ДНІПРОВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ІМЕНІ ОЛЕСЯ ГОНЧАРА

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

КАФЕДРА КОМП’ЮТЕРНИХ ТЕХНОЛОГІЙ

**Лабораторна робота №1**

**з курсу «Операційні системи»**

**Варіант №7**

Виконав:

студент групи ПА-19-2

Iльяшенко Єгор

Дніпро, 2021

[Вступление 3](#_Toc85870783)

[Установка ОС Windows 4](#_Toc85870784)

[Процесс загрузки MS-DOS 14](#_Toc85870785)

[Процесс загрузки Windows 16](#_Toc85870786)

[Процесс загрузки Linux 22](#_Toc85870787)

[Процесс загрузки QNX 23](#_Toc85870788)

[Процесс загрузки UNIX 26](#_Toc85870789)

[Процесс загрузки Mac-OS 30](#_Toc85870790)

[Команды ОС Windows 31](#_Toc85870791)

[Реестр ОС Windows 33](#_Toc85870792)

[Работа с потоками 35](#_Toc85870793)

[Выводы 38](#_Toc85870794)

[Список литературы 39](#_Toc85870795)

### Вступление

В лабораторной работе №1, необходимо рассмотреть установку пяти ОС на персональный компьютер и научиться работе с потоками при помощи языка программирования с++

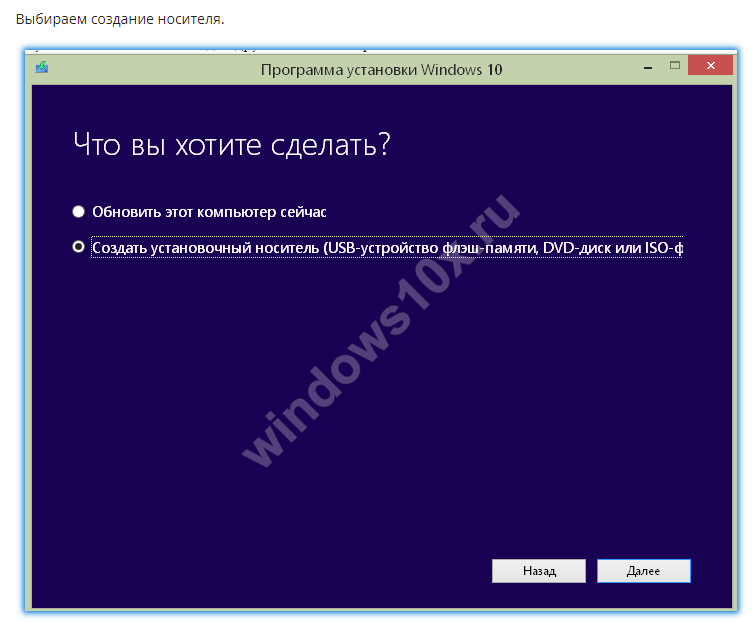
### Установка ОС Windows

Изображение выглядит как текст

Автоматически созданное описание

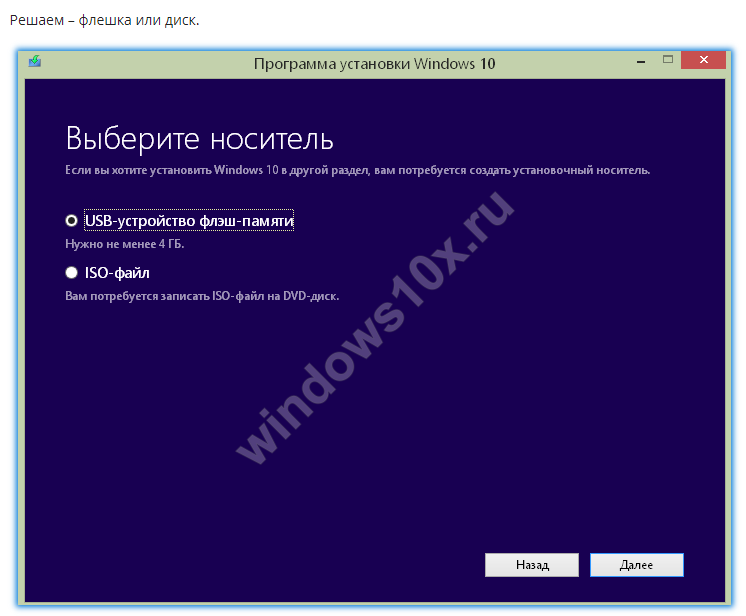
Изображение выглядит как текст

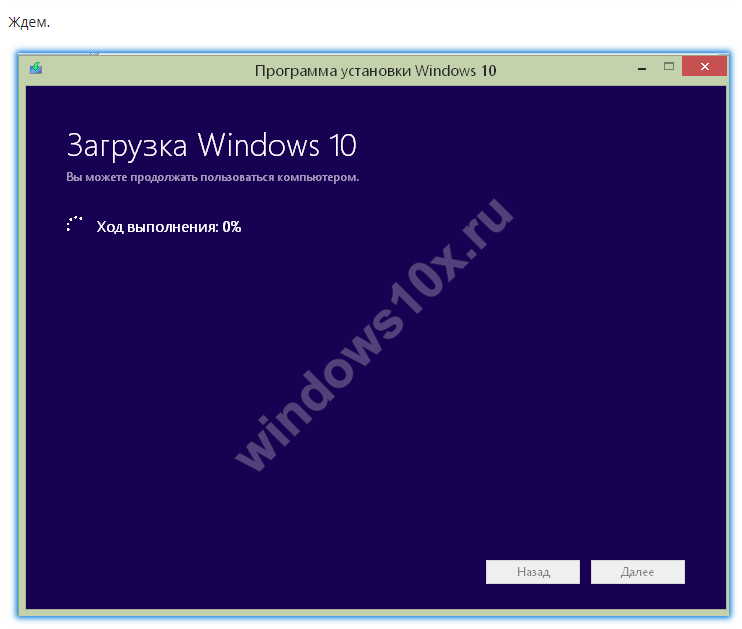
Автоматически созданное описание



Изображение выглядит как текст

Автоматически созданное описание





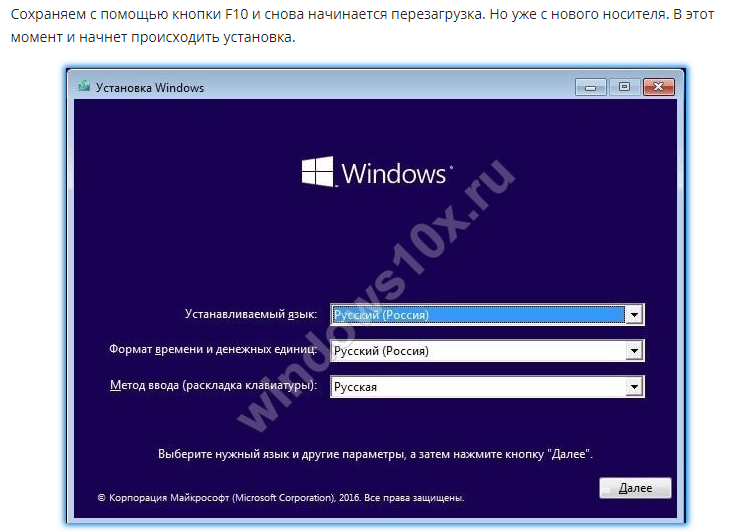


Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание



Изображение выглядит как текст, снимок экрана, электроника, дисплей

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, монитор

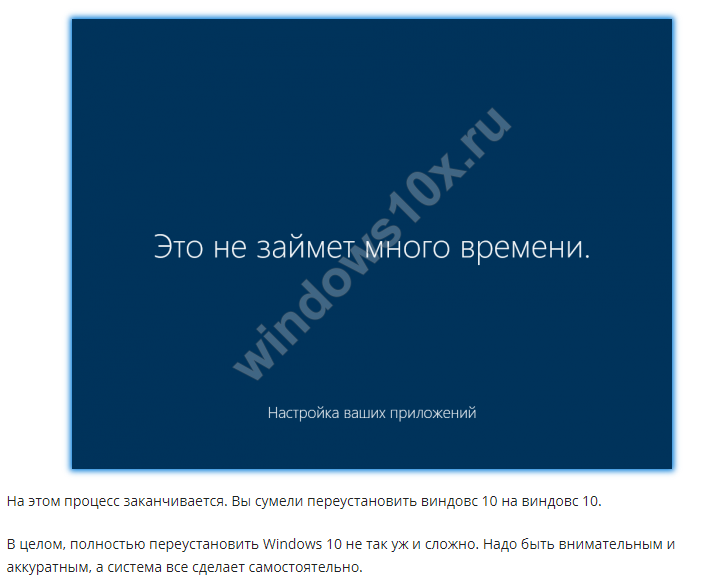
Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание



### Процесс загрузки MS-DOS

После загрузки в память системных файлов начинается конфигурирование системы. Из корневого каталога загружаемого диска считывается файл CONFIG.SYS и выполняются его директивы: загружаются в память и устанавливаются драйверы, указанные в директивах DEVICE; выделяются буферы DOS, описанные в директиве BUFFERS; расширяется системная таблица файлов в соответствии с директивой FILES и т. д. Таким образом, в процессе конфигурирования определяется объем оперативной памяти, занимаемый резидентной частью DOS и, соответственно, оставшийся объем свободной памяти для загрузки прикладных программ.

После окончания обработки файла CONFIG.SYS с диска считывается и загружается в память командный процессор COMMAND.СОМ. Этот файл COMMAND.COM отыскивается в корневом каталоге того диска, с которого выполнялась загрузка системы (т. е, А: или С:). Если, однако, в файл CONFIG.SYS включена директива SHELL с описанием пути к командному процессору, он будет считан из указанного в этой директиве каталога. Таким образом, возникает возможность "подставить" системе тот вариант файла COMMAND.COM, который нас больше устраивает. Необходимость в этом может возникнуть, например, из-за повреждения основного файла COMMAND.COM.

Командный процессор, получив управление, прежде всего считывает из корневого каталога загружаемого диска файл AUTOEXEC.BAT (если он существует) и выполняет включенные в него команды. Выполнив последнюю команду этого файла, командный процессор выводит на экран системный запрос и входит в цикл ожидания команд оператора. При необходимости загрузиться повторно (например, при "зависании" системы, когда она перестает функционировать и не отзывается на нажатия клавиш клавиатуры) можно выключить машину и тут же включить ее снова. Однако каждое включение компьютера, как и любого электронного или электрического прибора, отрицательно сказывается на его долговечности. Поэтому при зависании системы желательно не выключать машину, а воспользоваться возможностью "мягкой" перезагрузки, которая реализуется одновременным нажатием трех клавиш: "Ctrl", "Alt" и "Del". Эта команда заставляет ПЗУ BIOS выполнить всю описанную выше процедуру начальной загрузки за исключением тестирования. Жесткую перезагрузку можно выполнить, нажав клавишу "Reset".

В процессе работы на компьютере довольно часто возникает необходимость изменить конфигурацию DOS (например, загрузить новый драйвер или выгрузить установленный ранее), а также подкорректировать команды файла AUTOEXEC.BAT. Неудачное изменение состава файлов AUTOEXEC.BAT и, особенно, CONFIG. SYS часто приводит к неэффективной или неправильной работе системы. Для облегчения отладки процесса загрузки в состав MS DOS 6.2 включена возможность пошагового выполнения директив файла CONFIG.SYS и команд файла AUTOEXEC.BAT.  
  
Через короткое время после включения компьютера начинается чтение системных файлов и на экран выводится сообщение **Starting MS-DOS... (Стартует MS-DOS...)**  
  
Нажатие в этот момент определенных клавиш позволяет задать режим последующего выполнения файлов CONFIG.SYS и AUTOEXEC.BAT.  
  
Нажатие клавиши "F5" отменяет последующее чтение и анализ файлов CONFIG.SYS и AUTOEXEC.BAT, что в определенных случаях позволяет загрузить компьютер при наличии грубых ошибок в файле CONFIG.SYS.

### Процесс загрузки Windows

**Включение питания**

Не жалуется ли пользователь, что ничего не происходит, когда он нажимает кнопку питания? Если да, то сначала проверьте кабель.

Вот старый администраторский трюк на случай, если проходится иметь дело с выключенным из розетки компьютером при разговоре с пользователем по телефону. Пользователи часто не проверяют, включен ли их компьютер в розетку, а когда вы указываете им на такую возможность, возмущаются. Пользователь может воскликнуть: «Конечно, он включен!», но следует проверить, правда ли это. Попросите пользователя вынуть вилку из розетки и вставить заново, ссылаясь на необходимость «проверить полярность». Постарайтесь не хихикать. Просто невероятно, как часто пользователи отвечают: «О, сработало!»

Если дело не в вилке, то, возможно, в источнике бесперебойного питания — тоже уязвимом аппаратном компоненте системе. Источники бесперебойного питания стоят недорого, но замена аккумуляторов в них — утомительная и трудоемкая задача.

**Проверка оборудования и BIOS**

Если пользователь видит сообщение об ошибке в процессе POST или если компьютер просто зависает, прежде чем запускается операционная система, значит, проблема в оборудовании или в BIOS. Система выводит на экран сообщения об ошибках оборудования и BIOS и подает характерные звуковые сигналы. Некоторые ошибки BIOS выводятся в виде чисел и некогда все производители BIOS использовали одни и те же числа (те, что придумали в IBM), но теперь все изменилось. Теперь, чтобы интерпретировать номер ошибки, администратору требуется документация, пришедшая вместе с компьютером. Также можно поискать ее на Web-сайте производителя BIOS. Однако, скорее всего, на экране появится текст, а не числа, например Hard drive controller failure («Ошибка контроллера жесткого диска») или забавное сообщение Keyboard error, press F1 to continue («Ошибка клавиатуры, нажмите F1 для продолжения»).

Возможно, появится сообщение об ошибке, относящейся к проблемам с памятью. Когда-то память имела дополнительный компонент, называвшийся «чип четности», и частью теста BIOS была проверка на четность. Компоненты памяти больше не содержат чип четности, поскольку теперь в этом нет необходимости: производители памяти так продвинулись по пути совершенствования своих продуктов, что ошибки памяти стали крайне редки. Тем не менее после добавления в компьютер памяти при последующей загрузке может появиться сообщение об ошибке памяти. Сообщение содержит примерно такой текст: Mismatched memory information («Несоответствие параметров памяти»). В действительности это сообщение является подтверждением того, что система видит установленную память, но обнаружила, что она не соответствует общему количеству, записанному в CMOS.

Чтобы решить эту проблему, можно попробовать перезагрузить компьютер и войти в режим настройки BIOS. По своему опыту знаю, что стоит только нажать клавиши, необходимые для вызова окна BIOS, как сразу же автоматически выполняется корректный подсчет памяти, и все, что остается сделать, это выйти из программы настройки BIOS. Вход в программу настройки BIOS вызывает проверку счетчика памяти и его регулировку относительно имеющейся физической памяти.

Если после добавления в компьютер памяти система выдала сообщение об ошибке, в котором не говорится о разночтении в подсчете объема памяти, значит, проблема оказалась серьезнее. Система не распознает новую память. Эта ситуация почти всегда вызвана неаккуратными действиями при установке памяти, такими как использование неверного слота. В то же время мне доводилось видеть, как проблема вызывалась установкой памяти неверного типа (например, установкой плат памяти DRAM в старый компьютер с Enhanced Data Output, EDO), когда материнская плата не была предназначена для установки памяти смешанного типа, SIMM и DIMM, или когда материнская плата не воспринимала смешивание плат памяти с разными скоростями. Некоторые материнские платы при добавлении памяти нуждаются в изменении положения двухпозиционных переключателей и перемычек, хотя такие требования становятся все менее распространенными. Чтобы избежать подобных проблем, всегда перед добавлением памяти проверяйте документацию материнской платы.

Если во время POST обнаруживается ошибка жесткого диска, значит, работа предстоит немалая. Вообще-то я обнаружила, что в половине случаев проблема оказывается в контроллере, а не в самом диске, и замена контроллера позволяет диску нормально выполнять загрузку с сохранением всех данных неповрежденными (все гениальное просто!). Если контроллер жесткого диска первоначальной комплектации выходит из строя, не нужно сразу бежать за новой материнской платой. Достаточно купить новый контроллер. Вместо поиска встроенной микросхемы прочитайте в документации к материнской плате, что необходимо сделать, чтобы BIOS видел новую плату.

Если проблема действительно в диске, задача не ограничивается только заменой контроллера. Помимо замены диска придется переустановить операционную систему и приложения, а также выполнить восстановление данных из резервной копии, которая, конечно, датирована вчерашним днем, не правда ли?

**Управление передается записи Master Boot Record**

Далее компьютер начинает загрузку операционной системы. Во время установки программа загрузки Windows помещает данные на первый сектор главного раздела компьютера (загрузочный сектор). Эти данные — не что иное, как Master Boot Record, MBR (главная загрузочная запись), которая содержит исполняемые команды. Программа установки также копирует два файла, инициирующие начальный этап загрузки Windows, — Ntldr и Ntdetect — в корневой каталог загрузочного диска. Кроме того, Windows Setup копирует boot.ini, файл, содержащий параметры загрузки, в корневой каталог загрузочного диска.

Кроме исполняемых команд, MBR содержит таблицу, определяющую расположение главных разделов диска. При установке Windows необязательно делать так, чтобы системный раздел и загрузочный раздел совпадали, хотя это распространенный подход. Загрузочные файлы Windows расположены в системном разделе, а файлы операционной системы — в загрузочном (логика именования отстала от жизни).

Системный раздел содержит файлы привязки к аппаратным компонентам, которые необходимы для загрузки Windows, включая MBR. Этот раздел должен быть главным разделом и отмечается как активный. Это всегда диск 0, поскольку именно к этому диску обращается BIOS, чтобы передать процесс загрузки файлу MBR. Загрузочный раздел содержит файлы операционной системы (папка \%systemroot%) и файлы поддержки операционной системы (\%systemroot%System).

На последнем шаге аппаратной части начальной загрузки компьютер считывает файл MBR в память и передает управление компьютером коду в MBR. Исполняемый код просматривает таблицу главного раздела и ищет отметку, указывающую на загружаемый раздел. Когда MBR находит первый загружаемый раздел, он считывает первый сектор раздела, который является загрузочным сектором.

**Запуск Ntldr**

Код загрузочного сектора считывает Ntldr в память, чтобы запустить процесс начальной загрузки операционной системы. В Ntldr хранится доступный только для чтения код NTFS и FAT. Он начинает работать в реальном режиме, и первая выполняемая им задача заключается в переключении системы в некоторую форму защищенного режима. Более подробно об этих режимах рассказано во врезке «Реальный режим против защищенного». Этот первоначальный вариант защищенного режима не может в полной мере выполнять аппаратно-зависимых преобразований, которые обеспечивают защиту оборудования, — данная функция становится доступной, когда операционная система загрузится полностью.

Теперь вся физическая память доступна операционной системе, и компьютер работает как 32-разрядная машина. Ntldr делает возможной постраничную загрузку и создает таблицы страниц. Затем Ntldr считывает boot.ini из корневого каталога и, если на машине многовариантная загрузка или если выполнена настройка boot.ini на отображение меню, на экране появляется меню выбора вариантов загрузки. Если Ntldr отсутствует или поврежден, система выдаст сообщение об ошибке Ntldr is missing. Press Ctrl-Alt-Del to restart.

Не стоит зря тратить время, выполняя предложенное действие: после очередного цикла система вернется к тому же сообщению. Необходимо заменить Ntldr. Если создан загрузочный флоппи-диск, можно использовать его для копирования Ntldr в корневой каталог основного загрузочного диска (обычно C). Если Ntldr отсутствует, следует просто скопировать его. Если файл имеется на жестком диске, возможно, он поврежден. Для того чтобы заменить его, сначала нужно изменить его атрибут «только для чтения». Если загрузочного флоппи-диска под рукой нет, придется запустить программу Setup с компакт-диска Windows и выбрать Repair.

**Запуск Ntdetect**

Ntldr запускает Ntdetect, который запрашивает BIOS системы о данных конфигурации устройств. Система отправляет информацию, которую Ntdetect собирает в реестр и помещает в подразделы HKEY\_LOCAL\_MACHINEHARDWARE DESCRIPTION.

Если возникнет какая-нибудь проблема с Ntdetect (утрата или повреждение), возможно, система не выдаст никакого сообщения об ошибке. Обычно в таком случае процесс загрузки просто останавливается. Единственным действенным средством при утрате или повреждении файла Ntdetect является его замена. Необходимо воспользоваться для загрузки загрузочным флоппи-диском, затем скопировать Ntdetect с этого флоппи-диска в корневой каталог жесткого диска. Или же запустите Setup с компакт-диска Windows и выберите Repair.

**Запуск Ntoskrnl и загрузка HAL**

После того как Ntdetect закончит выполнение подпрограмм проверки оборудования, он передает процесс загрузки обратно файлу Ntldr, который запускает ntoskrnl.exe и загружает .dll-файл Hardware Abstraction Layer (HAL). (Оба файла расположены в папке \%systemroot%system32.) Ntoskrnl — это главный файл ядра Windows и исполнительных подсистем. Он содержит Executive, Kernel, Cache Manager, Memory Manager, Scheduler, Security Reference Monitor и другие. Именно Ntoskrnl приводит в действие Windows. Для работы Ntoskrnl необходим файл hal.dll, который содержит код, позволяющий оборудованию взаимодействовать с операционной системой.

На экране может появиться сообщение об ошибке, говорящее о наличии проблемы с Ntoskrnl, но почти всегда это сообщение фиктивно и появляется потому, что ссылка на папку в boot.ini не совпадает с именем папки, в которую были установлены системные файлы Windows.

Как правило, это означает, что кто-то переименовал папку \%systemroot% или создал новую папку и переместил в нее файлы Windows. В таком случае нужно переместить файлы обратно в то место, которое указано в boot.ini. Если boot.ini. был кем-то отредактирован, следует исправить эту ошибку.

**Загрузка драйверов и служб**

Теперь Ntldr загружает низкоуровневые системные службы и драйверы устройств, но службы не инициализируются — это происходит позже. На этом фаза начальной загрузки завершается и начинается процесс основной загрузки (load sequence или kernel phase).

При загрузке системных служб и драйверов устройств Ntldr следует определенному порядку. В процессе установки Windows драйверы и системные службы копируются на компьютер, а информация о них записывается в реестр. Данные в реестре представляют собой шестнадцатеричную запись, оканчивающуюся числом в круглых скобках. Это число и определяет порядок, в котором Ntldr загружает драйверы и системные службы. Для примера следует открыть реестр и перейти в раздел HKEY\_LOCAL\_MACHINESYSTEM CurrentControlSetServices. На экране появится длинный список служб и драйверов устройств. Выберите любой подраздел и взгляните на данные типа REG\_DWORD под именем Start.

* (0) означает, что служба загружается во время фазы основной загрузки.
* (1) означает, что служба загружается во время фазы инициализации (следующая фаза).
* (2) означает, что служба загружается во время фазы загрузки, предназначенной для служб.
* (3) означает, что служба включена, но не инициализируется (запуск службы осуществляется вручную через оснастку Services консоли Microsoft Management Console (MMC)).
* (4) означает, что служба отключена.

**Загрузка операционной системы**

Ntoskrnl начинает загрузку операционной системы. Инициализируется ядро Windows и загружаются и инициализируются подсистемы. Эти действия формируют базовые элементы, необходимые для завершения загрузки операционной системы. Загрузочные драйверы, которые модуль Ntldr загрузил ранее, теперь инициализируются, вслед за чем инициализируются оставшиеся драйверы и службы. Когда инициализируются драйверы первого уровня, может возникнуть проблема в виде ошибки STOP или Blue Screen of Death. Это почти всегда бывает в процессе первой загрузки после обновления какого-либо драйвера. Во время инициализации драйвера файлом Ntoskrnl операционная система отторгает его.

Для решения проблемы необходимо перезагрузить компьютер, нажать F8 для отображения меню дополнительных возможностей (Advanced Options) и выбрать вариант загрузки последней успешной конфигурации (Last Known Good Configuration), соответствующей предыдущей версии драйвера.

Теперь ядро Windows и исполняющие модули работоспособны. Программа Session Manager Subsystem (smss.exe) настраивает пользовательскую среду. Система выполняет сверку с реестром, для того чтобы иметь возможность начать загрузку оставшихся драйверов и программного обеспечения, которые необходимо добавить. Ядро операционной системы также загружает файлы kernel32.dll, gdi32.dll и user32.dll, которые обеспечивают программное обеспечение пользователя доступом к Win32 API.

**Регистрация компьютера в домене**

В то время когда загрузка ядра и инициализация драйверов еще не закончилась, компьютер регистрируется в домене. Используя учетную запись компьютера (уникальное имя с собственным паролем), компьютер открывает защищенный канал к контроллеру домена (DC). Все это происходит до того, как на экране появляется диалоговое окно для регистрации пользователя в системе.

Учетные записи компьютера используются между клиентскими компьютерами (включая автономные серверы) и контроллерами доменов. В пределах одного домена один и тот же процесс происходит с участием нескольких DC. Поэтому важен порядок, в котором включаются компьютеры после штатного отключения. Защищенный канал используется компьютерами для обмена информацией, необходимой для аутентификации и авторизации. Учетные записи компьютера повышают безопасность сети, гарантируя, что пытающийся отправить важную информацию компьютер действительно является членом домена.

В качестве дополнительной меры безопасности компьютеры (как и пользователи в сети с усиленной настройкой системы безопасности) должны периодически менять пароли. По умолчанию интервал смены пароля составляет 30 дней. Когда настает время менять пароль, компьютер генерирует новый пароль и отправляет его по защищенному каналу (доступ к которому он получил, используя предыдущий пароль) на ближайший DC. В дальнейшем для доступа к защищенному каналу компьютер должен использовать новый пароль.

DC немедленно обновляет свою базу данных и реплицирует изменение пароля компьютера на другие DC домена. Пароли учетных записей компьютера содержат отметку о событии первостепенной важности (Announce Immediately), поэтому они не дожидаются следующей репликации DC по расписанию. Иногда эти события могут вызывать заметное снижение производительности. Если у многих или у всех компьютеров домена срок действия паролей заканчивается в один и тот же день, работа, которую должны будут выполнить контроллеры домена, сразу же затормозит другие важные задачи DC, такие как аутентификация пользователей или выполнение запланированных репликаций. Ситуация может осложниться еще и в том случае, если DC предоставляет другие услуги, такие как, например, услуги сервера DNS. Можно изменить способ обслуживания паролей компьютера для домена, для организационной единицы (OU) и для индивидуального компьютера, хотя стремление повысить производительность путем настройки компьютеров по одному неэффективно. В следующей статье я планирую рассказать о методах изменения процедуры регистрации компьютера в домене.

**Загрузка служб регистрации пользователя**

Подсистема Win32 запускает winlogon.exe, которая выводит на экран диалоговое окно регистрации пользователя и загружает процесс Local Security Authority (lsass.exe). Начинается процесс регистрации, и пользователь должен ввести имя и пароль в диалоговом окне Windows Log On To. Если пользователь указывает правильные имя и пароль, система завершает процесс регистрации, и пользователь может начинать работать. В этот момент загрузка Windows окончена, а текущие параметры загрузки сохраняются в так называемой последней успешной конфигурации (Last Known Good Configuration). Нужно иметь в виду, что успешная регистрация пользователя необходима для сохранения Last Known Good Configuration.

### Процесс загрузки Linux

При загрузке компьютера происходит последовательная передача управления от системной прошивки компьютера ([BIOS](https://ru.wikipedia.org/wiki/BIOS)) к [загрузчику](https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%B3%D1%80%D1%83%D0%B7%D1%87%D0%B8%D0%BA_%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%BE%D0%B9_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B), а от него — к [ядру](https://ru.wikipedia.org/wiki/%D0%AF%D0%B4%D1%80%D0%BE_%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%BE%D0%B9_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B). Затем ядро запускает [планировщик](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D1%81%D0%BF%D0%B5%D1%82%D1%87%D0%B5%D1%80_%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%BE%D0%B9_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B) (для реализации многозадачности) и выполняет программу [init](https://ru.wikipedia.org/wiki/Init" \o "Init) (которая настраивает пользовательское окружение и позволяет осуществлять взаимодействие с пользователем и вход в систему), после чего ядро переходит в состояние бездействия до тех пор, пока не получит внешний вызов.

Основные этапы загрузки:

1. Системная прошивка компьютера выполняет первичную проверку и инициализацию [аппаратного обеспечения](https://ru.wikipedia.org/wiki/%D0%90%D0%BF%D0%BF%D0%B0%D1%80%D0%B0%D1%82%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5).
2. В случае BIOS прошивка загружает в оперативную память и выполняет загрузочный код с одного из [разделов](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B7%D0%B4%D0%B5%D0%BB_%D0%B4%D0%B8%D1%81%D0%BA%D0%B0) заданного загрузочного устройства, который содержит *фазу 1* загрузчика Linux. Фаза 1 загружает *фазу 2* (значительный по размеру код загрузчика). Некоторые загрузчики могут использовать для этого промежуточный этап (под названием *фаза 1,5*), поскольку современные диски большого объёма могут некорректно считываться без дальнейшего кода. В случае UEFI запускается загрузчик загруженный со служебного раздела (EFS), который выбирается согласно настройкам приоритета загрузки определенного в энергонезависимой памяти компьютера. При этом возможна загрузка не только специализированного загрузчика, но можно загрузить и непосредственно ядро Linux (для этого ядро должно быть собрано с опцией EFI\_STUB).
3. Загрузчик зачастую предлагает пользователю меню с доступными вариантами загрузки. После выбора или после заданного тайм-аута загрузчик загружает ядро.
4. Загруженное ядро распаковывается в памяти, настраивает системные функции, такие как работа необходимого оборудования и управление страницами памяти, после чего делает вызов start\_kernel().
5. После этого start\_kernel() выполняет основную настройку системы (прерывания, остальные функции управления памятью, инициализацию устройств, драйверов и т. д.), а потом порождает [процесс бездействия](https://ru.wikipedia.org/wiki/%D0%91%D0%B5%D0%B7%D0%B4%D0%B5%D0%B9%D1%81%D1%82%D0%B2%D0%B8%D0%B5_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B), [диспетчер](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D1%81%D0%BF%D0%B5%D1%82%D1%87%D0%B5%D1%80_%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%BE%D0%B9_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B) и отдельно от них — процесс [init](https://ru.wikipedia.org/wiki/Init" \o "Init) (выполняющийся в пользовательском пространстве).
6. Планировщик начинает более эффективно управлять системой, в то время как ядро переходит к бездействию.
7. Процесс [init](https://ru.wikipedia.org/wiki/Init" \o "Init) выполняет необходимые сценарии, которые настраивают все службы и структуры, не относящиеся к уровню ядра, в результате чего будет создано пользовательское окружение, и пользователю будет предоставлен экран входа в систему.

Когда происходит завершение работы, [init](https://ru.wikipedia.org/wiki/Init" \o "Init) вызывается для управляемого закрытия программ пользовательского уровня, тоже согласно сценариям. После этого [init](https://ru.wikipedia.org/wiki/Init" \o "Init) закрывается, а ядро производит своё собственное завершение работы.

### Процесс загрузки QNX

Существующий релиз допускает два различных способа загрузки QNX:

1. непосредственно с дискового раздела QNX средствами boot loader.
2. косвенно через FAT-раздел посредством использования драйвера DOS.

**QNX boot loader**

QNX boot loader осуществляет загрузку образа из файла /.boot непосредственно из дискового раздела QNX (или /.altboot, если Вы нажимаете клавишу ESC при появлении сообщения о возможности загрузки alternate OS). Если Вы собираетесь создать новый загрузочный образ, то можно будет его скопировать (файл .ifs) в /.altboot и, в этом случае, нажав ESC во время загрузки, можно будет проверить на "загружаемость" новый образ.

**Драйвер DOS**

Загрузка с использованием DOS-драйвера предоставляет гораздо меньше гибких возможностей по сравнению с QNX boot loader, в силу того, что при загрузке использует только набор элементарных функций ввода-вывода, обеспечиваемых DOS. Но зато Вы можете использовать возможности меню, обеспечиваемые config.sys.

Чтобы добавить новый загрузочный образ, создайте новую строку меню и сконфигурируйте соответствующий блок команд для этого пункта меню.

Независимо от того, какой метод загрузки используется, главная идея остается неизменной. Как только образ (.ifs) загрузится в память, управление передается стартовому скрипту, содержащемуся также в образе. На этом этапе завершается распаковка образа, собирается информацию о системе, передаваемая ядру, ядро системы размещается в памяти, после чего ему передается управление.

**Что такое сценарий загрузки?**

В отличие от большинства операционных систем, загрузочный образ. ifs - это нечто большее, чем ядро и стартовый код. Образ также может содержать как приложения и библиотеки, так и простой стартовый shell-скрипт.

Новый образ можно построить, используя утилиту mkifs (Make Image FlieSystem), считывающую из текстового файла сценарий загрузки, описывающий содержимое образа, а также последовательность действий, описанную в стартовом shell-скрипте.

В установленной системе, в каталоге /boot/build, уже есть примеры готовых сценариев загрузки. Мы попробуем сейчас проанализировать содержимое файла qnxbasedma.build, одного из стандартных сценариев, который находится в этом же каталоге.

Первая часть этого файла предоставляет mkifs информацию о том, для какого процессора (архитектуры) строится образ (x86) и какой IPL (Initial Program Loader) используется (bios). Далее описывается, какая стартовая программа используется (startup-bios) и какое микроядро+менеджер процессов (procnto). При создании нового загрузочного образа для стандартного ПК x86, Вам не придется изменять содержимое этой части сценария загрузки. Зато следующий раздел представляет определенный интерес.

Раздел сценария, начинающийся с описания [+script] - стартовый скрипт, который запускается после передачи управления ядру системы. Этот скрипт содержит последовательность команд или программ, которые должны быть выполнены в процессе загрузки системы.

В случае нашего сценария первое приложение - slogger, программа, осуществляющая ведение протоколов (логов) событий и сообщений, возникающих во время работы системы. После slogger запускается seedres - приложение, осуществляющее сбор информации об оборудовании, "железе" (аппаратные прерывания, адреса портов ввода-вывода и т.д.) для базы данных системных ресурсов. Затем следует pci-bios. Сервер pci-bios обеспечивает доступ к сервисам PCI-шины для приложений и драйверов устройств. Это очень критичное по отношению к "железу" приложение, и, чтобы обеспечить его нормальное функционирование, оно должно выполняться на современном ПК, имеющем PCI-шину. Чтобы оставшиеся приложения не начали выполняться ранее, нежели чем сервер PCI будет готов к работе, сюда же включена команда waitfor, ожидающая появление в файловой системе устройства /dev/pci.

В завершение работы стартового скрипта выполняется программа diskboot. diskboot определяет тип контроллера жестких дисков и проверяет все разделы жесткого диска на наличие способной к загрузке ОС QNX.

В следующих строках сценария, идущих после скобки} , завершающей стартовый скрипт, перечислены библиотеки и бинарные файлы, включаемые в загрузочный образ. Вначале перечисляются библиотеки, т.к. они являются разделяемыми (shared) элементами. Строка [data=c] означает, что описанные ниже нее бинарные файлы требуют уникальные сегменты данных, но код может быть разделяемым (shared). Если Вы выполните команду ls -l /proc/boot на работающей системе, Вы увидите список файлов, которые изначально были включены в образ, с которого была загружена система.

**Как создать загрузочный образ?**

В завершение этой статьи, попробуем внести небольшие изменения в qnxbasedma.build, построить образ и попытаться загрузить его.

Создайте локальную копию сценария в Вашем домашнем каталоге:

cp /boot/build/qnxbasedma.build $HOME

Теперь, используя Ваш любимый редактор, добавьте строку в сценарий. (Будьте осторожны при использовании Photon Editor (ped) - он может быть не настроен и сохранять данные о форматировании в конце файла). В блоке startup-script добавьте строку, обеспечивающую при загрузке вывод сообщения на консоль.

Мы добавили строку с командой display\_msg. В результате, перед запуском программы diskboot, на консоль будет выводиться сообщение. Заодно, увидев его, Вы убедитесь, что загружается именно созданный Вами образ.

Теперь нужно запустить программу mkifs и инсталлировать результирующий файл нового образа. Откройте окно терминала (pterm), перейдите в домашний каталог и выполните команду:

% mkifs -v qnxbasedma.build custom.ifs

В процессе работы mkifs, на терминал будет выводиться информация о файлах, включаемых в образ. По завершении работы Вы получите файл образа custom.ifs, который можно будет скопировать в /.altboot или в каталог DOS (обычно в /fs/hd0-dos/Program Files/qnx/boot/fs). Если Вы используете DOS для загрузки Вашего образа, приготовьтесь отредактировать config.sys под DOS, НЕ ПОД QNX - многие редакторы в среде QNX сохранят файл config.sys в формате текстового файла UNIX, который DOS впоследствие "не поймет".

Перезагрузив систему, Вы увидите на экране Ваше сообщение, которое появится в процессе загрузки системы.

### Процесс загрузки UNIX

**Изображение выглядит как текст

Автоматически созданное описание**

**Изображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

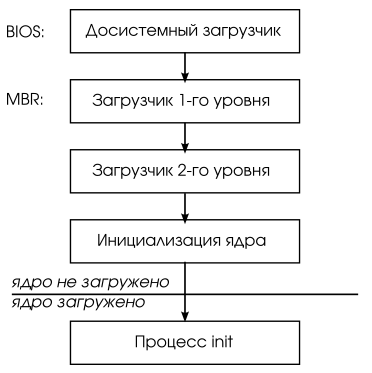
Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описание**

### Процесс загрузки Mac-OS

Startup Chime

Этот звук неизменно сопровождает каждый запуск вашего Mac, сигнализируя об инициализации и запуске BootROM (железа и прошивки-биоса), который хранится во флеш-памяти материнской платы и является, по сути дела, миниатюрной операционной системой. После подачи питания прошивка запускает тесты POST (Power-On Self Test), проверяющие работу процессора, памяти, [Wi-Fi](http://macovod.net/tag/wi-fi/), Ethernet и разных периферийных интерфейсов вроде USB, FireWire и Bluetooth.

Если ваше оборудование проходит POST, то звучит тот самый звук при запуске системы. Спустя секунды экраны всех подключенных к компьютеру мониторов «окрашиваются» в светло-серый цвет, а биос начинает искать загрузчик.

Далее происходит 2 вещи. Во-первых, загружается ядро Mac OS X, во-вторых — его расширения, так называемые кексты (kext, kernel extension), поэтому ядро в состоянии «взвалить на плечи» систему и продолжить её запуск самостоятельно. Если ядро успешно загружено, то вы увидите под логотипом Apple вращающуюся шестеренку.

Также, в этот момент стартует launchd, который является родительским процессом для других процессов в системе и «помогает» им запускаться.

Обычно в Windows синий экран не предвещает ничего хорошего, однако, в Mac OS X светло-синий экран говорит о том, что launchd помог запустить другой не менее важный процесс — WindowServer, использующийся для отрисовки пользовательского интерфейса.

А еще через некоторое время опять-таки launchd запустит loginwindow.app, являющийся одновременно фоновым процессом и приложением, либо отобразит рабочий стол.

### Команды ОС Windows

Хоть ОС Windows – ОС с визуальным интерфейсом, все команды выполняются в командной строке. Эту командную строку можно запустить через сочетание клавиш WIN+R. Внутри пишем название консоли “cmd”, нажимаем enter.

Запускается оконное приложение консоли. Внутри мы можем писать различные команды. Например:

[CD](https://ab57.ru/cmdlist/cd.html) - смена каталога (Change Directory)

[DEL](https://ab57.ru/cmdlist/del.html) - удаление одного или нескольких файлов

[HELP](https://ab57.ru/cmdlist/help.html) - вызов справки командной строки Windows

[PING](https://ab57.ru/cmdlist/ping.html) утилита проверки доступности узла

[PAUSE](https://ab57.ru/cmdlist/pause.html) - пауза при выполнении командного файла

[EXIT](https://ab57.ru/cmdlist/exit.html) - выход из процедуры или командного файла

[ECHO](https://ab57.ru/cmdlist/echo.html) - вывод текста на экран консоли

Изображение выглядит как текст, табличка, снимок экрана

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Справка по всем командам командной строки: [Здесь](https://ab57.ru/cmdlist.html)

**Изображение выглядит как текст

Автоматически созданное описание**

### Реестр ОС Windows

**Реестр Windows** — иерархически построенная [база данных](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) параметров и настроек в большинстве операционных систем [Microsoft Windows](https://ru.wikipedia.org/wiki/Microsoft_Windows).

Реестр содержит информацию и настройки для [аппаратного обеспечения](https://ru.wikipedia.org/wiki/%D0%90%D0%BF%D0%BF%D0%B0%D1%80%D0%B0%D1%82%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5), [программного обеспечения](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5), профилей пользователей, предустановки. Большинство изменений в [Панели управления](https://ru.wikipedia.org/wiki/%D0%9F%D0%B0%D0%BD%D0%B5%D0%BB%D1%8C_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F_(Windows)), ассоциации файлов, системные политики, список установленного ПО фиксируются в реестре.

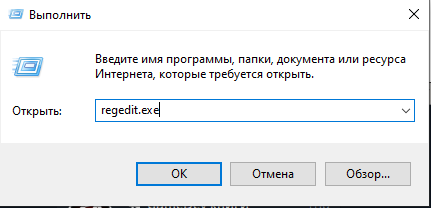
Реестр в том виде, как его использует Windows и как видит его пользователь в процессе использования программ работы с реестром, формируется из различных данных. Чтобы получилось то, что видит пользователь, редактируя реестр, происходит следующее.

Вначале, в процессе установки (инсталляции) и настройки Windows, на диске формируются файлы, в которых хранится часть данных, относящихся к конфигурации системы.

Затем, в процессе каждой загрузки системы, а также в процессе каждого входа и выхода каждого из пользователей, формируется некая виртуальная сущность, называемая «реестром» — объект REGISTRY\. Данные для формирования «реестра» частично берутся из тех самых файлов (Software, System …), частично из информации, собранной ntdetect при загрузке (HKLM\Hardware\Description).

То есть часть данных реестра хранится в файлах, а часть данных формируется в процессе загрузки Windows.

Для редактирования, просмотра и изучения реестра стандартными средствами Windows (программы regedit.exe и regedt32.exe) доступны именно ветки реестра. После редактирования реестра и/или внесения в него изменений эти изменения сразу записываются в файлы.



Однако есть программы сторонних разработчиков, которые позволяют работать непосредственно с файлами.

**Изображение выглядит как текст

Автоматически созданное описание**

### Работа с потоками

**Изображение выглядит как текст

Автоматически созданное описание**

**Изображение выглядит как текст

Автоматически созданное описание**

**Индивидуальное задание. Вариант 7**

**Изображение выглядит как текст

Автоматически созданное описание**

****

**Изображение выглядит как текст

Автоматически созданное описание**

**Изображение выглядит как текст

Автоматически созданное описание**

### Выводы

Выполняя лабораторную работу №1, я разобрался с устройством разных операционных систем, архитектурой загрузки ОС в оперативную память. Научился устанавливать ОС Windows 10. Ознакомился с командной строкой, узнал возможности её использования. С помощью команд можно выполнять любые действия, которые доступны из визуального интерфейса.

Работу с потоками реализовал при помощи библиотеки <thread>. Она позволяет запускать потоки на функциях и методах и передавать туда параметры. Передача данных через динамический массив. Библиотека имеет все необходимые деструкторы, очищать память не требуется.

### Список литературы

[**http://junior.ru/wwwexam/opsys/msdos\_z.htm**](http://junior.ru/wwwexam/opsys/msdos_z.htm)

[**https://habr.com/ru/post/113350/**](https://habr.com/ru/post/113350/)

[**https://losst.ru/protsess-zagruzki-linux**](https://losst.ru/protsess-zagruzki-linux#%D0%9F%D0%B5%D1%80%D0%B2%D1%8B%D0%B9_%D1%8D%D1%82%D0%B0%D0%BF_%D0%B2%D0%BA%D0%BB%D1%8E%D1%87%D0%B5%D0%BD%D0%B8%D0%B5_%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%B0)

[**https://studfile.net/preview/9768333/page:26/**](https://studfile.net/preview/9768333/page:26/)

[**https://pandia.ru/text/78/459/53389.php**](https://pandia.ru/text/78/459/53389.php)

[**https://ab57.ru/cmdlist.html**](https://ab57.ru/cmdlist.html)