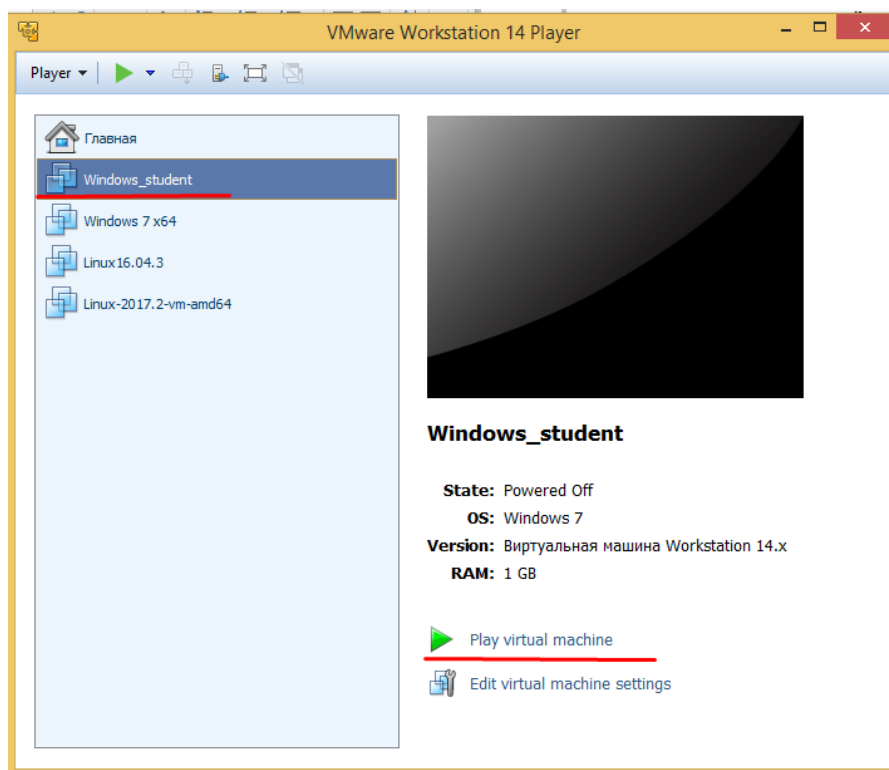


## Работа с виртуальной машиной VMWare Player

VMware Workstation Player (ранее VMware Player) — бесплатный для некоммерческого использования программный продукт, на основе виртуальной машины VMware Workstation, но с ограниченной функциональностью, предназначенный для запуска образов виртуальных машин, созданных в других продуктах VMware, а также в Microsoft VirtualPC и Symantec LiveState Recovery. Начиная с версии 3.0 VMware Player позволяет также создавать образы виртуальных машин. Ограничение функциональности теперь касается в основном функций, предназначенных для IT-специалистов и разработчиков ПО. Например, отсутствует возможность тонкого настраивания виртуальных сетевых адаптеров через Virtual Network Editor.



Для запуска виртуальной машины выбираем `Windows_student` и нажимаем *Play virtual machine*

Ошибка виртуализации может возникать, если в BIOS (UEFI) не включена поддержка аппаратной виртуализации Intel-VT или AMD-V.

Для возврата курсора в хост машину используйте `Ctrl+Alt`.

## Работа с программой Diskpart

Diskpart — консольная утилита для управления разделами жестких дисков. Впервые появилась в линейке операционных систем Windows NT, начиная с Windows 2000, заменив собой fdisk, который использовался в основанных на MS-DOS операционных системах.

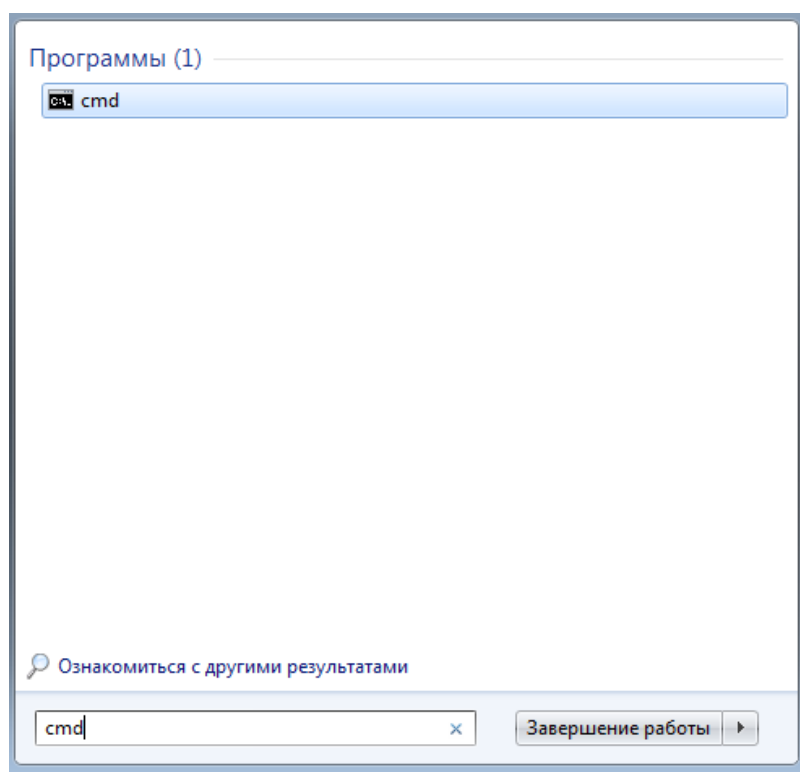
Является достаточно мощным инструментом для работы с дисковыми данными. При отсутствии определенных знаний и понимания того, что выполняют определенные команды, можно вред компьютеру, вплоть до потери информации, находящейся на дисковом накопителе.

Для работы будем использовать виртуальную машину с Windows 7.

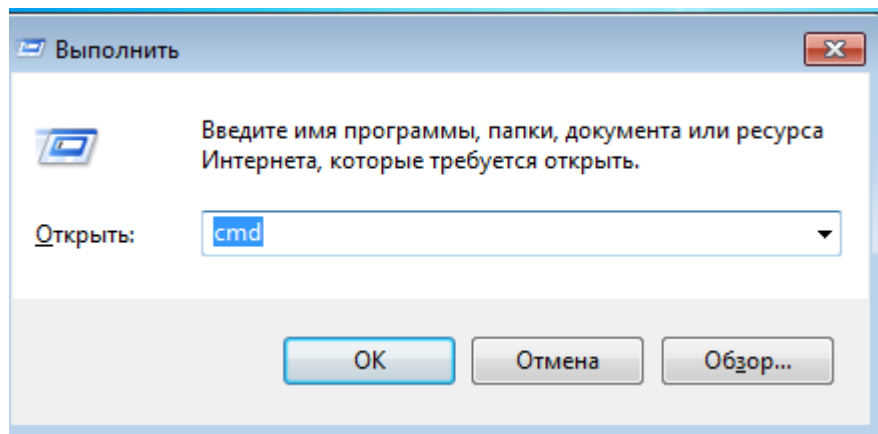
Для работы программы необходимы права администратора.

Запустите виртуальную машины VMWare Player и выберете Windows\_student.

Для запуска программы diskpart необходимо нажать на «Пуск» и в строке поиска написать **cmd** и выбрать соответствующую программу.

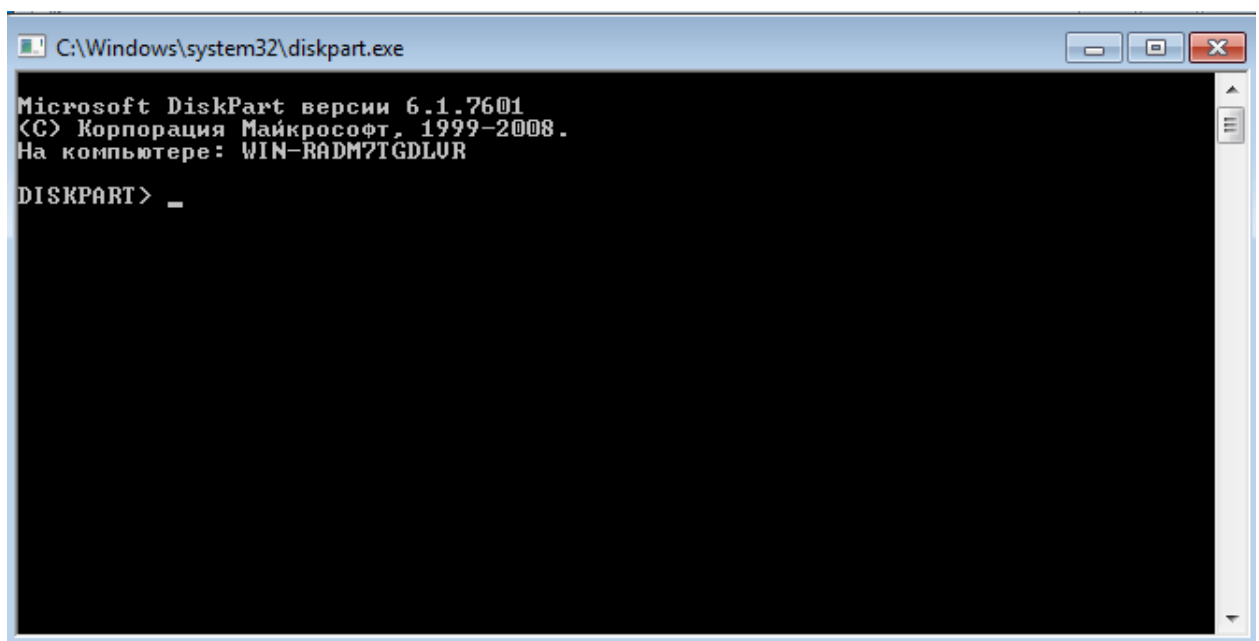


Также можно нажать комбинацию клавиш Win+R и в появившемся окне написать **cmd** и нажать ОК.



В появившемся окне необходимо ввести **diskpart** и нажать enter.

На предложение запустить программу от имени администратора согласиться.



Основные команды DISKPART:

Для перечисления доступных дисков, разделов и томов пред назначены команды:

**list disk** — перечисляет все физические жесткие диски компьютера;

**list volume** — перечисляет все дисковые тома (включая разделы жесткого диска и логические диски);

**list partition** — перечисляет разделы на диске, который находится в фокусе.

При использовании команд **list** рядом с диском, томом или разделом в фокусе появляется звездочка (\*). Вы выбираете диск, том, или раздел по его номеру или букве диска, например disk 0, partition 1, volume 2 или volume D.

**ЗАДАНИЕ 1.** Попробуйте все перечисленные выше команды, посмотрите, чем они отличаются. Ниже пример выполнения **list disk** в программе diskpart.

```
DISKPART> list disk

Диск ###  Состояние  Размер  Свободно  Дин  GPT
Диск 0     В сети        20 Гбайт  1024 Кбайт

DISKPART>
```

Полный список команд можно посмотреть через команду HELP [команда], команды представлены ниже

ACTIVE	Пометка выбранного раздела как активного.
ADD	Добавление зеркала к простому тому.
ASSIGN	Назначение имени или точки подключения выбранному тому.
ATTRIBUTES	Работа с атрибутами тома или диска.
ATTACH	Присоединяет файл виртуального диска.
AUTOMOUNT	Включение и отключение автоматического подключения базовых томов.
BREAK	Разбиение зеркального набора.
CLEAN	Очистка сведений о конфигурации или всех данных на диске.
COMPACT	Попытки уменьшения физического размера файла.
CONVERT	Преобразование форматов диска.
CREATE	Создание тома, раздела или виртуального диска.
DELETE	Удалить объект.
DETAIL	Просмотр параметров объекта.
DETACH	Отсоединяет файл виртуального диска.
EXIT	Завершение работы DiskPart.
EXTEND	Расширить том.
EXPAND	Увеличение максимального доступного пространства на виртуальном диске.

FILESYSTEMS	Отображение текущей и поддерживаемой файловых систем для тома.
FORMAT	Форматирование данного тома или раздела.
GPT	Назначение атрибутов выбранному GPT-разделу.
HELP	Отображение списка команд.
IMPORT	Импорт группы дисков.
INACTIVE	Пометка выбранного раздела как неактивного.
LIST	- Отображение списка объектов.
MERGE	Объединение дочернего диска с родительскими.
ONLINE	Перевод объекта, помеченного как "вне сети", в состояние "в сети".
OFFLINE	Перевод в состояние "вне сети" объекта, помеченного как "в сети".
RECOVER	Обновление состояния всех дисков выбранного пакета. Попытка восстановить диски неправильного пакета и повторная синхронизация зеркальных томов и томов RAID5 с устаревшим плексом или данными четности.
REM	Не выполняет никаких действий. Используется для комментирования сценариев.
REMOVE	Удаление имени диска или точки подключения.
REPAIR	Восстановление тома RAID-5 с отказавшим участником.
RESCAN	Поиск дисков и томов на компьютере.
RETAIN	Размещение служебного раздела на простом томе.
SAN	Отображение или установка политики SAN для текущей загруженной ОС.
SELECT	Установка фокуса на объект.
SETID	Изменение типа раздела.
SHRINK	Уменьшение размера выбранного тома.

UNIQUEID	Отображение или установка кода таблицы разделов GUID (GPT) или подписи основной загрузочной записи (MBR) диска.
----------	---

## ЗАДАНИЕ 2.

Установить имя тому с размером 500 MB в соответствии с первой буквой Вашей фамилии.

Для этого посмотрите доступные тома командой **list volume**, выберете целевой том командой **select volume N**, где N – номер тома в соответствии с выводом команды list volume.

Присвойте имя выбранному тому командой assign letter k, где k –буква первой буквы Вашей фамилии.

Посмотрите вывод list volume снова. Сохраните скринтош и добавьте к нему описание.

```
DISKPART> list volume
```

Том	###	Имя	Метка	ФС	Тип	Размер	Состояние	Сведения
Том 0		D	GSP1RMCPRUO	UDF	DUD-ROM	2213 Мб	Исправен	
Том 1		C		NTFS	Раздел	19 Гб	Исправен	Системны
Том 2		E	Новый том	NTFS	Раздел	499 Мб	Исправен	

```
DISKPART> select volume 2
Выбран том 2.
DISKPART> assign letter k
DiskPart: назначение имени диска или точки подключения выполнено успешно.
DISKPART> list volume
```

Том	###	Имя	Метка	ФС	Тип	Размер	Состояние	Сведения
Том 0		D	GSP1RMCPRUO	UDF	DUD-ROM	2213 Мб	Исправен	
Том 1		C		NTFS	Раздел	19 Гб	Исправен	Системны
* Том 2		K	Новый том	NTFS	Раздел	499 Мб	Исправен	

## ЗАДАНИЕ 3.

Отформатируйте раздел 500MB в fat32, а после снова в ntfs. После каждого раза сохраняйте скриншоты. Убедитесь, что у вас выбран именно том с размером 500 MB (стоит звездочка напротив имени тома) и потом используйте команды format fs=fat32 и fs=ntfs. Для такого маленького диска можно

использовать полноценное форматирование. Либо добавить в конец команды quick – для быстрого форматирования.

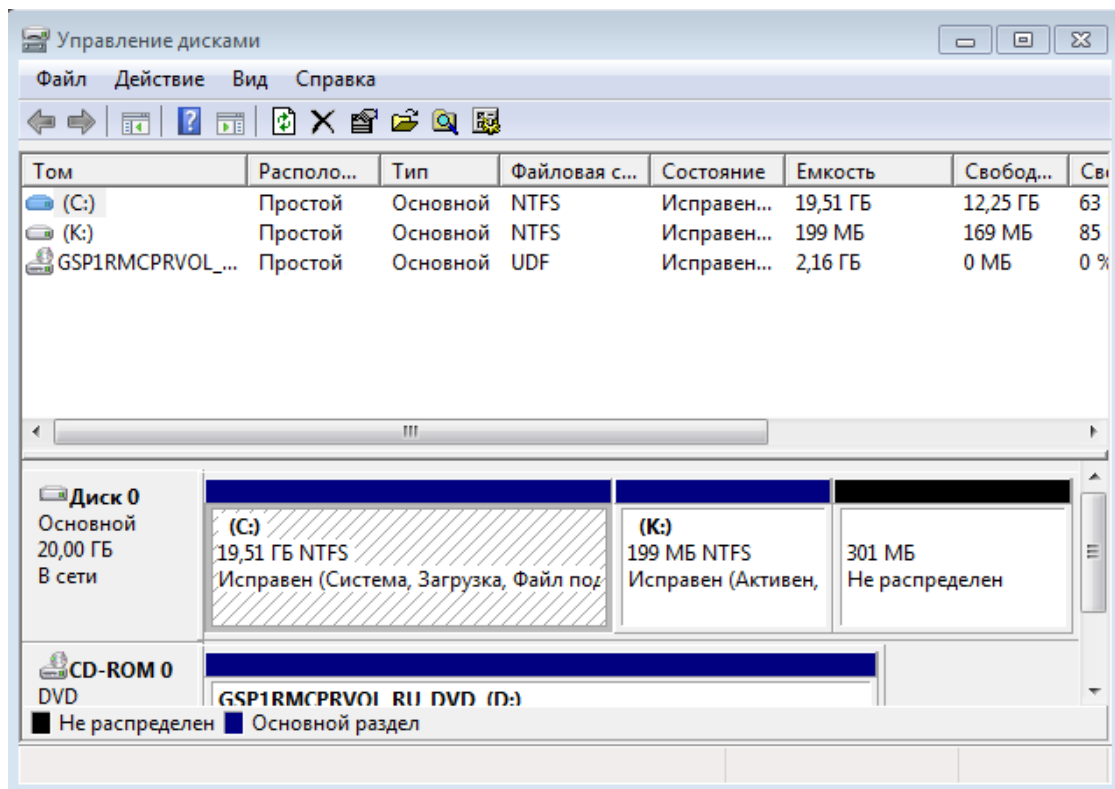
#### ЗАДАНИЕ 4.

Разбить диск 500 MB на диски по 300 и 200 MB. После чего снова объединить их обратно.

1. В командной строке DISKPART введите select volume <volumenumber>. Выбирает простой том volumenumber, который требуется сжать (наш том, размером 500 MB)

2. В командной строке DISKPART введите shrink [desired=<desiredsize>] [minimum=<minimumsize>]. Сжимает выбранный том до размера desiredsize в мегабайтах (MB), если возможно, или до размера minimumsize, если размер desiredsize слишком велик.

Откройте «Управление дисками» - через Панель управления или Win+R и напишите **diskmgmt.msc**



Можно наглядно увидеть, что теперь у нас есть основной диск C, том K, ужатый до 199 MB и новый, нераспределенный том 301 MB.

Командой **list disk** также можно увидеть, что у нас есть свободная нерасмеченная область на диске 0.

```
DISKPART> list disk
```

Диск ###	Состояние	Размер	Свободно	Дин	GPT
* Диск 0	В сети	20 Гбайт	301 Мбайт		

```
DISKPART>
```

Для создания раздела, воспользуемся командой **create partition primary**.

Введите снова **list volume**

```
DISKPART> create partition primary
DiskPart: указанный раздел успешно создан.
DISKPART> list volume
```

Том	###	Имя	Метка	ФС	Тип	Размер	Состояние	Сведения
Том 0	D	GSP1RMCPRV0	UDF	DUD-ROM	2213 Мб	Исправен		
Том 1	C		NTFS	Раздел	19 Гб	Исправен		Системны
Том 2	K		NTFS	Раздел	199 Мб	Исправен		
* Том 3			RAW	Раздел	301 Мб	Исправен		

```
DISKPART>
```

Как видите, у нас создался основной раздел (Том 3) с файловой системой RAW.

Теперь отформатируйте этот раздел в ФС NTFS, как в задании 3, только убедитесь вначале, что у Вас выбран этот том (стоит звездочка напротив него).

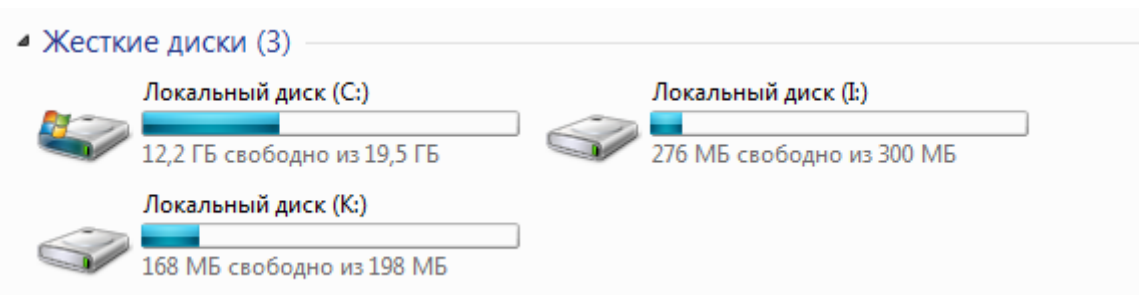
Присвойте получившемуся диску первую букву Вашего имени.

```
DISKPART> format fs=ntfs
Завершено (в процентах): 100
Программа DiskPart успешно отформатировала том.
DISKPART> assign letter=I
DiskPart: назначение имени диска или точки подключения выполнено успешно.
DISKPART> list volume
```

Том	###	Имя	Метка	ФС	Тип	Размер	Состояние	Сведения
Том 0	D	GSP1RMCPRV0	UDF	DUD-ROM	2213 Мб	Исправен		
Том 1	C		NTFS	Раздел	19 Гб	Исправен		Системны
Том 2	K		NTFS	Раздел	199 Мб	Исправен		
* Том 3	I		NTFS	Раздел	301 Мб	Исправен		

Мы создали новый том с ФС ntfs, размером 301 MB.





## ЗАДАНИЕ 5.

Вернуть все как было

Удаляем тома 301 МВ и 199 МВ используя команду **delete**

```
DISKPART> list part

Раздел    ###  Тип              Размер    Смещение
-----
Раздел 1   1     Основной         19 Гб     1024 Кб
Раздел 2   2     Основной         199 Мб     19 Гб
* Раздел 3  3     Основной         301 Мб     19 Гб

DISKPART> delete vol

DiskPart: том успешно удален.

DISKPART> list part

Раздел    ###  Тип              Размер    Смещение
-----
Раздел 1   1     Основной         19 Гб     1024 Кб
Раздел 2   2     Основной         199 Мб     19 Гб

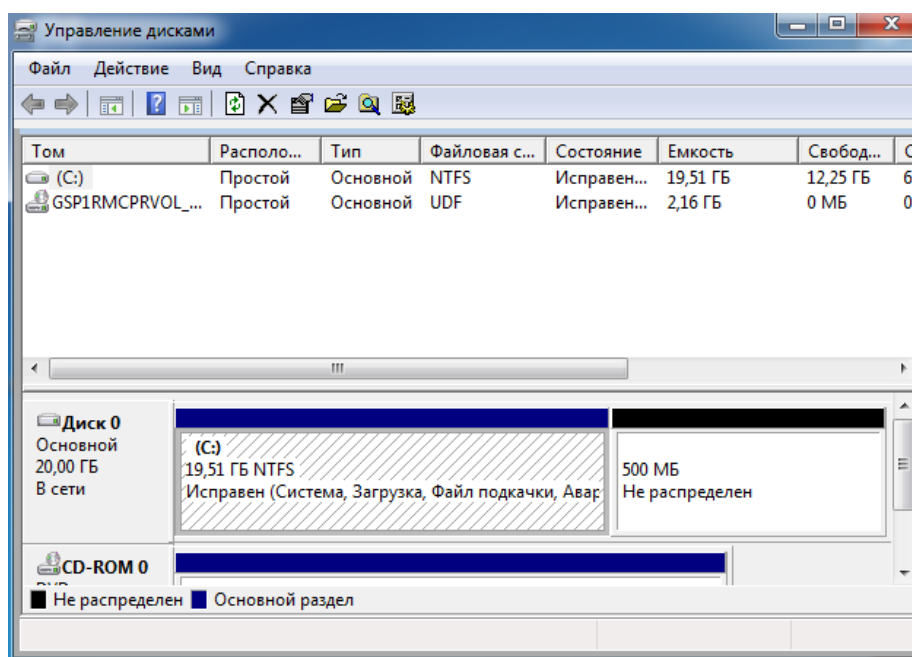
DISKPART> select part 2

Выбран раздел 2.

DISKPART> delete part

DiskPart: выбранный раздел успешно удален.
```

Теперь в «Управление дисками» видна неразмеченная область в 500МВ



Создадим новый раздел как в задании 4 и отформатируем его в NTFS, присвоив любую доступную букву, скажем E.

```
DISKPART> create partition primary
DiskPart: указанный раздел успешно создан.
DISKPART> list partition

Раздел    ###    Тип                Размер    Смещение
-----
Раздел 1   1       Основной          19 Gб    1024 Kб
* Раздел 2   2       Основной          500 Mб    19 Gб

DISKPART> format fs=ntfs

Завершено (в процентах): 100
Программа DiskPart успешно отформатировала том.
DISKPART> assign letter=E
DiskPart: назначение имени диска или точки подключения выполнено успешно.
DISKPART> list vol

Том    ###    Имя    Метка    ФС    Тип                Размер    Состояние    Сведения
-----
Том 0   0       D      GSP1RMCPRU0  UDF    DUD-ROM            2213 Mб    Исправен
Том 1   1       C      NTFS         NTFS    Раздел             19 Gб    Исправен    Системны
* Том 2   2       E      NTFS         NTFS    Раздел             500 Mб    Исправен
```

## ЗАДАНИЕ 6. Сделаем раздел с загрузчиком неактивным

Выберете диск 0, если он еще не выбран командой **select disk 0**

Командой **list part** посмотрите список разделов, выберете раздел размером 19GB (тот, на котором располагается загрузчик) и посмотрите свойство командой **detail part**.

```
DISKPART> list part

Раздел    ###    Тип                Размер    Смещение
-----
* Раздел 1   1       Основной          19 Gб    1024 Kб
  Раздел 2   2       Основной          500 Mб    19 Gб

DISKPART> select part 1
Выбран раздел 1.
DISKPART> detail part

Раздел 1
Тип : 07
Скрытый: Нет
Активный: Да
Смещение в байтах: 1048576

Том    ###    Имя    Метка    ФС    Тип                Размер    Состояние    Сведения
-----
* Том 1   1       C      NTFS         NTFS    Раздел             19 Gб    Исправен    Системны
```

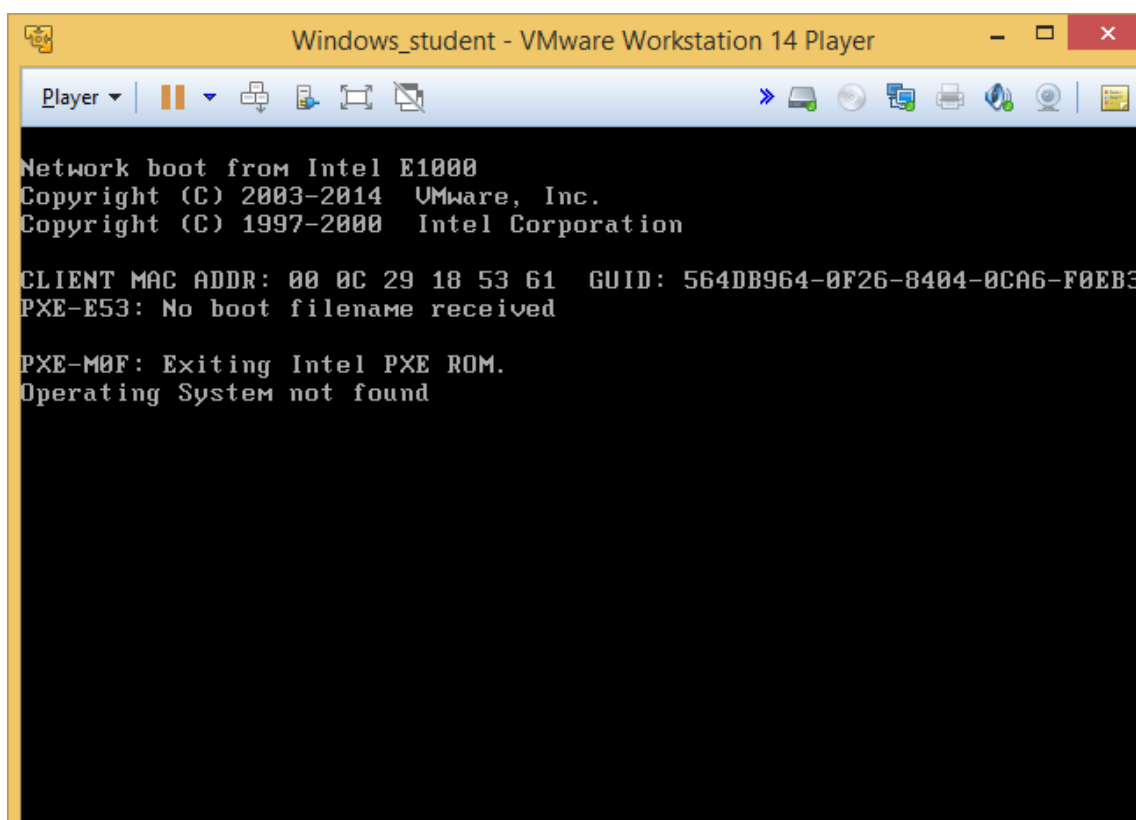
Командой **inactive** сделайте раздел неактивным и проверьте, снялась ли активность.

```
DISKPART> inactive
Текущий раздел помечен как неактивный с помощью программы DiskPart.
DISKPART> detail part
Раздел 1
Тип : 07
Скрытый: Нет
Активный: Нет
Смещение в байтах: 1048576
```

Том	###	Имя	Метка	ФС	Тип	Размер	Состояние	Сведения
* Том 1		C		NTFS	Раздел	19 Gб	Исправен	Системны

Перезагрузите виртуалку, как стандартный Windows

Вы увидите, что в таблице разделов нет активного раздела, где должен располагаться загрузчик и ОС не запускается.

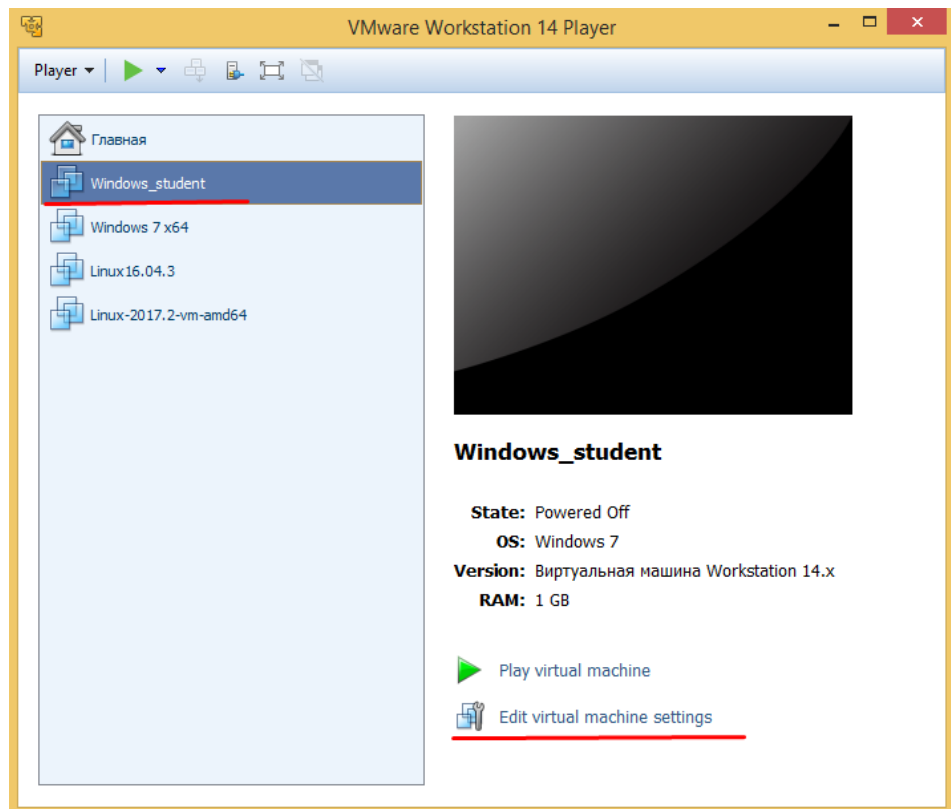


Выключаем нашу виртуальную машину.

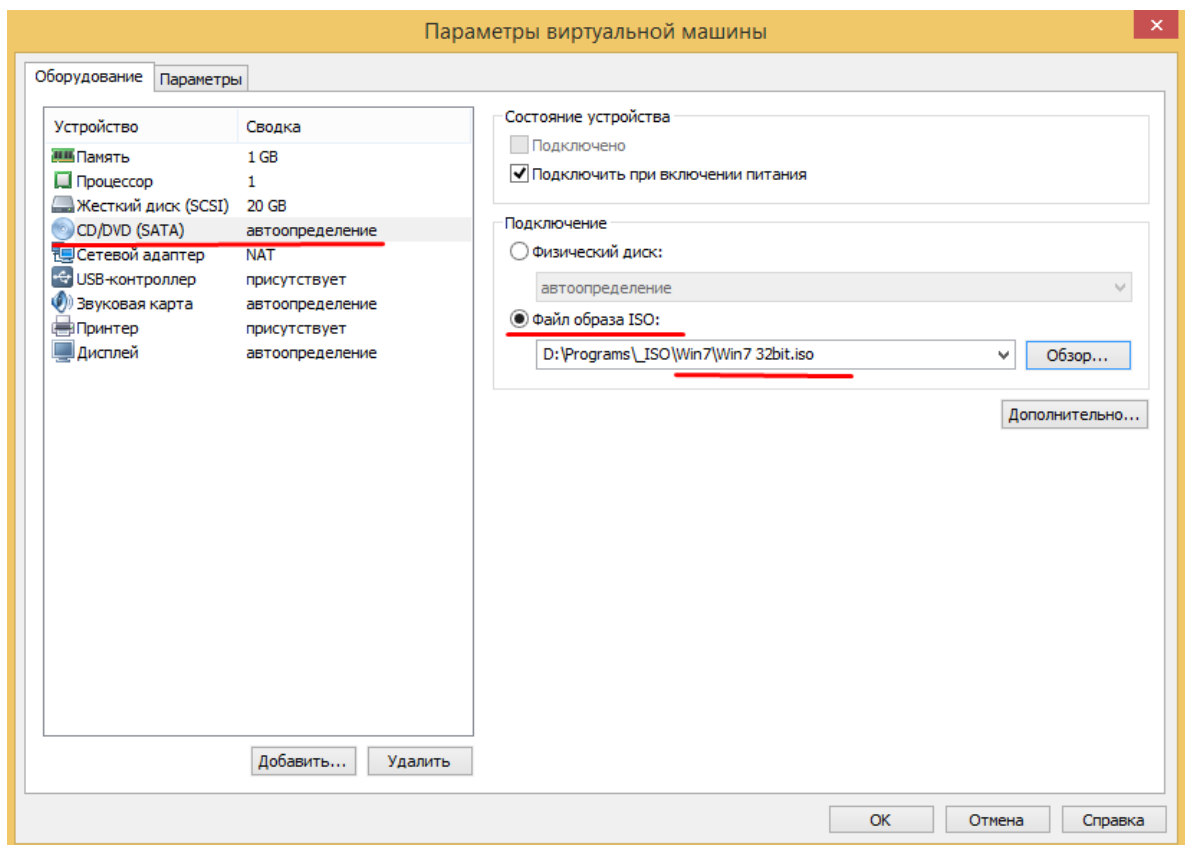
## ЗАДАНИЕ 6. Сделаем раздел с загрузчиком активным

Для того, чтобы снова сделать загрузчик активным понадобится диск с установочником Windows. В случае со стационарным компьютером или ноутбуком это может быть CD диск, флешка. В нашем случае, т.к. мы используем виртуальную машину, то будем использовать iso файл ОС.

Выбираем нашу виртуальную машину и нажимаем *Edit virtual machine settings*



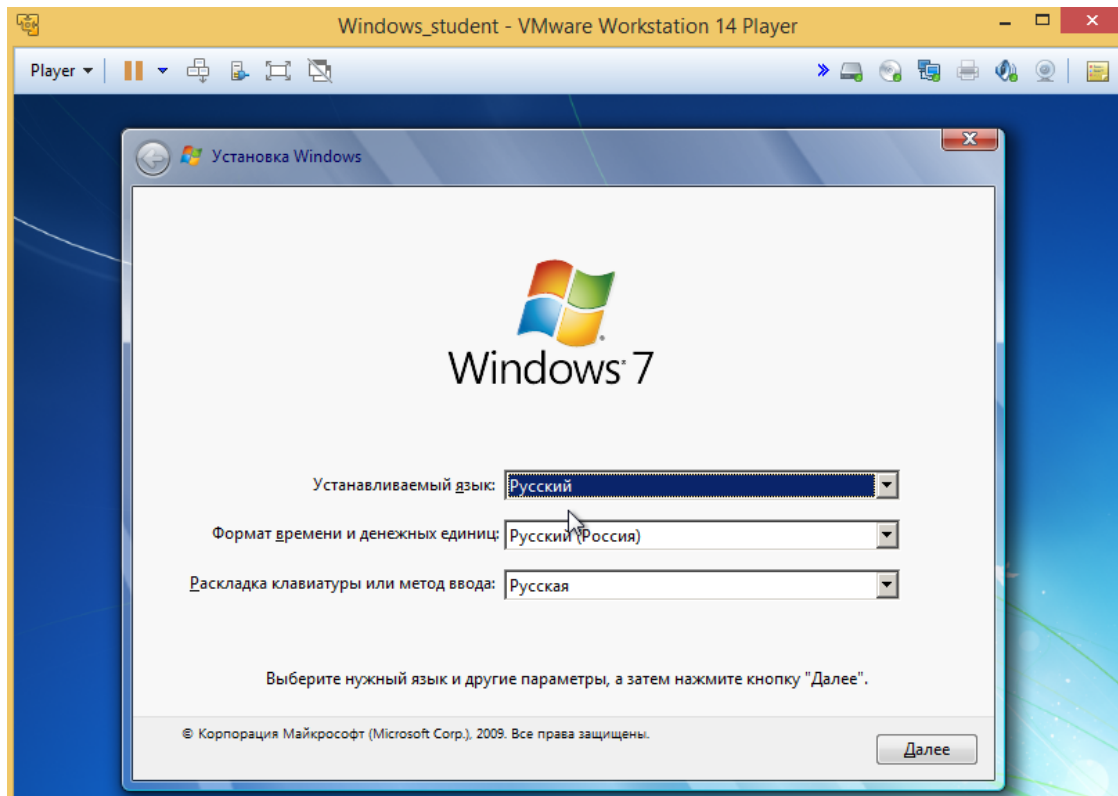
Настройки виртуальной машины позволяют гибко настраивать различные параметры: количество ОЗУ для виртуалки, сетевые адаптеры, подключение различных устройств, выделение ядер процессора, измерение размера жесткого диска и др. Выбираем CD/DVD/SATA и файл образа Win7 32bit.iso



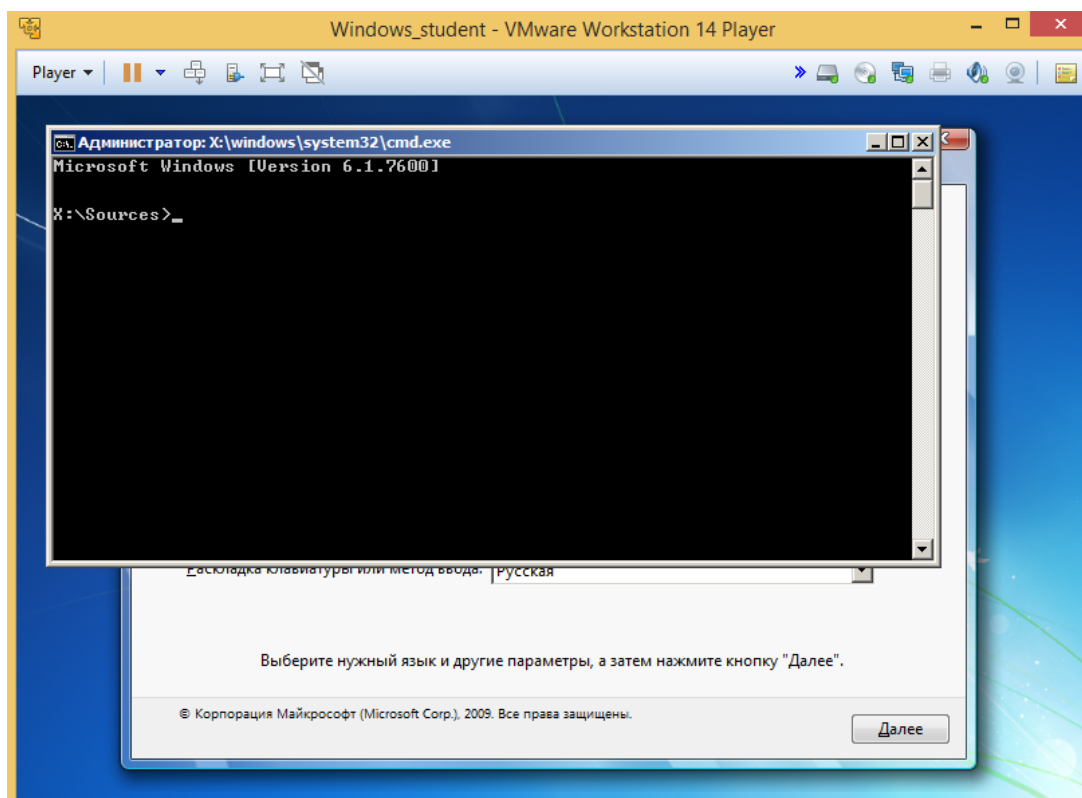
Файл iso лежит на диске D:\virtuals\_windows7 вашей хостовой машины.

Нажимаем ОК и запускаем нашу виртуальную машину.

Начнется подготовка к процессу установки ОС Windows. Нам необходима командная строка Windows. На этапе начала установки нажмите Shift+F10.

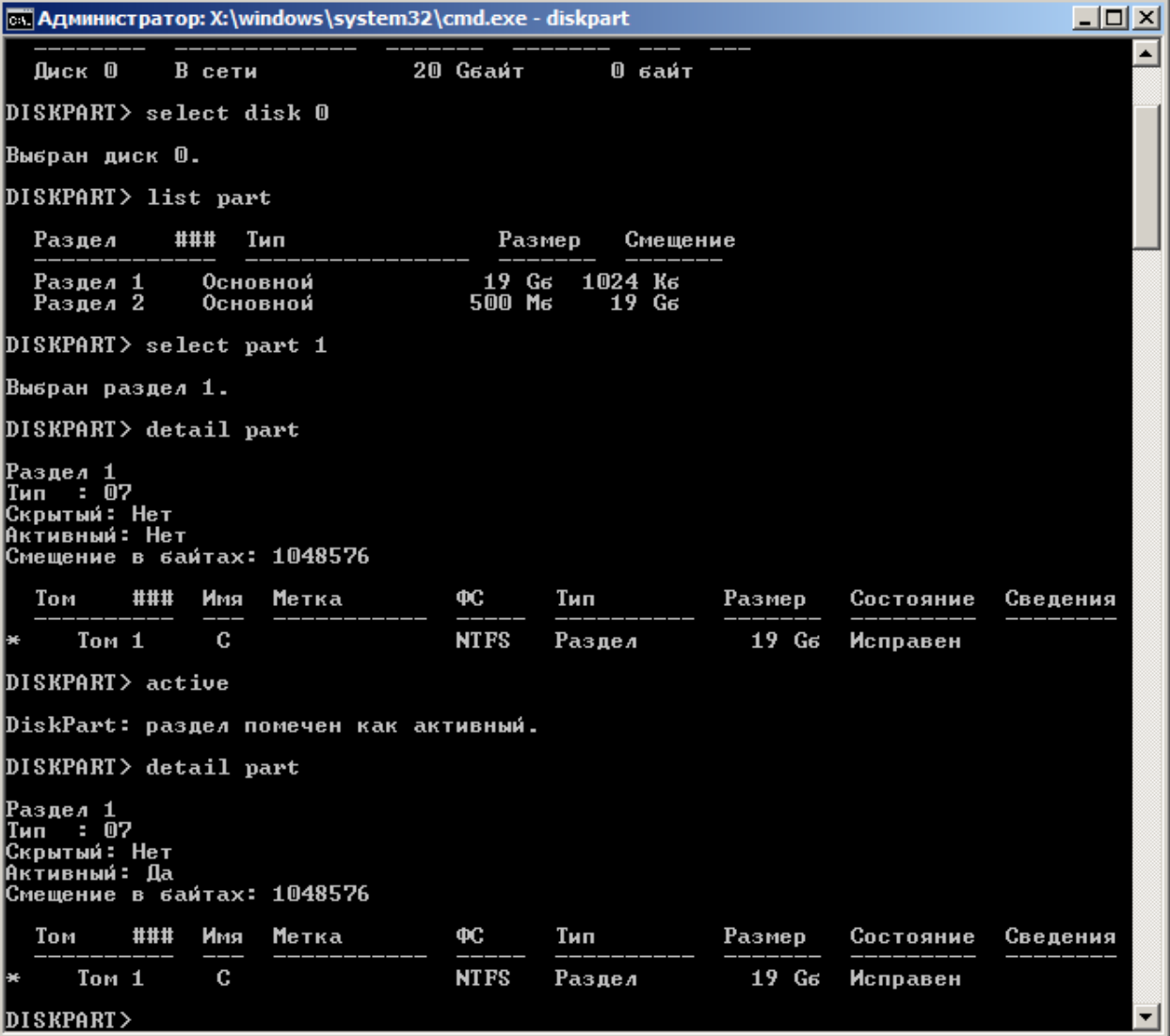


У вас появится командная строка



Запускаем программу **diskpart**

Просматриваем список доступных дисков, выбираем диск 0, выбираем на этом диске раздел с загрузчиком (19GB), просматриваем, что он не активный, делаем его активным и смотрим, что он стал активным. Все команды для работы были рассмотрены ранее.



```
Администратор: X:\windows\system32\cmd.exe - diskpart
Диск 0      В сети      20 Гбайт      0 байт
DISKPART> select disk 0
Выбран диск 0.
DISKPART> list part
Раздел      ###      Тип      Размер      Смещение
-----
Раздел 1      Основной      19 Гб      1024 Кб
Раздел 2      Основной      500 Мб      19 Гб
DISKPART> select part 1
Выбран раздел 1.
DISKPART> detail part
Раздел 1
Тип : 07
Скрытый: Нет
Активный: Нет
Смещение в байтах: 1048576
Том      ###      Имя      Метка      ФС      Тип      Размер      Состояние      Сведения
-----
*      Том 1      C      NTFS      Раздел      19 Гб      Исправен
DISKPART> active
DiskPart: раздел помечен как активный.
DISKPART> detail part
Раздел 1
Тип : 07
Скрытый: Нет
Активный: Да
Смещение в байтах: 1048576
Том      ###      Имя      Метка      ФС      Тип      Размер      Состояние      Сведения
-----
*      Том 1      C      NTFS      Раздел      19 Гб      Исправен
DISKPART>
```

Выходим из командой строки (можно просто закрыть) и выходим из установочника, можно тоже нажать «заккрыть» и на предложение, что установка будет прервана соглашаемся.



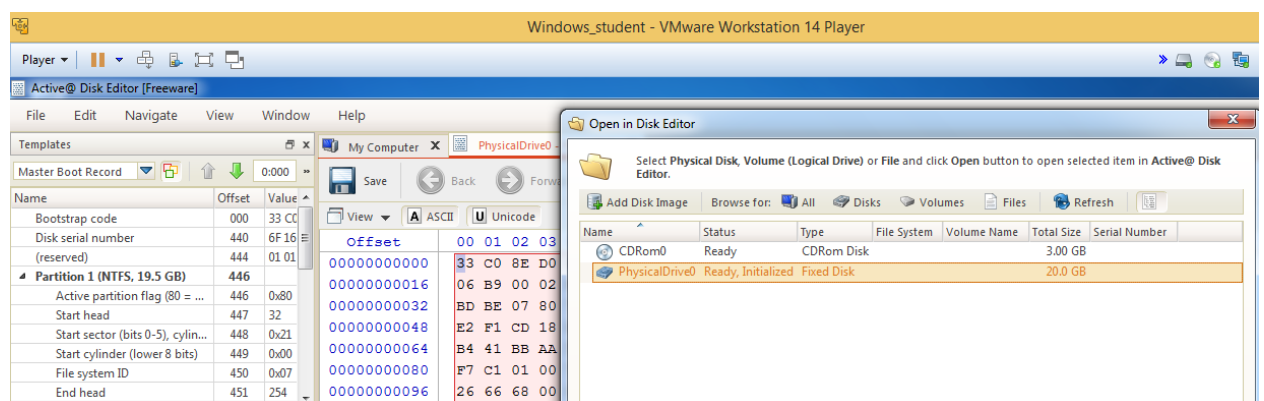
Перезагружаем виртуалку и убеждаемся, что все запускается.

## ЗАДАНИЕ 7. Сломаем загрузчик

Запускаем нашу виртуальную машину

Запускаем программу Active@ Disk Editor от имени администратора

Выбираем открыть диск и выбираем физический диск 20GB



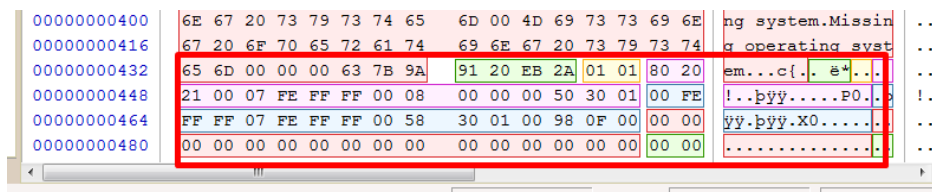
Active@ Disk Editor - это программа, которая позволяет редактировать сектора жестких дисков и логических разделов, а также HEX-код любых типов файлов. Программа обладает настраиваемым интерфейсом и может работать из командной строки. Active@ Disk Editor также может открывать и работать с сжатыми и RAW-образами дисков.

Утилита распространяется совершенно бесплатно и отличается от множества похожих инструментов удобным интерфейсом и большим количеством дополнительных возможностей. При запуске приложения пользователю будет предложено выбрать интересующее его устройство, раздел или логический диск. Active@Disk Editor также позволяет открывать образы дисков в распространенных форматах DIM, Microsoft VHD и VMWare VMDK. Содержимое выбранного носителя будет отображено в окне HEX-редактора. Для просмотра загрузочного сектора, элементов каталога, главной файловой таблицы (MFT) и других элементов файловых систем NTFS, FAT, FAT32, exFAT, HFS, ext2 и ext3 пользователю достаточно выбрать соответствующий шаблон из списка. К примеру, после выбора пункта «Master Boot Record» приложение мгновенно перейдет к соответствующей области диск. В окне «Templates» вы найдете полезную дополнительную информацию о выбранном элементе, а для интерпретации предлагаемых значений можно воспользоваться инструментом Data Inspector.

Если вас интересуют конкретные данные (например, содержимое потерянного файла), однако вы не знаете точно, где их нужно искать, на помощь придет функция «Find». Этот инструмент позволит выполнять поиск заданной последовательности символов ANSI, Unicode или hex-значений и может похвастаться поддержкой регулярных выражений и групповых символов.

Разумеется, приложение позволяет не только изучать содержимое дисков, но и вносить необходимые изменения. Функция «Navigate» обеспечит быстрое обнаружение нужного сектора и пользователь сможет редактировать байты данных, копировать и вставлять информацию, очищать или заполнять выбранные блоки. Стоит отметить, что использовать доступные средства редактирования необходимо с предельной осторожностью. Любое неосторожное действие может привести к катастрофическим последствиям, вплоть до безвозвратной утери важных данных. В случае если вы допустили ошибку, воспользуйтесь опциями «Undo» и «Revert Changes» для отмены операции.





My Computer X PhysicalDrive0 - Fixed Disk X	
Save Back Forward Edit Find Navigate	
View A ASCII U Unicode	
Offset	00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15
00000000000	33 C0 8E D0 BC 00 7C 8E C0 8E D8 BE 00 7C BF 00
00000000016	06 B9 00 0E EE F3 A4 50 68 1C 06 CB FB B9 04 00
00000000032	BD FE 07 80 EE EE 00 7C 0B 5F 84 35 01 83 C5 10
00000000048	E2 F1 CD F8 88 56 00 55 C6 46 11 05 C6 46 10 00
00000000064	B4 41 FB AA 55 CD 13 5D 72 0F 81 FB 56 7A 75 09
00000000080	F7 C1 01 00 74 03 FE 46 10 66 60 43 7E 10 00 74
00000000096	26 66 68 00 67 00 00 66 FF 67 08 68 00 0D 68 00
00000000112	76 78 34 00 68 10 00 B4 42 8A AD 00 67 F4 DD 13
00000000128	9F 83 C4 10 9E EB 14 B8 01 02 BB 00 7C 8A 56 00
00000000144	8A 76 03 4A 4E 67 8A 6E 03 CA 13 66 61 73 1C FE
00000000160	4E 11 56 0C 80 7E 00 80 0F 84 8A 00 B6 70 EB 84
00000000176	55 32 E4 8A 56 00 CD 13 5D E5 3E 81 3E FE 7D 55
00000000192	AA 75 6E FF 76 00 E8 8D 00 75 17 F4 B0 D1 E6 64
00000000208	E8 83 06 70 DF E6 60 E8 7C 0F B0 FF E6 D4 E8 75
00000000224	00 FB B8 0D BB CD 1A 66 23 C0 7F 3B D6 81 FB 54
00000000240	43 50 F1 75 F2 81 D9 02 01 72 2C 6D 68 07 BB 00
00000000256	00 66 68 00 02 00 00 66 68 08 0D FD 00 66 53 66
00000000272	53 66 55 66 68 00 00 00 00 66 68 00 7C 00 00 66

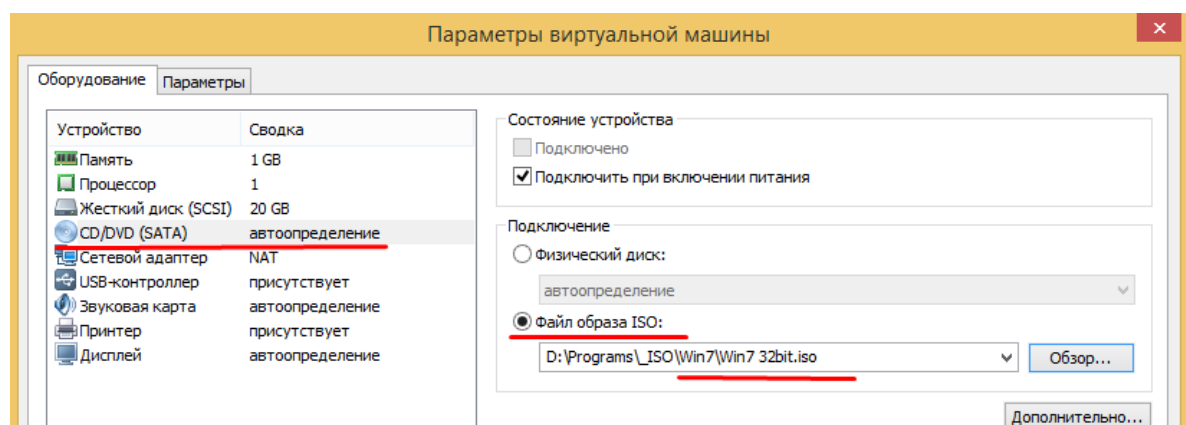
После изменения нажмите кнопку Save. Согласитесь с появившемся предупреждением.

Перезагрузите виртуальную машину.

В зависимости от того, насколько сильно вы поменяли загрузчик, могут происходить различные ситуации: система сразу начнет грузиться с iso образа (если он не был убран после заданий выше), система не будет грузиться вообще, система будет грузиться с ошибками.

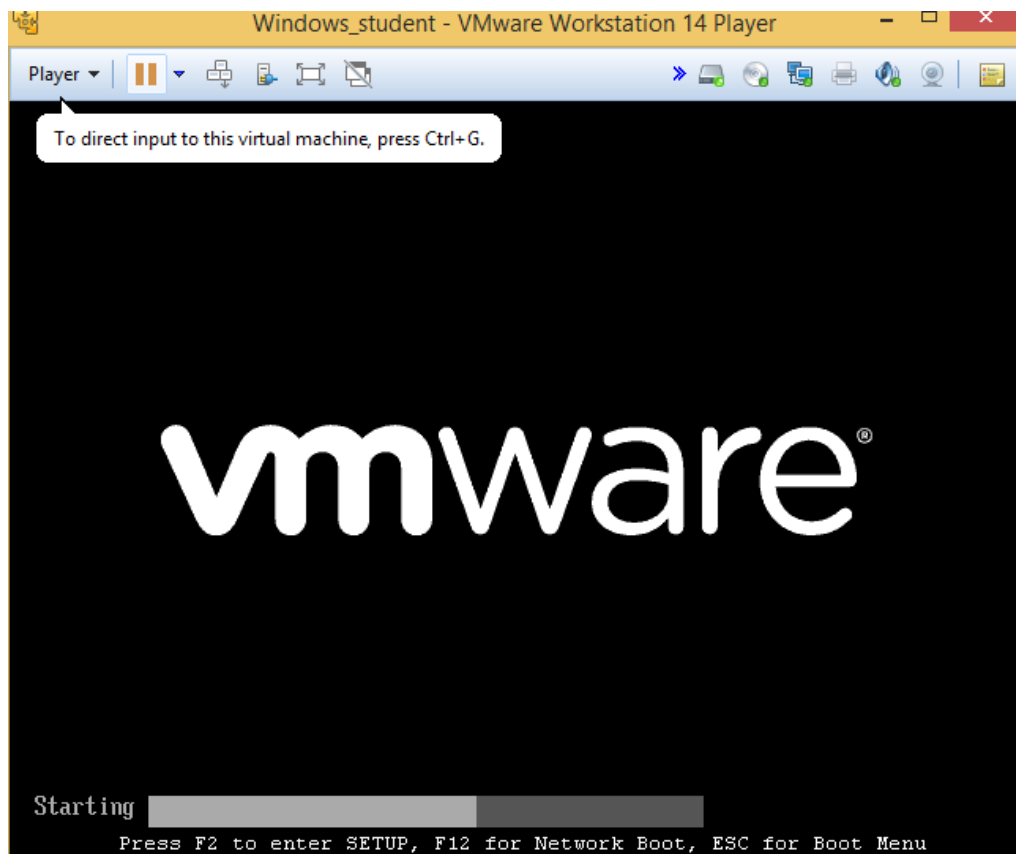
## ЗАДАНИЕ 8. Восстановим загрузчик

Если ничего не происходит, хотя в настройках виртуальной машины стоит iso образ, необходимо явно указать, откуда грузиться



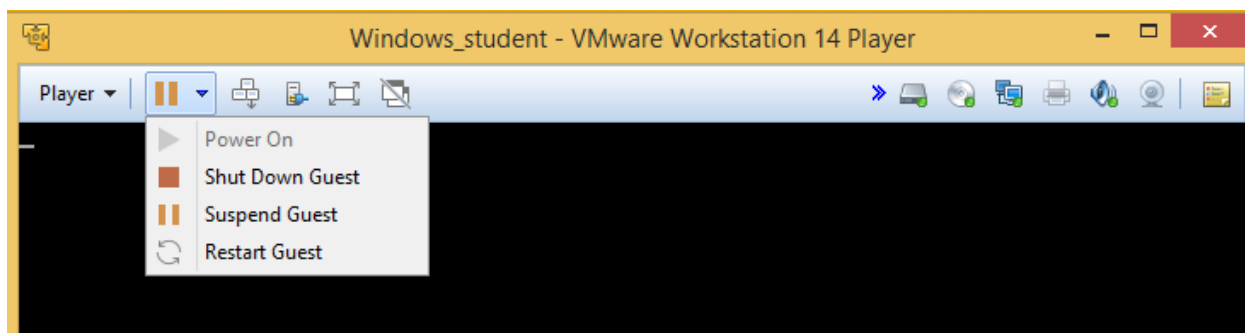
Для этого нужно сделать следующее:

При старте виртуальной машины необходимо отметить, какая клавиша отвечает за вход в BIOS



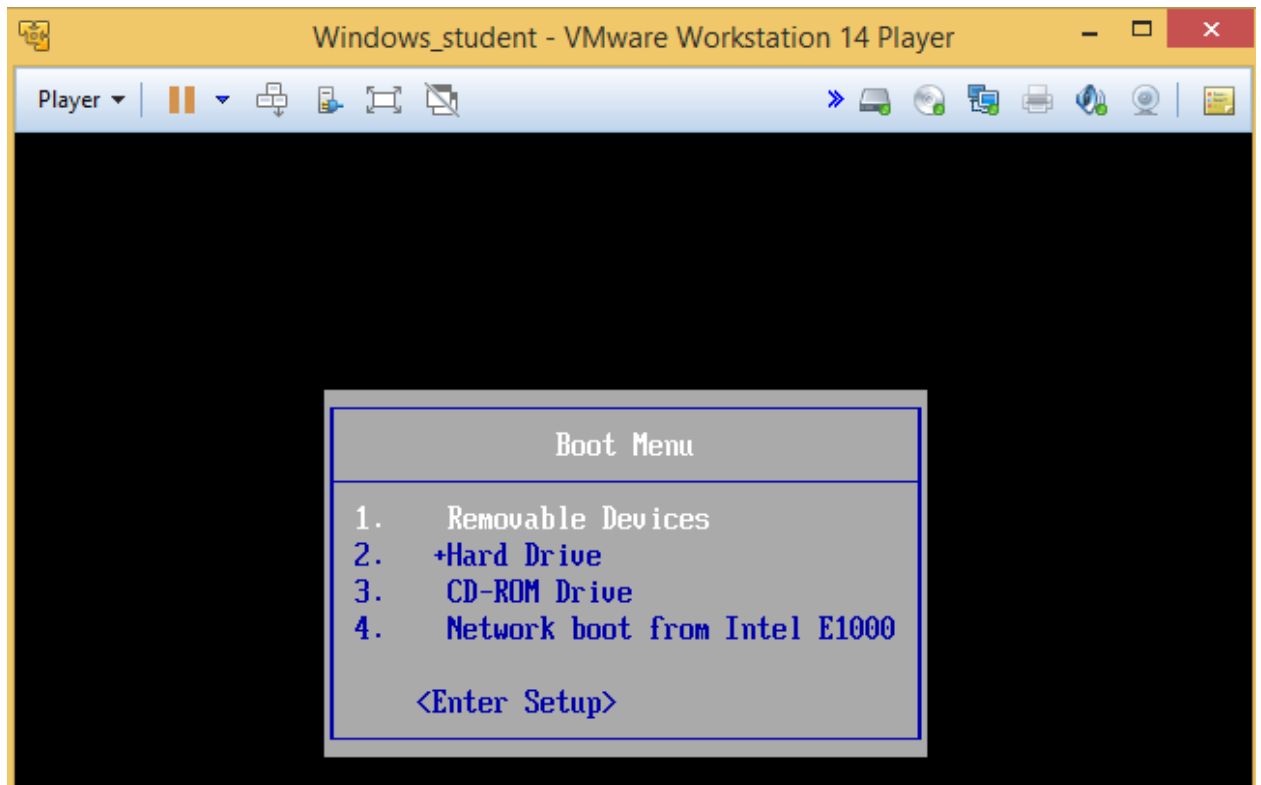
В данном случае F2 – вход в BIOS, F12 для загрузки по сети и ESC для выбора с какого устройства грузиться.

Перезагружаться в случае, если ОС не запускается, можно кнопками в статус баре.



Для того, чтобы произвести выбор, необходимо при старте виртуальной машины щелкнуть по черному фону (передать фокус в виртуалку) и нажать соответствующую клавишу.

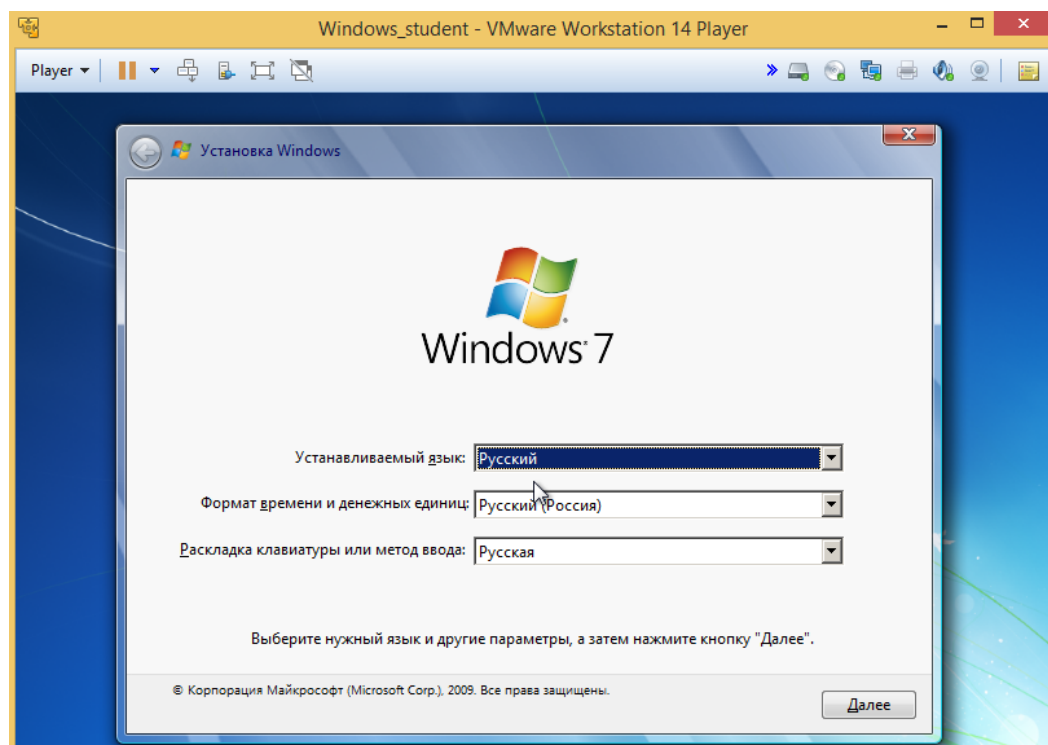
При нажатии Esc будет предложено выбрать устройство загрузки



Выбирайте CD-ROM Drive и нажмите Enter.

После этого будет предложено нажать любую клавишу, чтобы начать процесс установки системы.

Дождись появления выбора языка установки



Нажмите Shift + 10

Необходимо определить диск, на котором следует восстановить загрузчик. В нашем случае диск один. Если дисков было бы несколько можно было перемещаться по ним, используя стандартные команды командной строки Winndow для того, чтобы определиться, где находился наш загрузчик.

Запускаем программу **diskpart**.

Командой **list vol** смотрим доступные тома

```
DISKPART> list vol
```

Том	###	Имя	Метка	ФС	Тип	Размер	Состояние	Сведения
Том 0		E	17.06	UDF	DUD-ROM	3070 Мб	Исправен	
Том 1		C		NTFS	Раздел	19 Гб	Исправен	
Том 2		D		NTFS	Раздел	500 Мб	Исправен	

```
DISKPART> exit
```

Завершение работы DiskPart...

X:\Sources>

Видим, какие тома у нас есть, наш том с загрузчиком это том 1 – диск C.

Выходим из diskpart командой **exit**.

Воспользуемся утилитой bootsect.exe

Утилита BOOTSECT.EXE позволяет изменить программный код загрузчика Windows для переключения между двумя вариантами диспетчера загрузки - BOOTMGR или NTLDR . Загрузчик ntldr использовался до появления операционной системы Windows Vista. В процессе начальной загрузки, программный код загрузочного сектора раздела (PBR - Partition Boot Sector) обеспечивал поиск, считывание в память и передачу управления файлу ntldr , который размещался в корневом разделе загрузочного диска. Конфигурирование загрузчика ntldr выполнялось с помощью простого текстового файла boot.ini, содержимое которого задавало список загружаемых операционных систем, их параметры загрузки, размещение системных файлов и т.п. В операционных системах Windows Vista /Server 2008 и более поздних, загрузчик ntldr не используется, и заменен диспетчером загрузки bootmgr. Соответственно, изменился и программный код загрузочного сектора раздела, обеспечивающий передачу управления файлу bootmgr. Новый диспетчер загрузки использует собственные данные конфигурации загрузки (Boot Configuration Data - BCD) и может выполнять, при определенных настройках,

загрузку любых операционных систем семейства Windows . Загрузчик ntldr не поддерживает возможность загрузки Windows Vista и старше. Для конфигурирования диспетчера загрузки bootmgr используется команда BCDEDIT

Команда BOOTSECT позволяет записывать заданный программный код загрузочных секторов, обеспечивающих загрузку либо ntldr, либо bootmgr .

Формат командной строки:

**bootsect {/help|/nt60|/nt52} {SYS|ALL|< DriveLetter >} [/force] [/mbr]**

Параметры командной строки Bootsect:

**/help** - отображение справочной информации;

**/nt52** - запись программного кода загрузочного сектора, обеспечивающего использование загрузчика ntldr для операционных систем, предшествующих Windows Vista.

**/nt60** - запись программного кода в загрузочные секторы для обеспечения загрузки файла bootmgr - диспетчера загрузки Windows Vista/Server 2008 и более поздних ОС семейства Windows.

**SYS** - запись будет выполнена в секторы системного раздела, который использовался для загрузки Windows, в среде которой выполняется данная команда.

**ALL** - запись программного кода будет выполнена для всех существующих разделов, которые могут быть использованы для загрузки Windows.

**DriveLetter** - буква диска, для которого будет выполнена перезапись программного кода загрузочных секторов.

**/force** - принудительное отключение используемых другими программами томов дисков для обеспечения монопольного доступа утилиты bootsect.exe

**/mbr** - изменение программного кода главной загрузочной записи (MBR - Master Boot Record) без изменения таблицы разделов диска. При использовании с параметром /nt52, MBR будет совместима с предшествующими Windows Vista версиями. При использовании с

параметром /nt60 - MBR будет совместима с операционными системами Windows Vista и более поздними.

### Примеры:

**bootsect /nt52 E:** - создать для диска E: загрузочные записи для операционных систем Windows XP/2000/NT, т.е для загрузки на базе ntldr;

**bootsect /nt60 /mbr C:** - изменить загрузочные сектора диска C: для обеспечения загрузки диспетчера bootmgr

**bootsect /nt60 SYS** - изменение загрузочных секторов для раздела, с которого выполнена загрузка текущей ОС Windows.

Воспользуемся примерами и используем команду bootsec для восстановления загрузчика

Пропишем **bootsect /nt60 /mbr C:** в командной строке и нажмем Enter

Должно появиться сообщение, что bootcode успешно обновлен.

```
X:\Sources>bootsect /nt60 c: /mbr
Target volumes will be updated with BOOTMGR compatible bootcode.
C: (\??\Volume{a5663748-562c-11e9-9f46-806e6f6e6963})
    Successfully updated NTFS filesystem bootcode.
\??\PhysicalDrive0
    Successfully updated disk bootcode.
Bootcode was successfully updated on all targeted volumes.
```

Перезагружаемся (можете или убрать галочку с запуска с \_iso файла, либо при загрузке не нажимать не какую из клавиш, чтобы снова не пошла установка Windows)

Теперь все должно загрузиться

## Основы работы с командной строкой Windows

Для автоматизации процесса обработки вводимых команд возможно создавать простейшие скрипты на языке оболочки командной строки.

Выше вы уже познакомились с интерпретатором командной строки cmd.exe. Давайте теперь попробуем автоматизировать процесс создания дерева папок. Создадим папку firstcat, в которой будет папка secondcat, которая в свою очередь будет содержать папку thirdcat.

**Примечание:** для тестовых случаев используем простейшие команды, изученные выше. На самом деле, язык cmd.exe относительно сложный и многообразный, в то же время он позволяет автоматизировать большое количество рутинных задач (проверка доступности компьютеров в сети, запись в файлы времени их простоя, создание каталогов с данными, передача сообщений и др.) В тоже время, использование такого подхода позволяет не использовать сложные языки программирования для автоматизации несложных задач.

**Создайте вручную папку, где будет производить все дальнейшие действия**

Для этого создадим для начала простой текстовый файл, в котором пропишем следующие строки:

```
echo off
rem create new catalog
md firstcat
rem go to into new catalog
cd firstcat
md secondcat
cd secondcat
md thirdcat
rem this need for pause of programm
pause
```

Рассмотрим более подробно описанное сверху



*echo off* – команда, которая обычно указывается в начале скриптов, для такого, чтобы при запуске скрипта не выводились все последовательности шагов и не загромождали место на экране

*rem* – обозначение комментария, это означает, все что идет далее в строке не будет учитываться в программе. Такие строки необходимы, чтобы не запутаться в написанном коде. Также это является хорошим тоном при написании программ. В данном случае, комментарии необходимо писать латинскими буквами

*md firstcat* – создание первого каталога

*cd firstcat* – переходим внутрь первого каталога

Далее аналогично создаются второй и третий каталоги

*pause* – команда, которая приостанавливает выполнение программы

Мы создали текстовый файл с набором строк. Как теперь сделать, чтобы это превратилось в запускаемый скрипт? Очень просто. Достаточно поменять расширение файла с .txt на .bat. Данный файл стал исполняемым, т.е. если вы попытаетесь его открыть, начнет выполняться программный код. Для изменения файла нажмите на него правой кнопкой мыши и выберите изменить. Все. Теперь запустите скрипт, и вы увидите, что создано 3 папки – одна внутри другой.

### **Задания.**

1. Попробуйте запустить скрипт несколько раз, создаются ли папки снова?
2. Закомментируйте или удалите строку *echo off*, запустите программу, посмотрите, как изменился вывод в консоль (верните после этого как было)
3. Попробуйте изменить один из комментариев на кириллические символы. Что получилось? (верните после этого как было)
4. Уберите команду *pause*. Как изменился запуск программы?

Используя данный скрипт можно создавать необходимые папки в любом месте (там, где находится ваш исполняемый файл .bat)

Теперь рассмотрим удаление созданных папок.

Выше вы рассмотрели удаление папки с помощью команды *rd*.

Попробуйте создать новую программу (.bat файл лежит там же, где и созданные каталоги) и прописать в ней строки

```
echo off
```

```
rd firstcat
```

Программа выдаст вам ошибку о том, что внутри папок находятся дополнительные папки.

Для того, чтобы удалить все папки, вам необходимо двигаться в обратном порядке. Вначале необходимо зайти в первый каталог, из него перейти во второй каталог, после чего удалить третий каталог. Далее вернуться на уровень выше в первый каталог и удалить второй, после чего выйти из первого и удалить первый.

### **Задание.**

Напишите программу в новом файле .bat, которая удаляет папки по описанном выше алгоритму.

Подсказка, первые строки должны выглядеть примерно так:

```
echo off
```

```
cd firstcat
```

```
cd secondcat
```

```
rd thirdcat
```

Теперь рассмотрим пример, автоматического создания файлов в наших папках. Примечание: для создания пустого файла используется команда `echo > "название файла"`. Создайте новый файл .bat в котором будет следующая программа

```
echo off
```

```
rem go to into firstcat
```

```
cd firstcat
```

```
rem create firstfile
```

```
echo > firstfile.txt
```

```
rem go to into secondcat
```

```
cd secondcat
```

```
rem create secondfile
echo > secondfile.txt
rem go to into secondcat
cd thirdcat
rem create thirdfile
echo > thirdfile.txt
pause
```

Запустите вначале скрипт создания папок, после чего запустите скрипт создания файлов.

Проверьте, в каждой папке теперь должно оказаться еще и по файлу.

Попробуйте теперь запустить скрипт удаления папок. Программа выдаст ошибку, потому что наши папки не пусты. Необходимо удалить файлы. Это можно сделать по аналогии с файлами, но, как мы рассмотрели выше, скриптовый язык призван облегчить жизнь, а не усложнять ее, потому воспользуемся всего одной командой – рекурсивное удаление.

```
echo off
del firstcat /S /Q /F
pause
```

Строка `del firstcat /S /Q /F` производит рекурсивное удаление файлов, находящихся в наших папках

Попробуйте создайте новый .bat файл, добавьте строки выше, сохраните и запустите.

Как видите, папки остались, но их содержимое было удалено.

Теперь вы можете запустить скрипт удаления папок – и ваши папки будут удалены.

Таким образом у вас должно получиться 4 скрипта, которые вы можете запускать в следующей последовательности

1. Скрипт создания папок
2. Скрипт создания файлов в папках
3. Скрипт удаления файлов
4. Скрипт удаления папок

## **Задание**

Создайте скрипт, который строит дерево каталогов, верхнем будет папка с вашей фамилией, в ней папки с именем и отчеством.

Создайте скрипт, который создавал в папке с вашей фамилией файл, название которого это дата вашего рождения (например, 20121999), а в папке с вашим отчеством – файл, имя которого – номер компьютера, за которым вы сидите.

Создайте скрипт, удаляющий файлы

Создайте скрипт, удаляющий папки

Объедините эти скрипты в один так, чтобы после каждого действия (создание папок, создание файлов, удаление файлов, удаление папок) пользователю предлагалось нажать любую клавишу.