

# Лабораторная работа №1

## «Основа работы в командной строке Astra Linux»

### 1. Теоретическая часть

#### 1.1. Появление UNIX-подобных систем

История операционных систем начинается в те времена, когда у компьютеров не было мониторов. На заре вычислительной техники все компьютеры были так называемыми мэйнфреймами (от англ. *mainframe*) и не содержали никаких операционных систем. Каждый программист, получавший доступ к большой машине, приносил с собой набор перфокарт или перфолент, содержащих программу, которую мэйнфрейм должен был выполнить и выдать результат.

Но вычислительная техника усложнялась и каждый раз программировать простейшие процессы взаимодействия софта с железом было слишком накладно. Поэтому возникла необходимость в прослойке, избавляющая программистов от ненужной траты времени. Компьютер – дорогое и медленное устройство, при этом оператор использует данное ему процессорное время слишком неэффективно, отчего компьютер в большую часть времени ничего не делает.

Так возникла идея операционной системы разделяемого времени, которая позволяет нескольким операторам одновременно использовать компьютер. Например, пока один осуществляет ввод данных, система обрабатывает программу другого.

В 60-е года XX века усилиями нескольких компаний была разработана система *Multiplexed Information and Computing Service* (MultICS). Однако её громоздкость и другие недостатки привели к тому, что один из разработчиков MultICS, Кен Томпсон, решил создать собственную операционную систему с иерархической файловой системой, концепцией процессов, являющейся

основой любой многозадачной операционной системы, командный интерпретатор, текстовый редактор и ассемблер в 1969 г. Система получила название Uniplexed Information and Computing Service (UNICS). Для простоты звучания аббревиатуру сократили до UNIX.

Изначально система была написана на языке ассемблера мини-компьютера PDP-7, однако позднее при поддержке Денниса Ритчи была переписана на высокоуровневый язык программирования C, благодаря чему UNIX не был привязан к конкретному аппаратному обеспечению и стал портируемым, что позволило этой системе быстро и широко распространиться на все типы компьютеров.

***Операционная система** — это программа, которая управляет всеми другими прикладными программами, а также распределением и использованием аппаратных ресурсов компьютера.*

Со временем UNIX был лицензирован и оброс проприетарным программным обеспечением. Это означало, что пользователь не имел права изменять программы так, как ему было бы удобнее для каких-либо задач, и передавать копии кому-либо.

Недовольный сложившейся ситуацией, в 1983 г. Ричард Столлман решил создать свою собственную операционную систему, подобную UNIX, которую назвал GNU (рекурсивный акроним от англ. GNU is Not UNIX). По идее, GNU должен был быть похожим по возможностям на UNIX, но при этом свободно распространяемым. Это значит, что любой пользователь может вносить свои правки в исходные коды ПО и свободно распространять модифицированную программу с указанием авторства.

В течение десятилетия Столлман и его единомышленники написали множество свободных программ проекта GNU, включая GNU Compiler Collection (набор инструментов для компиляции программ), текстовый редактор Nano, загрузчик GRUB и т.д. Однако у проекта не было самого

главного - ядра операционной системы, без которого программное обеспечение не работало бы.

***Ядро** - часть операционной системы, обеспечивающая приложениям координированный доступ к ресурсам компьютера.*

## 1.2. Рождение Linux

В 90-х годах студент по имени Линус Торвальдс приобрёл для себя домашний компьютер, чтобы он мог заниматься любимым делом – программированием. Однако предустановленная операционная система MS-DOS не удовлетворяло его из-за её ограниченности в использовании аппаратного обеспечения, поэтому он стал искать другую систему. Изначально планировал использовать UNIX, с которой работал во время обучения в университете, однако её высокая стоимость вынудила Линуса перейти к более доступной для студента MINIX, клону UNIX.

***Клон** - программа, функциями и поведением похожая на другую программу, но не содержащая её исходные коды.*

Однако, хотя MINIX был намного мощнее MS-DOS, он все же имел ряд серьёзных недостатков:

- не весь исходный код был обнародован;
- отсутствовали некоторые возможности UNIX;
- взимался немалый (хотя и более дешёвый, чем для многих других операционных систем) лицензионный сбор.

Таким образом, Торвальдс решил создать новую операционную систему с нуля, основанную как на MINIX, так и на UNIX. Когда он решил опубликовать свою работу на FTP-сервере, его система умела запускать оболочку bash и GCC.

При этом Линус решил выпустить Linux под GPL (GNU General Public License), которая была разработана Ричардом Столлманом. Она позволяет любому изучать, использовать, изменять, расширять и распространять

программное обеспечение при условии, что они предоставляют свободный доступ к исходному коду для любых модифицированных версий, которые они создают и затем распространяют.

Linux - это прежде всего ядро, которое является важной частью операционной системы, но само по себе оно бесполезно. Linux обычно применяют в комбинации с системными утилитами и программами проекта GNU. Поэтому корректно будет называть всю систему GNU/Linux.

### **1.3. Дистрибутивы GNU/Linux**

Ещё с распространения UNIX среди образовательных и коммерческих организаций возникали различные сборки системы. То есть в основу брался исходный код UNIX, в него добавлялись новые функции, разрабатывались новые программы и всё вместе единым пакетом распространялось среди других пользователей.

Такую форму распространения программного обеспечения называют дистрибутивом (англ. distribution, от англ. distribute — распространять). Например, таким дистрибутивом UNIX являлся BSD (англ. Berkeley Software Distribution), от которой со временем образовались FreeBSD, OpenBSD и т.д.

Так как операционная система GNU/Linux выпущена под лицензией GPL, возникло великое множество её дистрибутивов. Наиболее популярными из них являются Debian, Ubuntu и Mint, которые используются для повседневных задач на ПК. На мобильных устройствах популярен Android, который в своей базе содержит модифицированный код ядра Linux и множество проприетарных программ компании Google. Одноплатные компьютеры Raspberry Pi, широко используемые в рамках концепции Internet of Things, имеют на борту Raspberry Pi OS, который в свою очередь является дистрибутивом Debian.

Поэтому не существует чистой операционной системы под названием Linux. В этом и главное отличие свободных систем GNU/Linux от проприетарной системы Windows, принадлежащей компании Microsoft.

### **1.3. Astra Linux**

«Группа Астра» - российский разработчик и изготовитель различных программных продуктов, в том числе российских операционных систем (ОС) семейства Astra Linux. Дистрибутивы семейства Astra Linux являются производными дистрибутива Debian GNU/Linux. Отличительной особенностью является упор на обеспечение информационной безопасности до уровня государственной тайны «особой важности». Благодаря наличию сертификатов ФСБ, система активно внедряется в государственные организации, а в качестве технологических партнёров «Группы Астра» являются известные российские компании.

### **1.4. Основа работы в командной строке**

В Astra Linux в качестве графического интерфейса используется Fly собственной разработки. С помощью него, как и в операционной системе Windows, предоставляется возможность удобного и наглядного управления системой, выхода в Интернет, а также использования разного рода приложений.

Однако системы GNU/Linux изначально управлялись только через командную строку. Более того, даже сегодня использование строки существенно облегчает задачу администрирования системы. GUI (Graphic User Interface) не всегда может предоставить функции, необходимые для выполнения каких-либо задач.

Операционная система взаимодействует с пользователем посредством оболочки (англ. shell) - программы, которая предоставляет традиционный,

только текстовый пользовательский интерфейс. Ее основная функция - считывать команды, которые вводятся в консоль. В системах с графическим интерфейсом многие пользователи редко взаимодействуют с оболочкой напрямую. Однако графические интерфейсы являются всего лишь интерфейсами для оболочек, то есть это просто привлекательные окна, которые создаются поверх оболочки и которые используют команды оболочки.

Для работы с командной строкой в Astra Linux предусмотрен эмулятор терминала Fly.

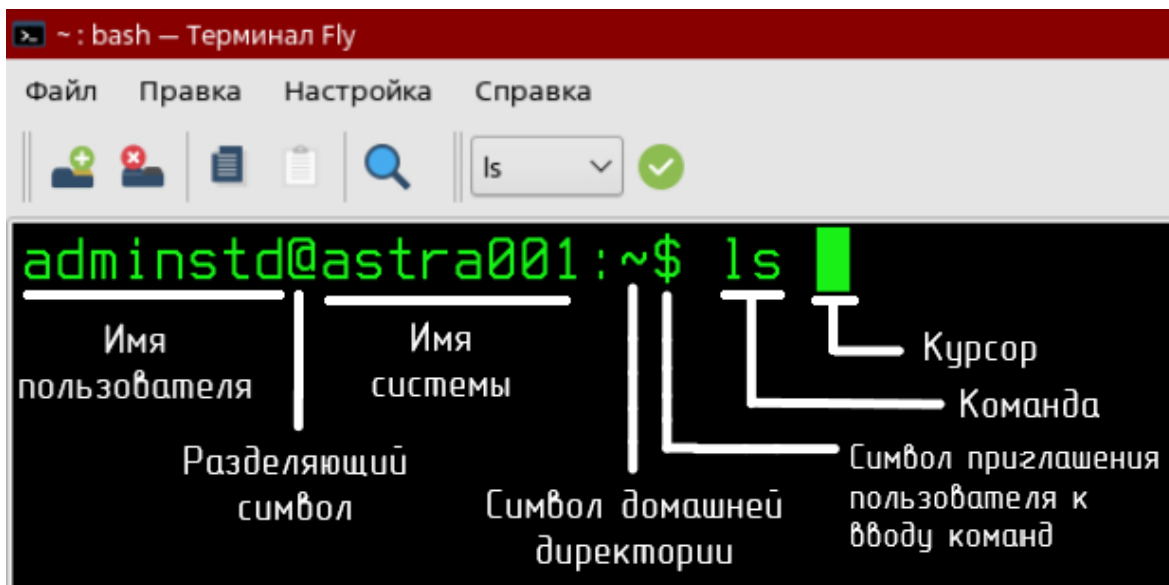
***Терминал** - устройство, предназначенное для последовательного ввода и вывода данных и предоставляющее пользовательский интерфейс.*

В прошлом терминалы представляли собой пишущие машинки, которые физически подключались к мейнфреймам и образовывали связь между оператором и машиной. Затем появились полноценные терминалы в виде устройства с клавиатурой и монитором, который подключался к компьютеру через последовательный интерфейс UART.

Спустя время терминал и компьютер слились воедино, а программная часть взаимодействия пользователя и система осталась такой же в UNIX-подобных системах. Разница состоит лишь в том, что терминал стал эмулируемым, то есть вместо физического терминала используется виртуальный.

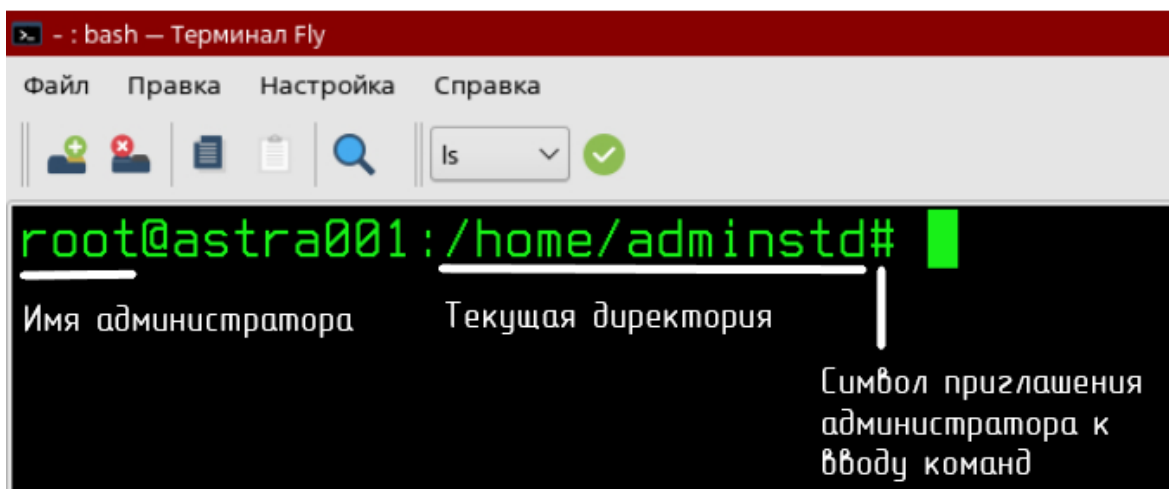
Для запуска терминала Fly в Astra Linux используется комбинация клавиш **alt+T** или через соответствующую иконку.

Эмулятор использует популярную оболочку `bash` (Bourne-again shell), которая является производной от оригинальной оболочки UNIX `sh`. Она имеет следующий вид:



При этом пользователь может зайти в систему под именем **root**, который обладает более широкими правами по управлению системы, как, например, установка и удаление программ, редактирование системных файлов и т.д.

В таком случае командная строка имеет вид:



Однако неопытным пользователям не рекомендуется работать с **root-правами**. Команды, которые могут нарушить работу системы, недоступны обычному пользователю.

В командной строке мы можем запускать программы, редактировать команды. Использование клавиш  $\uparrow$  и  $\downarrow$  позволяет прокручивать историю введённых строк. Клавиша **Tab** — автозавершение команды.

Сделаем обзор важных команд:

- **man** — программа, предназначенная для вывода подробной справочной информации о какой-либо программе. Например, чтобы побольше узнать о **man**, необходимо ввести команду **man man**. Нажав клавишу **h**, можно узнать клавиши управления программой, а чтобы выйти из программы, необходимо нажать клавишу **q**.

Для стандартизации составления справок было принято, что при необходимости информация о какой-либо программе должна быть в соответствующие секции:

№ секции	Название секции
1	Пользовательские программы
2	Системные вызовы
3	Библиотечные функции
4	Специальные файлы
5	Форматы файлов
6	Игры
7	Другое

Чтобы посмотреть, какие есть секции справочной информации у какой-либо программы, надо ввести, например, **whatis man**. Тогда при обращении к справке можно указать номер секции.

У программы **man** доступны секции №1 и №7. Для обращения к секции №7 надо ввести **man man.7** или **man 7 man**.

- Для очистки терминала используется команда **clear**.



- Команда **whoami** выводит имя пользователя, команда **hostname** — имя системы (хоста).
- Программа **uname** собирает информацию о системе. Наиболее полная информация будет выводиться при написании **uname -a**. Стоит обратить внимание, что был использован ключ **-a**. Для получения информации о том, какие ключи поддерживает та или иная программа, необходимо указать ключ **-h** или **--help**.
- Программа **hostnamectl** аналогично собирает информацию о системе. Есть множество программ, которые по функциональности похожи, но могут иметь различия, необходимые для выполнения определённых задач.
- Для вывода информации о дистрибутиве можно воспользоваться командой **lsb\_release** с ключом **-a**.
- Программа **who** выводит информацию о всех зарегистрировавшихся в системе пользователях. Из истории известно, что UNIX изначально задумывался, как многопользовательская система.
- Команда **echo** выводит текст в терминал. Например, **echo Hello, World!**.

## 1.5. Дата и время

В графической оболочке мы имеем возможность смотреть дату и время, не предпринимая никаких действий. Однако при отсутствии GUI для просмотра времени используется команда **date**. Подробный вывод даёт команда **timedatectl**.

Операционная система GNU/Linux для вычисления времени использует так называемое **UNIX time**. Оно определяется как количество секунд, прошедших с полуночи (00:00:00 UTC) 1 января 1970 года. Чтобы узнать текущее Unix-время в большинстве Unix-подобных систем, можно использовать команду **date +%s**.

## 1.5. Переменные окружения

*Переменная окружения — текстовая переменная операционной системы, хранящая какую-либо информацию.*

Они необходимы для упрощённого доступа к какой-либо системной информации, например, путь до файла, число и т.д.

Переменные можно разделить на два класса:

- переменные командного интерпретатора bash (shell variables);
- переменные окружения (environment variables).

Отличие между классами переменных заключается в том, что переменные окружения доступны во всех запущенных из оболочки процессах, а значение переменной командного интерпретатора только в текущей оболочке. Например, если запустить в терминале приложение, то из него будут доступны значения переменных окружения, но не переменных командного интерпретатора.

Список переменных окружения можно вывести с помощью команды **printenv**.

Для использования значения переменной используется символ разыменовывания **\$** (не путать с символом приглашения вне контекста).

Например, имя пользователя можно вывести командой **echo \$USER**, где **USER** — имя переменной, хранящей имя текущего пользователя.

Для создания пользовательской переменной командного интерпретатора используется конструкция **ИМЯ\_ПЕРЕМЕННОЙ="значение переменной"** (без пробелов!). Она хранится только для текущего сеанса терминала и не записывается в список переменных окружения.

Для удаления созданной переменной используется команда **unset ИМЯ\_ПЕРЕМЕННОЙ**.

Для добавления переменной окружения в список нужно воспользоваться командой **export ИМЯ\_ПЕРЕМЕННОЙ="значение переменной"**. Однако и она хранится только для текущего сеанса терминала.

Чтобы сделать переменную доступной после перезагрузки операционной системы, необходимо добавить ее объявление в соответствующий файл конфигурации следующим путём:

```
echo 'export ИМЯ=ЗНАЧЕНИЕ' >> ~/.bashrc  
source ~/.bashrc
```

Механизм выполнения данных команд подробно рассмотрим в следующей лабораторной работе.

## 1.6. История команд Linux

При запуске командного интерпретатора история команд из файла, путь до которого хранится в переменной окружения HISTFILE, загружается в буфер, куда добавляются новые введённые команды. При выходе из командного интерпретатора команды из буфера добавляются в файл.

Чтобы посмотреть буфер команд, которые вводились в терминал, используется команда **history**. Для очистки буфера используется ключ **-с**.

Чтобы выполнить поиск по истории прямо во время ввода используется комбинация клавиш **ctrl+R**, после чего вводится начало команды.

## 2. Практическая часть

Запустите с помощью командной строки приложение LibreOffice, введя команду **libreoffice**. Обратите внимание, что во время работы приложения терминал не принимает команды, так как он занят и отсутствует символ приглашения.

Создайте в терминале новую вкладку командной строки для выполнения заданий. В LibreOffice создайте отчёт в следующем формате: номер задания,

номер задачи, текст задачи, введённые команды, их результаты. Используйте форматирование текста, различные цвета и выделения по своему усмотрению для удобства чтения и проверки.

## 2.1. Задание 1

2.1.1. Определите название и версию дистрибутива ОС, с которой вы работаете. В случае, если у вас есть выход в Интернет, откройте WEB-страницу **astralinux.ru** и укажите актуальную версию ОС.

2.1.2. Определите версию ядра ОС. В случае, если у вас есть выход в Интернет, откройте WEB-страницу **kernel.org** и укажите актуальную версию ядра ОС.

2.1.3. Запустите программу тестирования 3D-графики **glxgears**. С периодом в 5 секунд снимите показания счётчика FPS 5 раз, внесите их в таблицу LibreOffice и с помощью встроенной формулы посчитайте среднее арифметическое FPS.

2.1.4. Выведите текущее время системы по стандарту UTC.

2.1.5. Выведите текущее время системы в представлении **Unix time**. Используя калькулятор, представленный в Astra Linux, посчитайте приблизительно, сколько лет прошло с момента запуска счётчика.

## 2.1. Задание 2

2.2.1. Создайте переменную командного интерпретатора «ABC», где ABC – ваши инициалы и поместите в нее значение с днем вашего рождения. Прочитайте её значение в командной строке.

2.2.2. Чтобы создать новый текстовый документ, в терминале необходимо ввести команду **libreoffice --writer**. Сократите ввод команды до **\$L**, создав переменную окружения. Просмотрите все переменные окружения и убедитесь, что ваша переменная была добавлена.

2.2.3. Откройте новое окно терминала и проверьте содержимое созданной переменной. Сохраните переменную таким образом, чтобы после перезагрузки системы её значение сохранялось.

### **Контрольные вопросы**

1. Что такое операционная система?
2. Какую операционную систему можно назвать самой первой? Какие были её особенности?
3. Что такое GNU/Linux и чем отличается от Linux и UNIX?
4. Перечислите особенности Astra Linux, отличающие её от других дистрибутивов.
5. Для чего предназначены переменные окружения?