«Министерство образования и науки Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования «Алтайский государственный технический университет им. И.И. Ползунова»

Факультет информационных технологий

Кафедра прикладной математики

Отчёт защищён с оценкой\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Преподаватель Ненайденко А.С.

«\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2021 г.

Отчёт

Лабораторной работе №6

«»

Студент группы ПИ 92 В.М. Шульпов

Преподаватель доцент, к.т. н. Ненайденко А.С.

Барнаул 2021

1. Получить дистрибутив операционной системы FreeBSD с сайта freebsd.org. Рекомендуется использовать стабильную ветку 12.2

2. Ознакомиться с процедурой начальной инсталляции по документации на сайте.

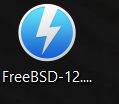
3. Инсталлировать базовую систему в виртуальное окружение. При инсталляции в соответствующих пунктах процедуры выбрать тип файловой системы и другие опции.

4. Инсталлировать графическое окружение рабочего стола и дополнительные графические программы.

5. Ознакомится с процедурой инициализации системы и сравнить ее с процедурами инициализации Linux.

6. Подготовить отчет и отправить его в Ilias

Для записи и установки с DVD или в виртуальную/реальную машину **скачиваем FreeBSD-\*-RELEASE-amd64-disc1.iso – образ**



"выбор программ" - отмечаем только lib32 - на всякий случай, если вдруг нам попадётся ПО, которое требует совместимости с 32-битным ПО (вряд ли, но вдруг).

Снимаем пометки с base-dbg, kernel-dbg, lib32-dbg - отладкой и анализом проблем в ОС я заниматься не буду.

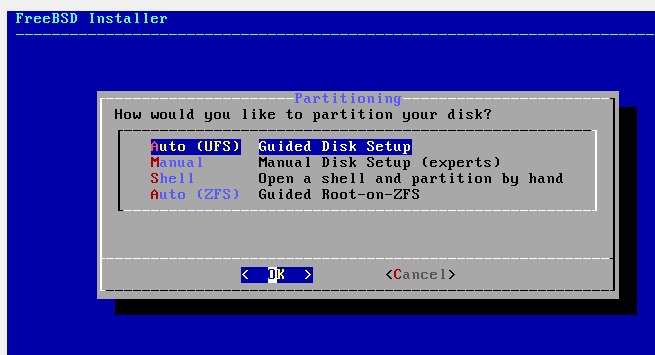
ports - нужны для сборки ПО из исходных текстов ПО.

src - исходные коды операционной системы - собирать свой вариант ядра FreeBSD мы не будем - будем использовать универсальный (GENERIC) вариант

tests - тестовый набор ПО - мне он тоже не нужен.

Их можно установить позже в любой момент.

**Разметка диска:**



При выборе zfs нужно будет сконфигурировать дисковый массив

**Про файловые системы:**

Файловая система связывает носитель информации (хранилище) с прикладным программным обеспечением, организуя доступ к конкретным файлам при помощи функционала взаимодействия программ API. Программа, при обращении к файлу, располагает данными только о его имени, размере и атрибутах. Всю остальную информацию, касающуюся типа носителя, на котором записан файл, и структуры хранения данных, она получает от драйвера файловой системы.

**Основными функциями файловой системы являются:**

* размещение и упорядочивание на носителе данных в виде файлов;
* определение максимально поддерживаемого объема данных на носителе информации;
* создание, чтение и удаление файлов;
* назначение и изменение атрибутов файлов (размер, время создания и изменения, владелец и создатель файла, доступен только для чтения, скрытый файл, временный файл, архивный, исполняемый, максимальная длина имени файла и т.п.);
* определение структуры файла;
* поиск файлов;
* организация каталогов для логической организации файлов;
* защита файлов при системном сбое;
* защита файлов от несанкционированного доступа и изменения их содержимого.

**Задачи файловых систем:**

* присвоение имен файлам;
* программный интерфейс работы с файлами для приложений;
* отображение логической модели файловой системы на физическую организацию хранилища данных;
* поддержка устойчивости файловой системы к сбоям питания, ошибкам аппаратных и программных средств;
* содержание параметров файла, необходимых для правильного взаимодействия с другими объектами системы (ядро, приложения и пр.).

**У windows меньше файловых систем:**

В случае с Windows все выглядит достаточно просто: NTFS на всех дисковых разделах и FAT32 (или NTFS) на флешках. Если установлен NAS (сервер для хранения данных на файловом уровне), и в нем используется какая-то другая файловая система, то практически никто не обращает на это внимания. К нему просто подключаются по сети и качают файлы

Пространство **FAT32** логически разделено на три сопредельные области:

* зарезервированный сектор для служебных структур;
* табличная форма указателей;
* непосредственная зона записи содержимого файлов.

**К недостаткам FAT32** относится ограничение размера файлов на диске до 4 Гб и всего раздела в пределах 8 Тб. По этой причине данная файловая система чаще всего используется в USB-накопителях и других внешних носителях информации. Для установки даже Windows 10 на внутреннем носителе потребуется более продвинутая файловая система.

После FAT32 разработали **exFAT** (расширенная таблица размещения файлов)

**NTFS(файловая система новой технологии)**

* предел размера файлов до 16 Гб
* поддерживает разделы диска до 16 Эб (эксабайт, 1018 байт)
* Использование системы шифрования Encryption File System (метод «прозрачного шифрования») осуществляет разграничение доступа к данным для различных пользователей, предотвращает несанкционированный доступ к содержимому файла
* Файловая система позволяет использовать расширенные имена файлов, включая поддержку многоязычности в стандарте юникода UTF, в том числе в формате кириллицы.
* Встроенное приложение проверки жесткого диска или внешнего накопителя на ошибки файловой системы [chkdsk](https://timeweb.com/go?url=https%3A%2F%2Fdocs.microsoft.com%2Fru-ru%2Fwindows-server%2Fadministration%2Fwindows-commands%2Fchkdsk&hash=e970cd170edaca0da388c33ce82f1276fa1ad373" \t "_blank) повышает надежность работы харда, но отрицательно влияет на производительность.

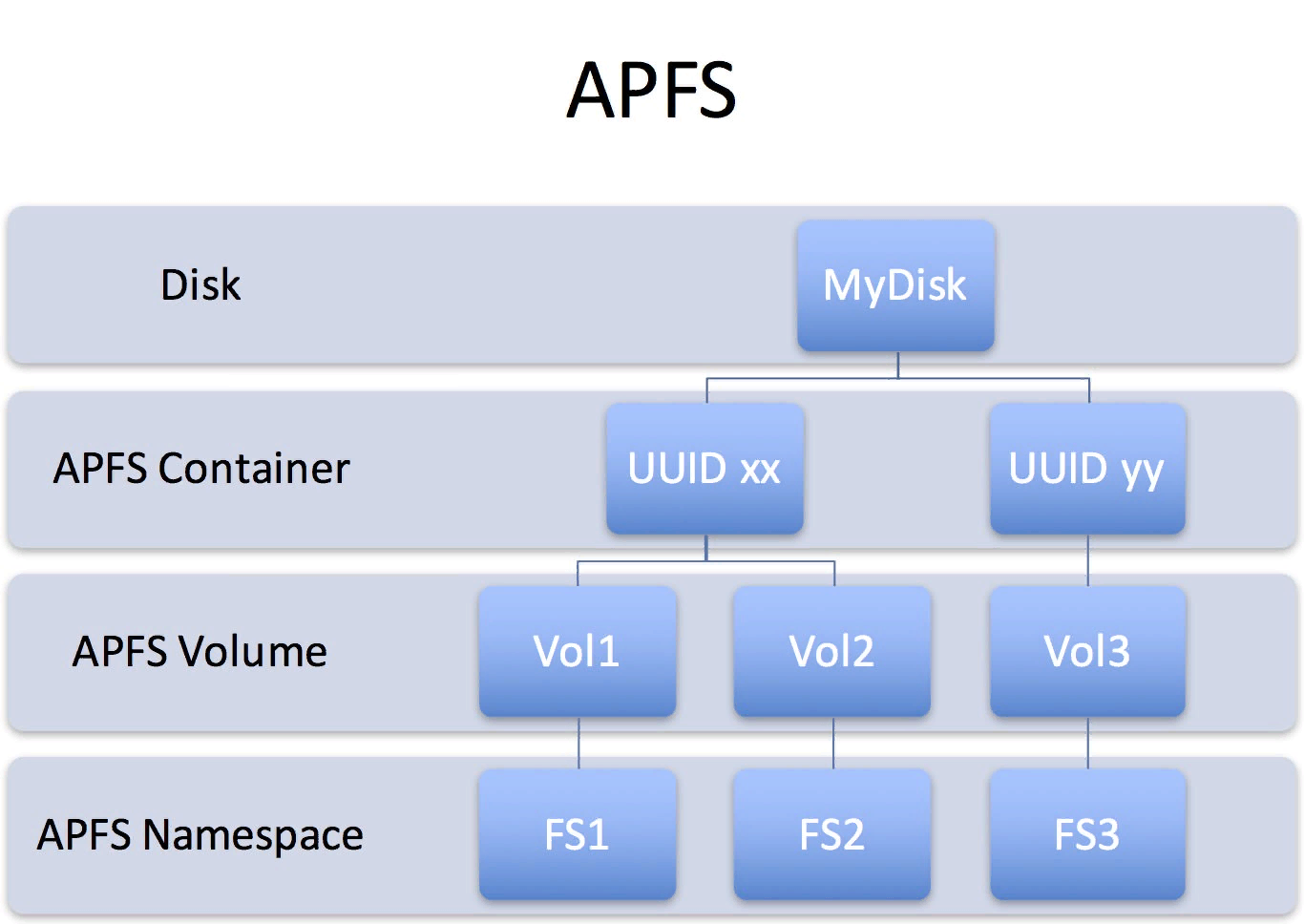
#### **ReFS (Resilient File System)**

Последняя разработка Microsoft, доступная для серверов Windows 8 и 10. Архитектура файловой системы в основном организована в виде B + -tree. Файловая система ReFS обладает высокой отказоустойчивостью благодаря реализации новых функций:

* Copy-on-Write (CoW) – никакие метаданные не изменяются без копирования;
* данные записываются на новое дисковое пространство, а не поверх существующих файлов;
* при модификации метаданных новая копия хранится в свободном дисковом пространстве, затем система создает ссылку из старых метаданных на новую версию.

Про **MacOS** подробно писать не буду, просто приведу схему:

HFS, HFS+, APFS…



Основные файловые системы, используемые в дистрибутивах **Linux**:

* Ext2;
* Ext3;
* Ext4;
* JFS;
* ReiserFS;
* XFS;
* Btrfs;
* ZFS.

1. **Ext2, Ext3, Ext4** или **Extended Filesystem** – стандартная файловая система.
2. **JFS**или **Journaled File System** используется там, где необходима высокая стабильность и минимальное потребление ресурсов (в первую очередь в многопроцессорных компьютерах).
3. **ReiserFS**такжеразработана в качестве альтернативы ext3, поддерживает только Linux. Динамический размер блока позволяет упаковывать несколько небольших файлов в один блок, что предотвращает фрагментацию и улучшает работу с небольшими файлами. Риск потери данных при отключении энергии.
4. **XFS** рассчитана на файлы большого размера, диски до 2 Тб, высокая скорость работы с большими файлами, отложенное выделение места, увеличение разделов на лету, незначительный размер служебной информации, невозможность уменьшения размера, сложность восстановления данных и риск потери файлов при аварийном отключении питания.
5. **Btrfs**или **B-Tree File System**легко администрируется, обладает высокой отказоустойчивостью и производительностью. Используется как файловая система по умолчанию в OpenSUSE и SUSE Linux.

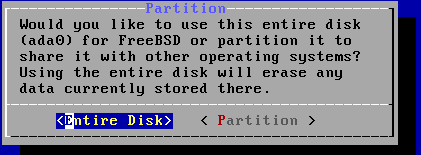
Другие ФС, такие как NTFS, FAT, HFS, могут использоваться в Linux, но корневая файловая система на них не устанавливается, поскольку они для этого не предназначены.

**FreeBSD:**

UFS (Unix File System), известной также под названием FFS (Fast File System). В современных компьютерных технологиях данные файловые системы могут быть заменены на альтернативные: ZFS для Solaris, JFS и ее производные для Unix.

Кластерные файловые системы включают поддержку распределенных хранилищ, расширяемость и модульность. К ним относятся:

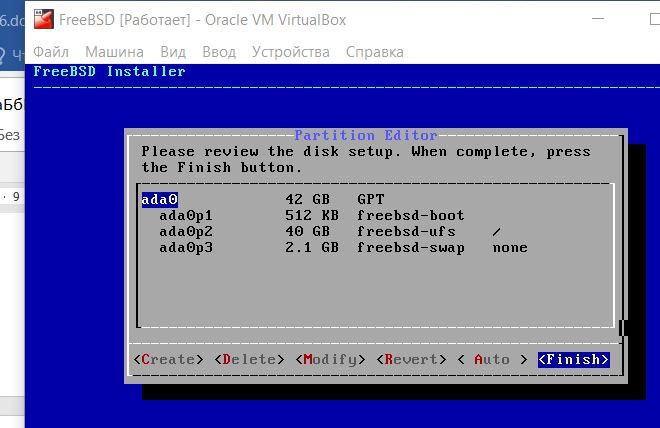
* ZFS
* Apple Xsan
* VMFS (Файловая система виртуальных машин)
* GFS – Red Hat Linux именуется как «глобальная файловая система» для Linux;
* JFS1 – используемой в старых системах хранения AIX.



**Partition schemes:**

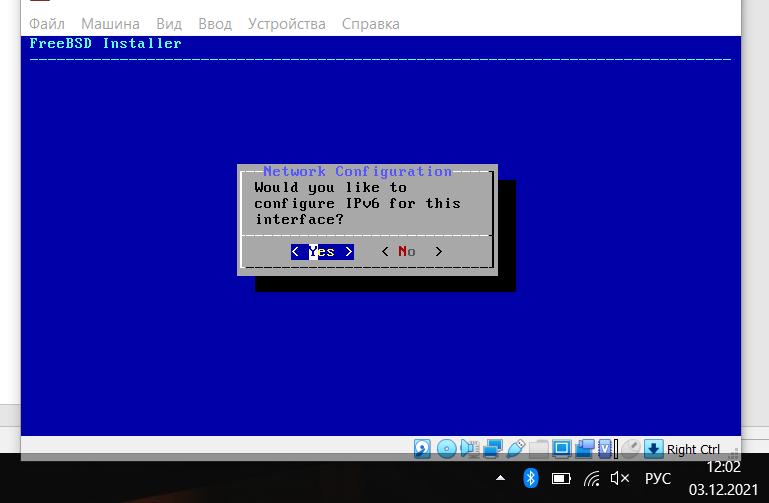
# GPT/MBR

* GPT допускает неограниченное количество основных разделов, в то время как MBR допускает только четыре основных, а остальные — дополнительные.
* GPT позволяет создавать разделы любого размера, в то время как MBR имеет ограничение в 2 ТБ.
* GPT хранит копию данных раздела, позволяя восстановить их в случае повреждения основного заголовка GPT; MBR хранит только одну копию данных раздела в первом секторе жёсткого диска, что может привести к потере всей информации в случае повреждении информации о разделах.
* GPT хранит значения контрольной суммы для проверки, что данные не повреждены, и может выполнить необходимое восстановление из других областей диска в случае повреждения; MBR не имеет способа узнать о повреждении данных, вы можете узнать об этом только если компьютер откажется загружаться или исчезнет раздел.
* GPT может быть несовместима со старыми системами.



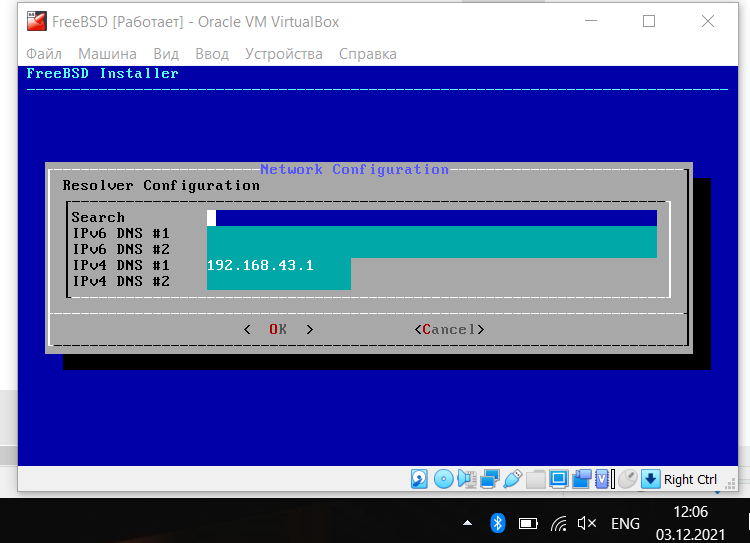
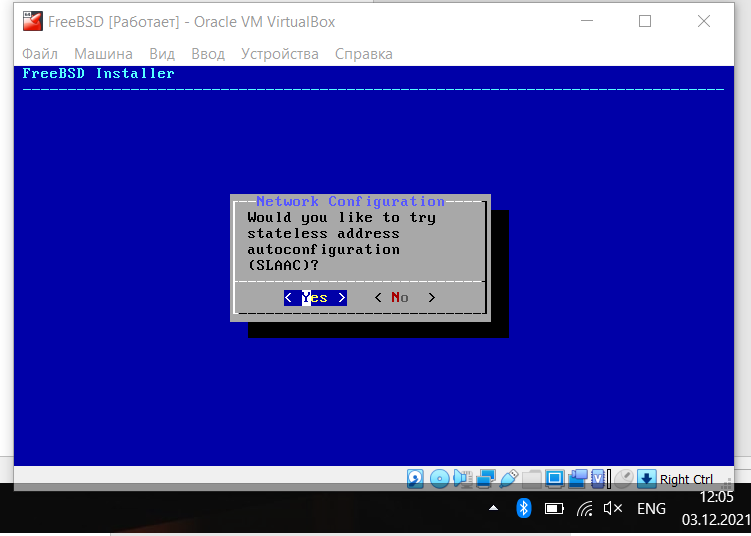
IPv4 оставил

DHCP оставил



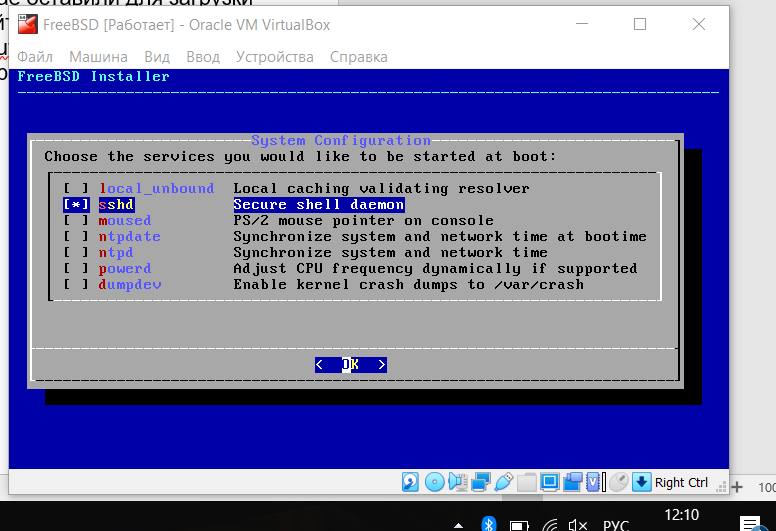
SLAAC оставил

**SLAAC** (**DHCPV6)** — это способ, который позволяет устройству получить свой префикс, длину префикса и адрес шлюза по умолчанию от маршрутизатора IPv6 без помощи DHCPv6-сервера. При использовании SLAAC для получения необходимой информации устройства полагаются на сообщения «Объявления маршрутизатора ICMPv6».

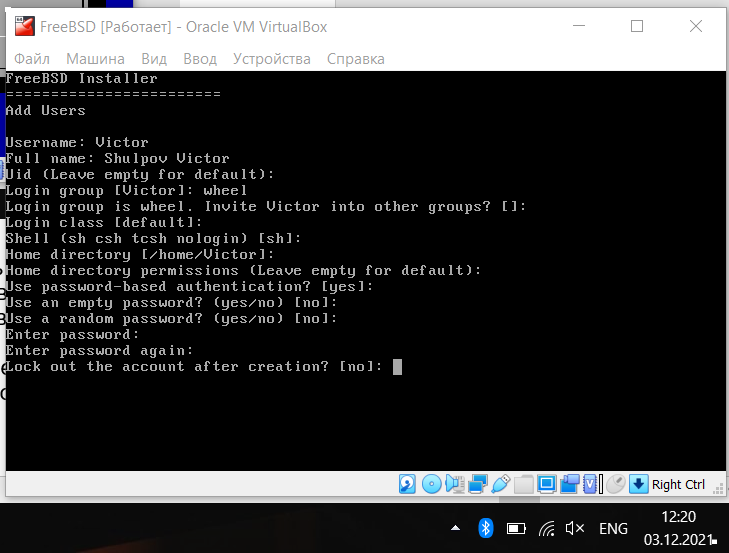


Далее были вопросы про время и часовой пояс

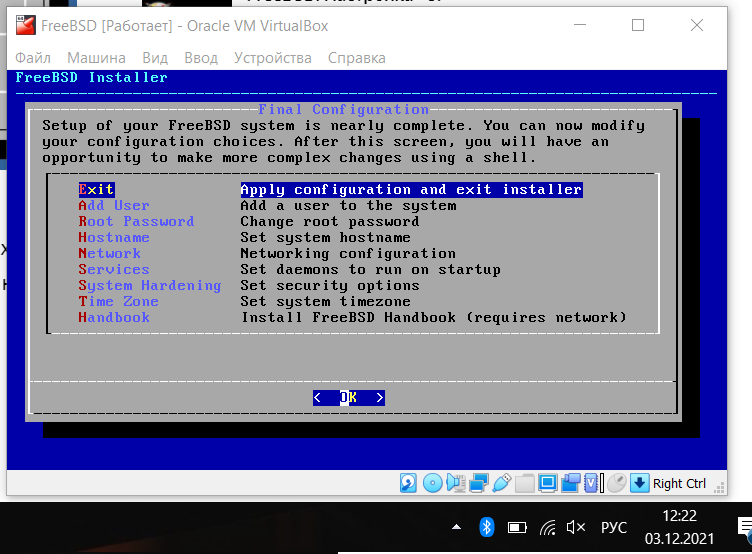
Далее для всех вопросов далее можно оставлять значения настроек по умолчанию. Их и другие всегда потом можно включить. В данном случае оставили для загрузки только сервис удалённого доступа sshd - он позволяет найти удалённо на вашу систему по протоколу ssh. Для Windows есть программа putty, которая позволяет организовать удалённый сеанс, а в *unix* в shell просто набрать "ssh IP".



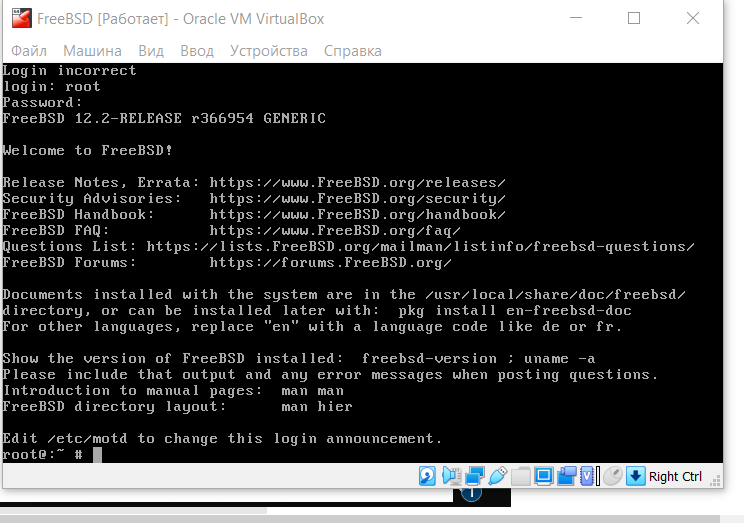
Я не занимаюсь разработкой для FreeBSD и не занимаюсь отладкой ПО и не нужен параноидальный контроль безопастности, поэтому расставляю параметры, как на предыдущей картинке. Кстати, "clear\_tmp" нужно активировать, так, как это такой же по принципу подход, как и шифрация области swap - в tmp много может быть много секретного и вообще размер свалки всякого в этом разделе может быть очень большим - поэтому автоматическое освобождение будет совсем не лишним.

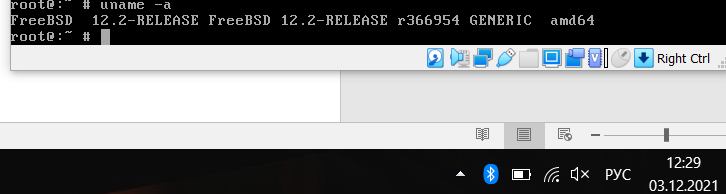


Не добавлял пользователя Victor в группы, кроме wheel

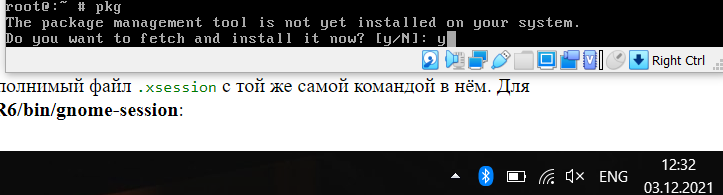


FreeBSD УСТАНОВЛЕНО!!!!!!!!!



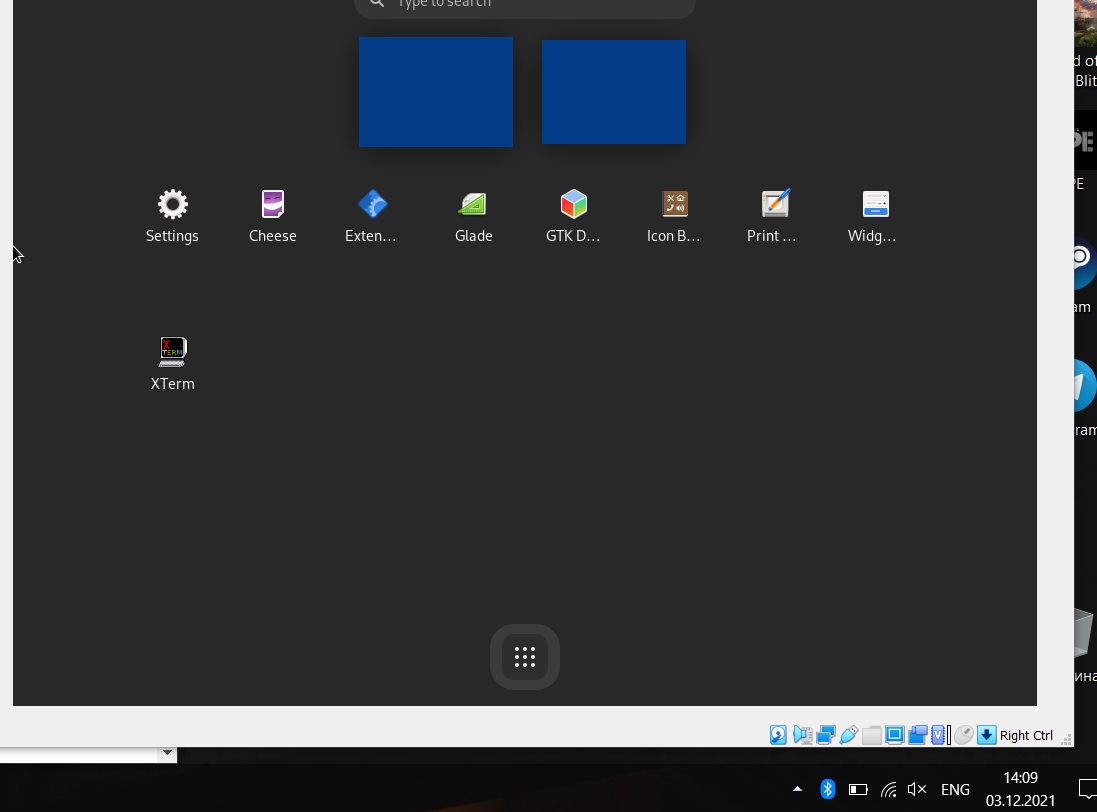


Установка пакетного менеджера:



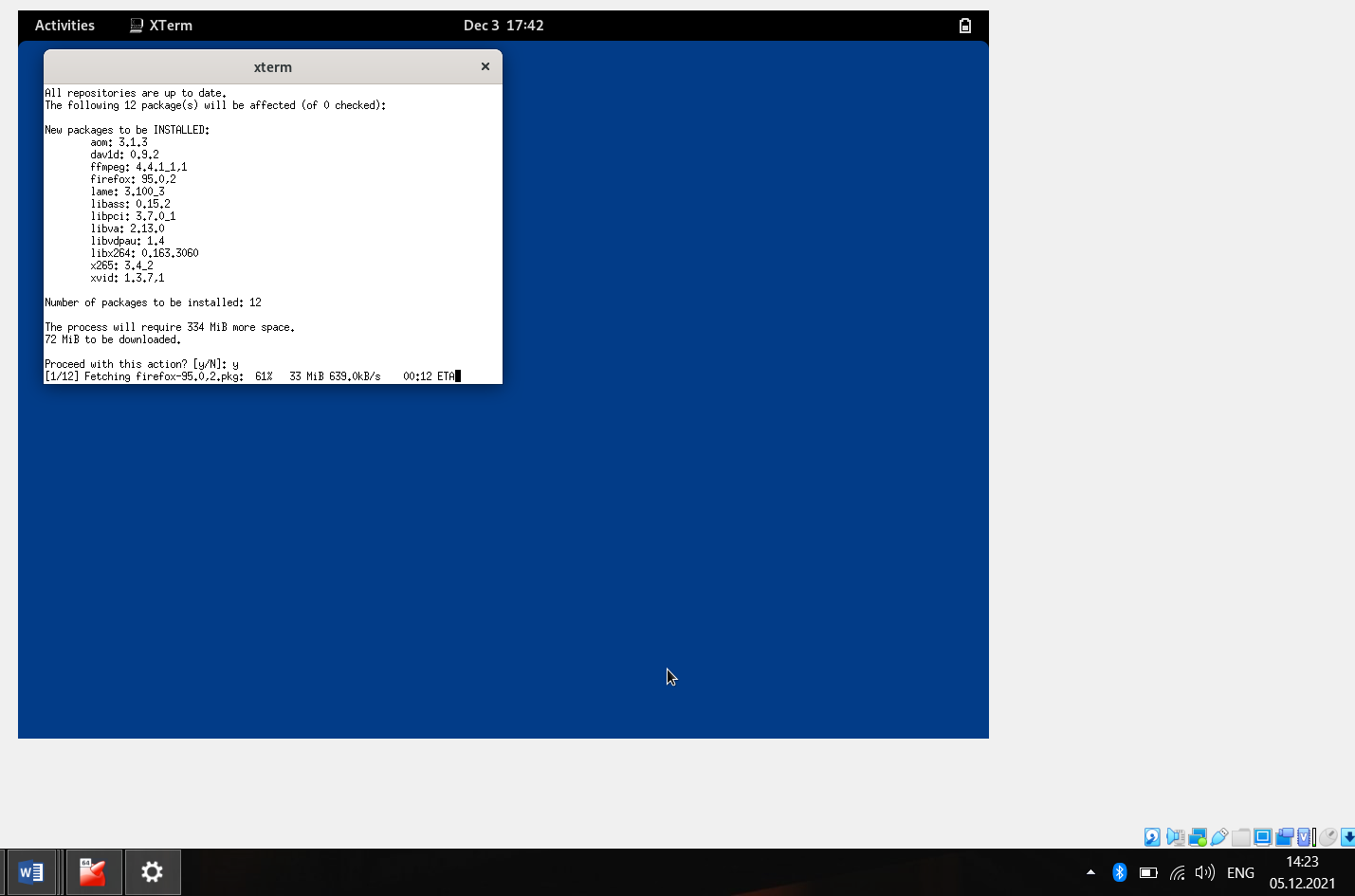
Установка графической оболочки.

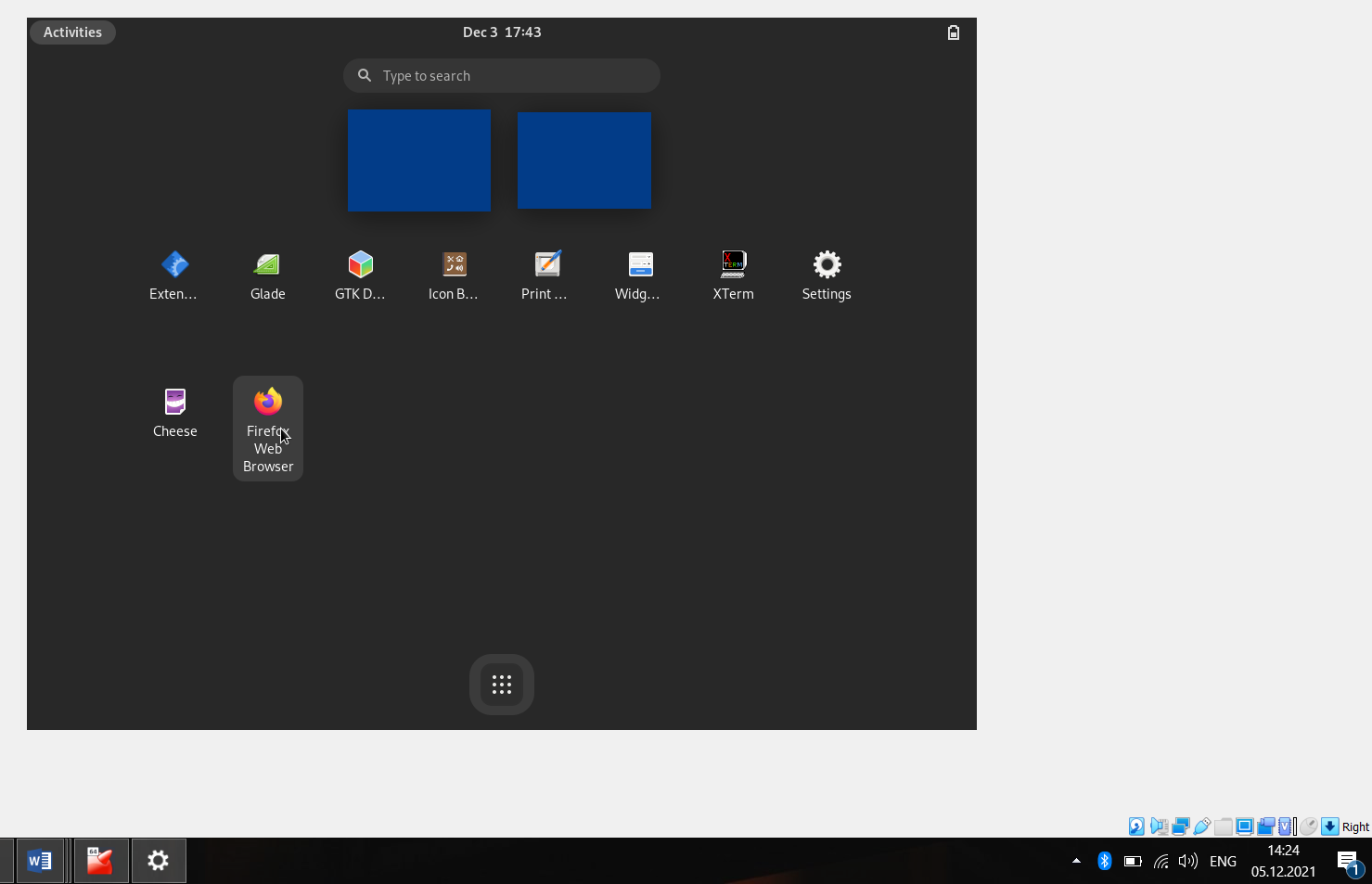
С помощью команды pkg install gnome-shell устанавливаем графическую оболочку gnome 3. Устанавливаем xorg с помощью команды pkg install xorg.



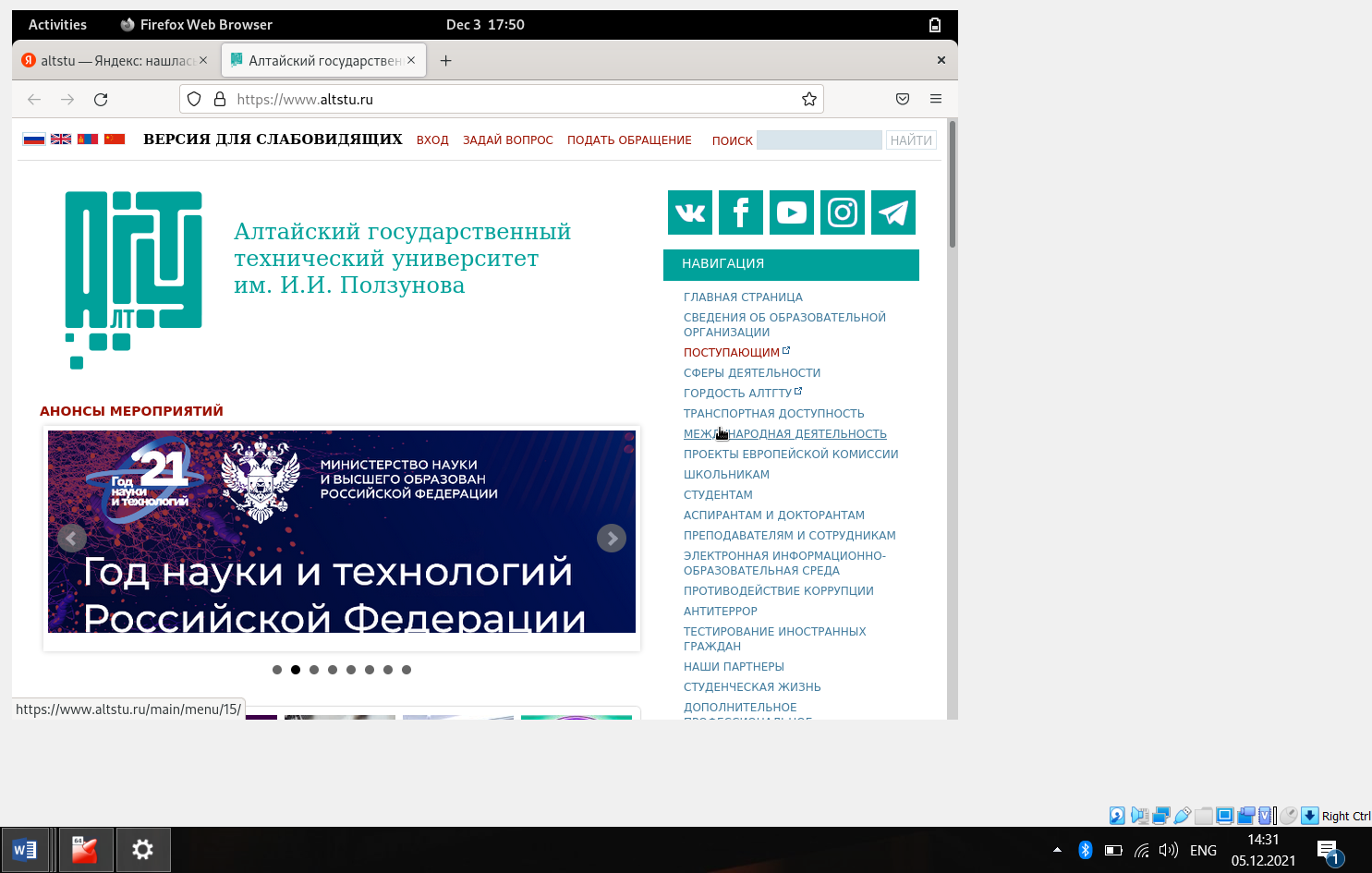
Графическая оболочка GNOME установлена!

Устанавливаем браузер: pkg install firefox





Браузер работает:

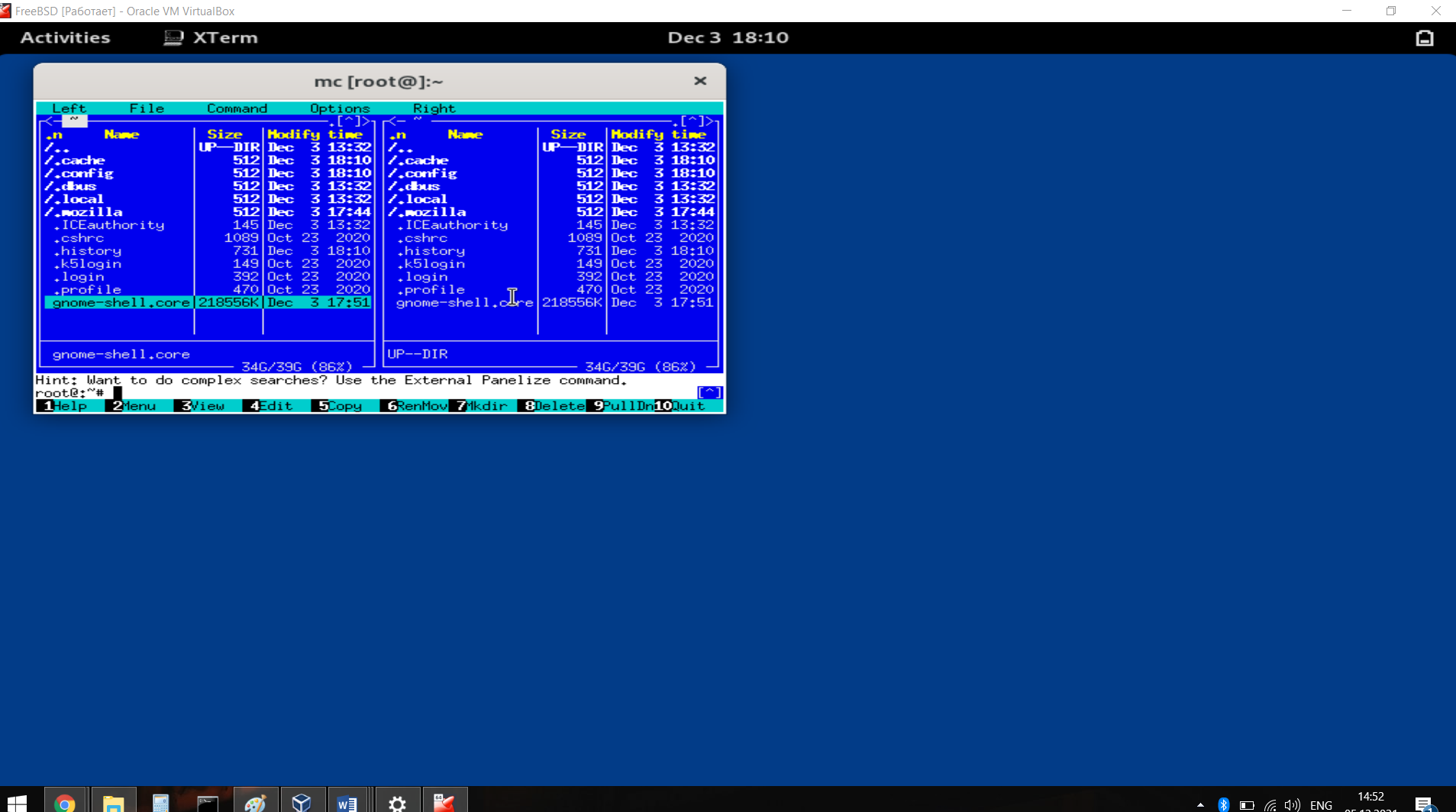


Установим nginx, а что нам ещё понадобиться на FreeBSD?

Nginx — веб-сервер и почтовый прокси-сервер, работающий на Unix-подобных операционных системах.

pkg install nginx

Установим файловый менеджер: pkg install mc



**Linux/FreeBSD**

Процесс инициализации в linux:

* 1. Самотестирование при включении (POST – Power-On Self-Test)

(будет слышно звуковые сигналы при ошибках)

* 1. BIOS ищет первый сектор на жестком диске, где стоит система (MBR – master boot record), загружает его в память и исполняет.
  2. MBR запускает GRUB (GRand Unified Bootloader), который загружает ядро.
  3. GRUB передает контроль Kernel (ядру)

Задача Kernel – запуск первого процесса

Данный процесс называется - init и имеет PID=**1**. Это прародитель всех **процессов** в системе.

(с 2010 года процесс init был вытесенен systemd — это система управления системой)

Процесс инициализации в FreeBSD похож на процесс инициализации Linux, но есть существенные отличия:

* Менеджер загрузки – boot0 (boot easy), grub, gag, lilo
* Оставшаяся часть системы начальной загрузки FreeBSD разделяется на три этапа.
  + запускается из MBR, и он знает достаточно для перевода компьютера в особое состояние и загрузки второго этапа.
  + Второй этап может делать несколько больше до запуска третьего этапа.
  + Заканчивается работа по загрузке операционной системы.

Работа разделена на эти три этапа, потому что стандарты ПК ограничивают размеры программ, которые могут быть запущены на первом и втором этапах. Последовательное выполнение работ позволяет FreeBSD получить более гибкий загрузчик.

* Затем стартует ядро, которое начинает опознавать устройства и выполняет их инициализацию. После завершения процесса своей загрузки, ядро передает управление пользовательскому процессу с именем [init(8)](https://www.freebsd.org/cgi/man.cgi?query=init&sektion=8&format=html), который выполняет проверку дисков на возможность использования.
* Затем [init(8)](https://www.freebsd.org/cgi/man.cgi?query=init&sektion=8&format=html) запускает пользовательский процесс настройки ресурсов, который монтирует файловые системы, выполняет настройку сетевых адаптеров для работы в сети и вообще осуществляет запуск всех процессов, обычно выполняемых в системе FreeBSD при загрузке.

**Основные сходства**

* Обе ОС относятся к семейству Unix-подобных операционных систем, а значит используют схожие паттерны. Например, они аналогичным образом выстраивают иерархию файловой системы и передают команды через командный интерпретатор (shell).
* Имеют открытый исходный код и разработаны на некоммерческой основе — то есть пользователи кода могут изменять его под свои цели.

**Главные различия**

* Дистрибутивы Linux используют лицензию GNU GPL, или «copyleft» («авторское лево»). Она позволяет использовать оригинальный код для создания новых продуктов, не запрашивая разрешения владельца исходных текстов, но сохраняя условия его распространения. FreeBSD использует BSD-лицензию, которая содержит больше свободы — не предполагает обязательного сохранения всех условий распространения. Это стоит учитывать при разработке проприетарного ПО.
* Каждая из ОС BSD не только работает на отдельном ядре, но и имеет собственные базовые компоненты. Для каждого отдельного дистрибутива они могут быть сформированы из отдельных источников. Linux же является ядром сама по себе.

При установке FreeBSD предлагается:

* выбрать сетевой интерфейс
* настроить для него IPv4 и IPv6
* подключить DHCP
* задать IP адрес, маску, шлюз по умолчанию, DNS сервера
* установить часовой пояс, время, дату, время
* настроить автозапуск служб
* настроить безопасность
* добавить пользователей
* при завершении установки можно исправить предыдущие настройки

**Вывод:** при установке ОС FreeBSD нужно выполнить достаточно гибкую настройку, которая может быть сложна для обычного пользователя. Данная операционная система имеет открытый исходный код и может быть дописана под нужды пользователя. По стандарту она не имеет графической оболочки и её нужно будет установить дополнительно, если потребуется. Устанавливать ПО для FreeBSD можно двумя способами (установить из портов. (это исходники программ, чтобы поставить, их надо собрать) или установить из пакетов. (это готовые установщики программ). FreeBSD очень стабильна, она предоставляет надежные сетевые службы и эффективное управление памятью.