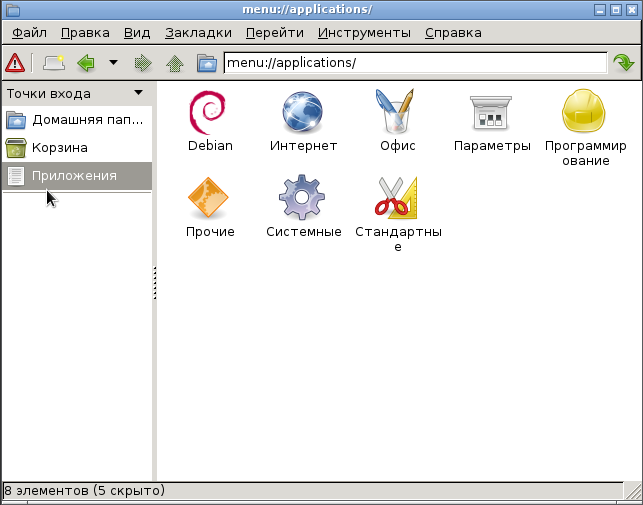
Для того, чтобы именно эти иконки использовались в *GTK-приложениях*, необходимо создать для них конфигурационный файл:

*leafpad ~/.gtkrc-2.0*

В этом файле необходимо прописать следующую строчку:

*gtk-icon-theme-name=”Tango”*

Таким образом, настроили отображение иконок в *PCManFM*.



**Установка и настройка conky**

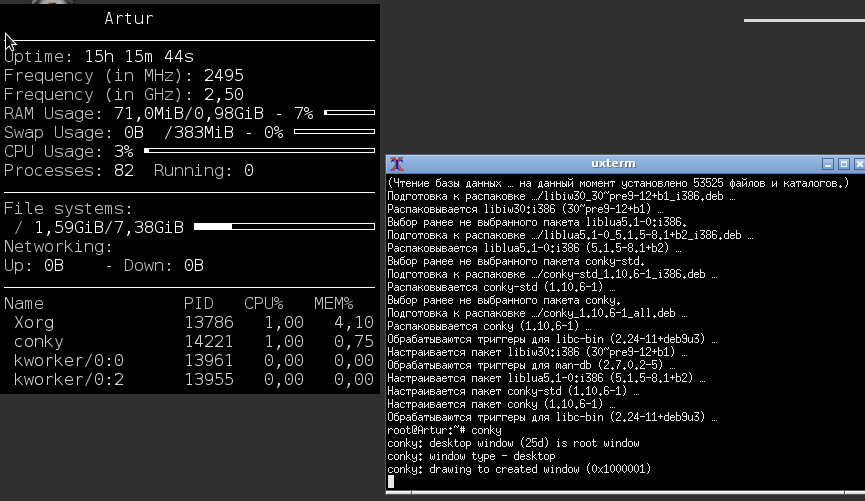
*conky* представляет собой менеджер системных ресурсов. Это гибкая утилита, позволяющая отслеживать такие параметры, как использование ОЗУ и диска, нагрузку на ЦП. Однако *conky* может отображать даже новости и погоду.

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.

Обратим 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* отредактируем его конфигурационный файл:

*leafpad /etc/conky/conky.conf*

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

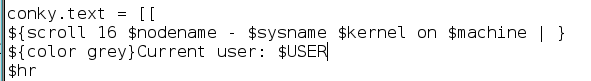
Необходимо изменить положение *conky*. Для этого заменим строку

*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*



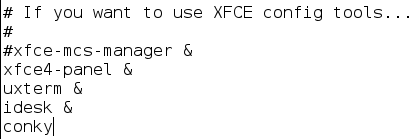
После настройки снова запустим *conky*. Отредактируем скрипт автозапуска, добавив в него *conky*, теперь скрипт будет выглядеть следующим образом:

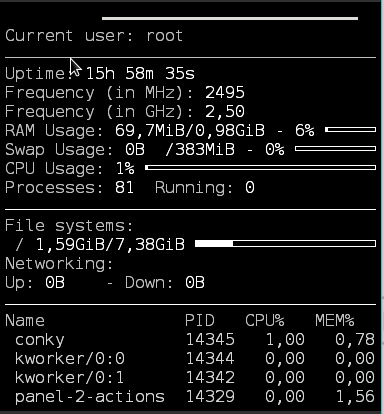
*xfce4-panel &*

*uxterm &*

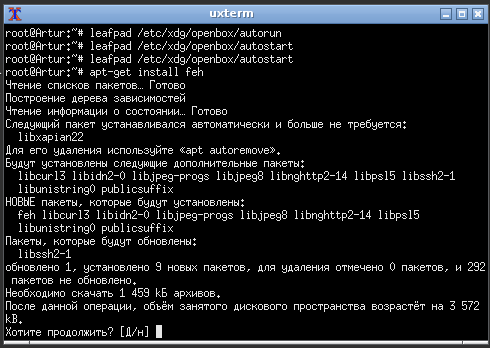
*idesk &*

*conky*





Настроим отображение фоновой картинки в качестве обоев. Для этого вместе с *Openbox* будем использовать программу *feh*. Установим её следующей командой: *apt-get install feh*



Затем скачаем через *w3m* картинку, которую будем использовать в качестве обоев для рабочего стола. Имя скаченного файла : *fon\_picture.jpg*

Для файла *fon\_picture.jpg* установка изображения в качестве фона может выглядеть следующим образом:

*feh --bg-scale ~/fon\_picture.jpg*

Отредактируем скрипт *autostart* так, чтобы фоновое изображение устанавливалось каждый раз. Для этого открываем в *leafpad* файл *autostart*. Его следует отредактировать следующим образом:

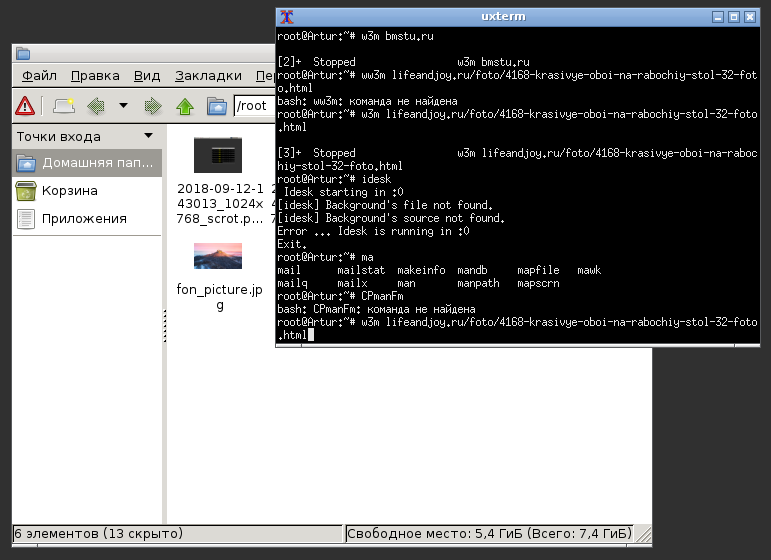
*xfce4-panel &*

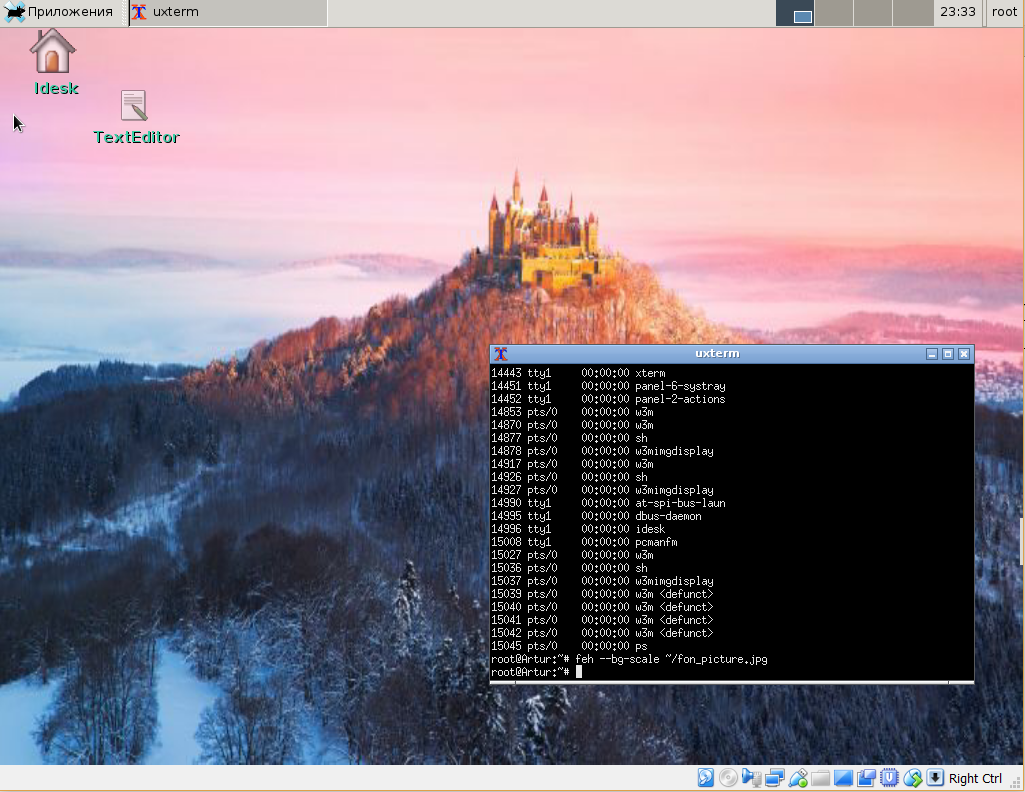
*uxterm &*

*idesk &*

*conky &*

*feh --bg-scale ~/fon\_picture.jpg*





**Управление с помощью консоли**

В данной 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*.

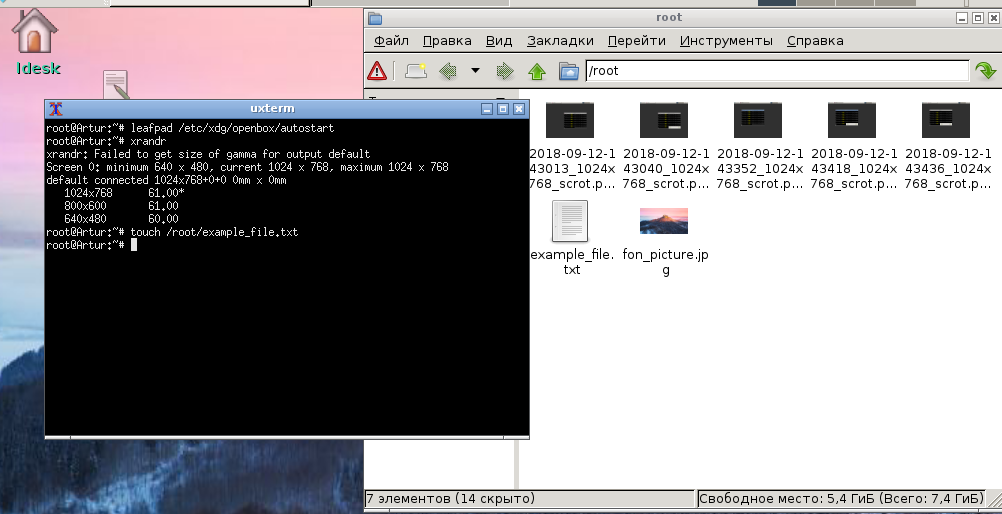
**Основные команды и утилиты**

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**

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

*touch /root/example\_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 «x» и оканчивающиеся на «e» (т.е. если в строке есть «e» после «x»):

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



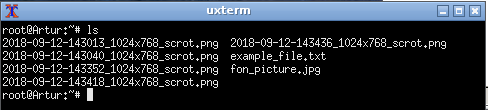
*mkdir* – создать новую папку.

*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**

*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.

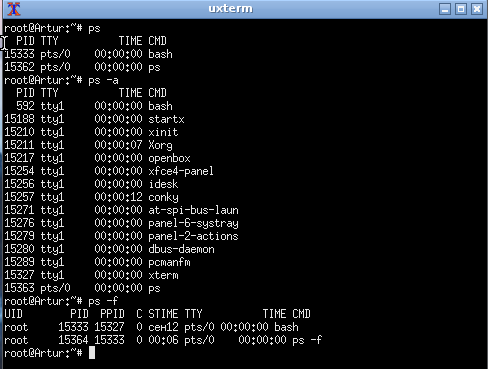
С использованием ключа -f выводятся следующие поля:

UID – идентификатор пользователя, вызвавшего процесс.

PPID – PID родителя (parent PID).

C – приоритет, используемый планировщиком для данного процесса.

STIME – время старта процесса.



*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.

*top* – утилита, показывающая запущенные процессы и сортирующая их в реальном времени. Некоторые ключи:

-n – ограничивает количество число обновлений программы (сколько раз обновить top до завершения).

-U – показать процессы, принадлежащие только определённому пользователю.

Некоторые другие параметры top можно было заметить ранее в конфигурационном файле conky.

Выход из top осуществляется с помощью нажатия клавиши q.

Некоторые из столбцов в top схожи со столбцами ps, ниже приведены только различающиеся:

PR – приоритет.

NI – значение параметра NICE, влияет на приоритет процесса.

VIRT – количество используемой виртуальной памяти.

RES – объём используемой физической памяти.

SHR – объём используемой разделяемой памяти.

S – статус процесса.

%CPU – процент использования ЦП.

%MEM – процент использования ОЗУ.

TIME+ - общее время активности процесса.

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* – 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):

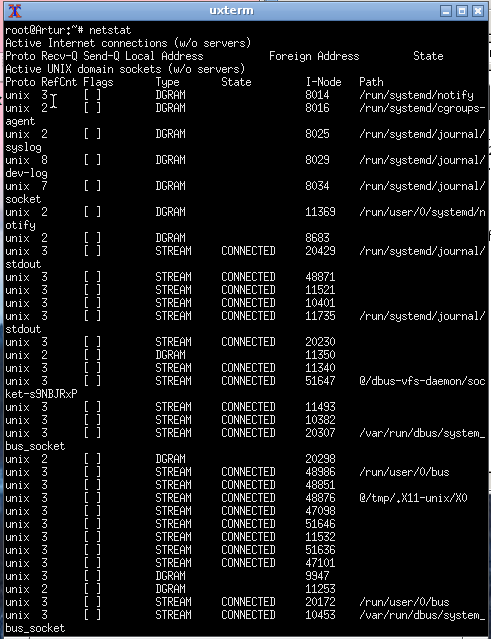
link- сетевой интерфейс.

address- адрес на интерфейсе в соответствии с протоколом.

route- запись таблицы маршрутизации.

rule- правило маршрутизации.

tunnel- туннель IP.



**Написание скриптов на bash**

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*.

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*

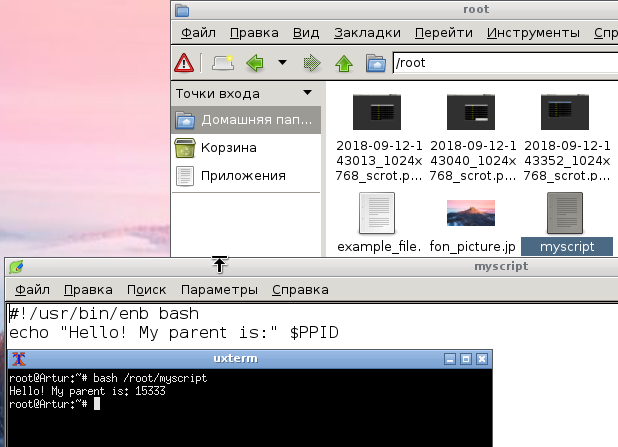
*echo “Hello! My parent is:” $PPID*

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 его 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.

**Вывод:** Таким образом, в ходе лабораторной работы было произведено рассмотрение и исследование среды управления Linux.