

Л. Г. Хорошилов¹, Н. А. Костин¹

¹ФГАОУ ВО «РГУ нефти и газа (НИУ) имени И. М. Губкина», г. Москва, Российская Федерация

Научный руководитель: В. В. Павловский, кандидат технических наук, доцент, кафедра информационной безопасности, Московский технический университет, Москва, Россия

СБОРКА И НАСТРОЙКА ОБРАЗА WINDOWS PE ДЛЯ ОБСЛУЖИВАНИЯ СИСТЕМ

Аннотация. В статье рассматривается проблема создания и настройки образа Windows PE в комплектации ПО для обслуживания систем. Актуальность темы обусловлена необходимостью наличия автономного инструмента, который может использоваться при диагностике, восстановлении и администрировании компьютерных систем без загрузки основной операционной системы. Особое внимание уделяется практическим этапам подготовки рабочей среды, отбору портативных программ, добавлению системных файлов и настройке компонентов, обеспечивающих запуск графического интерфейса. Предложенный метод основан на редактировании официального образа Windows и внедрении в него нужных программ и файлов. В работе описывается использование инструментов Windows ADK и DISM, монтирование и модификация WIM-образа, правка автономного реестра, настройка стартовых скриптов и последующая сборка загрузочного ISO-носителя. Отдельно рассматривается тестирование созданного образа в виртуальной машине, подтверждающее его работоспособность, стабильность графической оболочки и пригодность для выполнения задач аварийного обслуживания. Практическая ценность исследования заключается в возможности гибко изменять состав сервисной среды, исключать лишние компоненты и добавлять только те утилиты, которые необходимы для конкретных сценариев эксплуатации. Такой подход повышает удобство обслуживания систем и делает процесс восстановления более контролируемым для администрирования Windows. Разработанная методика носит прикладной характер и позволяет системным администраторам создавать легкие, автономные и функционально насыщенные сервисные диски, адаптированные под конкретные задачи.

Ключевые слова: Windows PE, сборка образа, предустановленная среда, обслуживание систем, системное администрирование, загрузочный ISO-носитель.

Введение

Актуальность. В современном мире, где критичность информационных инфраструктур и их непрерывный рост сложности, стандартные средства восстановления операционных систем семейства Windows зачастую недостаточны для решения нестандартных задач обслуживания. В связи с этим создание портативного и автономного инструмента на базе Windows PE становится одним из ключевых моментов стратегии обеспечения отказоустойчивости. Разработка методики целенаправленной сборки и настройки образа, оптимизированного под конкретные задачи системного администрирования, представляет собой актуальную практическую проблему.

Объект исследования — процесс сборки и настройки предустановленной среды Windows PE.

Предмет исследования — методики и инструментальные средства программного обеспечения для расширения функциональности среды Windows PE.

Цель исследования — разработать практико-ориентированную методику создания специализированного образа Windows PE.

Для достижения цели были поставлены следующие задачи:

1. Классифицировать и отобрать стороннее ПО для интеграции по заданным критериям.
2. Разработать методику добавления в образ недостающих системных файлов и библиотек.
3. Создать метод настройки среды через правку автономного реестра и стартовых скриптов.
4. Составить алгоритм сборки конечного образа с использованием DISM и сопутствующих утилит.

Литературный обзор

Windows Preinstallation Environment (WinPE) – минимальная операционная система на базе ядра Windows, предназначенная для подготовки компьютера к установке, развертывания

и восстановления полнофункциональной ОС. WinPE предоставляет ограниченный набор служб и запускается в оперативной памяти, что делает ее идеальной платформой для создания автономных сервисных сред [1].

Windows ADK – официальный пакет инструментов от Microsoft, включающий утилиты для создания, настройки и развертывания образов Windows, в том числе средства для работы с WinPE [1].

Существует ряд автоматизированных конструкторов WinPE (например, Win10XPE, WinBuilder), которые позволяют быстро создавать загрузочные образы. Однако они часто включают избыточные компоненты, скрывают от пользователя детали работы с DISM и WIM-структурами, а также могут содержать неуправляемые зависимости. Выбор ручного метода обусловлен необходимостью полного контроля над составом образа, отсутствием лишних компонентов, а также образовательной ценностью — глубоким пониманием работы инструментов развертывания Microsoft (DISM, Windows ADK).

Для обоснования выбора ручной методики были сопоставлены три аналога: базовая сборка WinPE по официальной документации, готовые инструкции и автоматизированные способы создания загрузочного диска, а также ручные пользовательские сборки. Сравнение выполнялось по следующим метрикам: полнота функциональности, размер итогового образа, прозрачность состава, воспроизводимость действий, сложность добавления собственного ПО и зависимость от сторонних сценариев. Базовая WinPE, создаваемая средствами Microsoft, наиболее стабильна и документирована, однако содержит минимум пользовательских инструментов и требует дополнительной настройки. Готовые способы создания загрузочного диска удобны для быстрого получения результата, но в меньшей степени позволяют контролировать набор компонентов и структуру WIM-образа [2]. Ручные практики сборки дают более высокую гибкость и позволяют точно добавлять графическую оболочку, файловые менеджеры и сервисные утилиты, однако требуют внимательной проверки зависимостей и корректности запуска.

При этом официальная процедура создания загрузочного носителя определяет базовый безопасный алгоритм подготовки USB/ISO-носителя. Общие сведения о назначении и ограничениях WinPE использовались для определения границ применения среды, а материалы по базовым загрузочным дискам — для уточнения типового минимального набора операций при построении сервисной среды. Дополнительно учитывались практические вопросы работы с ISO-образами в Windows и необходимость взаимодействия создаваемой среды с периферийными устройствами и накопителями при обслуживании компьютеров.

На основании литературного анализа можно сформулировать следующие гипотезы исследования:

1. H1: Комбинация официальных инструментов Microsoft (ADK/DISM) с систематизированными практическими методами сообщества позволяет создать стабильный и функционально насыщенный образ WinPE, превосходящий стандартную сборку по возможностям обслуживания систем;

2. H2: Разработанная методика ручной сборки образа Windows PE обеспечивает полную прозрачность и контролируемость каждого этапа создания среды, что позволяет гарантировать целостность, минималистичность и функциональную определённость итогового образа.

Методы исследования

Анализ документации: изучалась официальная техническая документация Microsoft (Windows ADK, DISM, WinPE) для выявления формальных возможностей, ограничений и санкционированных методов работы со средами предустановки [1]. Параллельно анализировались неофициальные источники (форумы, технические блоги, репозитории сборок) для выявления эмпирических практик, типовых проблем ручной сборки и способов создания загрузочных сервисных дисков [3].

Системный анализ и моделирование: исследуемый объект (процесс сборки WinPE) рассматривался как система взаимосвязанных этапов. Была разработана модель технологического конвейера, включающая последовательность операций: подготовка

Рубрика 1. Информатика и информационные процессы

исходников → добавление компонентов → модификация образа → тестирование → финальная сборка. Такой порядок соотносится с типовой логикой создания базового загрузочного диска Windows PE и последующего расширения его функциональности [4].

Ход исследования

Для начала работы необходимо скачать и установить комплект средств для развертывания и оценки Windows ADK. Для исследования достаточно выбрать только средства развертывания [1].

Далее для удобства работы создадим дерево каталогов. Создадим его через командную строку для упрощения действий.

Каждая папка будет впоследствии содержать в себе определенный набор файлов для создания образа.

1. В директории AddFiles будут находиться файлы программ, нужных для данного образа;
2. В директории originalWIM мы поместим оригинальный файл boot.wim полученный из скачанных дистрибутивов Windows;
3. В директории COMPLETE будет храниться готовый, полностью собранный WIM-образ;
4. Директория Mount — временная точка монтирования WIM-образа;
5. Директория util является хранилищем вспомогательных утилит;
6. Директория ISO будет содержать в себе отредактированный ISO образ, который в дальнейшем используется как загрузочный носитель или подключается к виртуальной машине для проверки [5].

В папку AddFiles добавим программы, которые необходимы данной сборки. В моем случае это будут 7-Zip, Explorer++, Notepad++. Скачиваем софт из доступных источников, устанавливаем на свой ПК, после чего копируем и переносим каталог с программой в папку Program Files. Помимо программ мы установим программу WinXShell для отображения работы образа в графическом интерфейсе. В случае с Explorer++, программу удобно вызывать из командной строки, поэтому добавим её в директорию Windows\System32.

Оригинальное средство Microsoft для скачивания образа Windows — MediaCreationTool, не умеет скачивать старые версии Windows, и качает лишь последнюю поддерживаемую Microsoft, но нам нужна самая первая версия Windows 10 1507, которую без труда можно скачать с GitHub. Качаем с помощью него Windows 10 1507 x32 — нам нужен ISO файл. Версия выбрана с целью не занимать большие объёмы пространства, но можно выбрать любую другую из доступных в MediaCreationTool. На базе этой версии мы будем строить образ, но для полноценной работы образа нам необходимо в следующих шагах иметь ряд файлов, отсутствующих в версии Windows 10 1507. Для этого скачиваем любую версию Windows 10 на x32, в моем случае это будет версия 21H2. Так же для работы с ISO файлами скачаем редактор iso файлов. Мой выбор пал на PowerISO, так как на практике для таких операций требуется корректное монтирование и изменение ISO-образа [5].

Использование системных файлов из Windows 10 21H2 в образе на базе версии 1507 обусловлено необходимостью обеспечения совместимости с современным программным обеспечением и драйверами, которые могут отсутствовать в старой версии. Такой подход позволяет сохранить малый размер образа за счет базовой версии 1507, одновременно расширив его функциональность за счет отдельных библиотек из более новой сборки. При выборе компонентов учитывалось, что WinPE является ограниченной средой и не заменяет полноценную установленную ОС, поэтому в образ включались только те файлы, которые нужны для выбранных сценариев обслуживания [6]. Чтобы снизить риск «DLL Hell», файлы не заменялись массово: в образ добавлялся ограниченный набор компонентов, необходимых для запуска графической оболочки и утилит. К импортируемым компонентам относились DLL/MUI-файлы, связанные с отображением проводника и оболочки Windows, обработкой окон и графики, а также с базовыми системными интерфейсами: ExplorerFrame.dll и ExplorerFrame.dll.mui, shellstyle.dll и shellstyle.dll.mui, oledlg.dll, ole32.dll.mui, shfolder.dll, thumbcache.dll, twinapi.dll, twinapi.appcore.dll, twinui.dll, twinui.appcore.dll, d3d11.dll, dcomp.dll,

ddraw.dll, dxgi.dll, DXCore.dll, opengl32.dll, glu32.dll, pdh.dll и pdhui.dll. Отдельно добавлялись локализованные MUI-файлы для корректного отображения элементов интерфейса.

Проверка совместимости выполнялась практическим способом: после добавления файлов образ монтировался и запускался в виртуальной машине, где проверялись загрузка WinPE, запуск WinXShell, открытие Explorer++, Notepad++, командной строки и меню графической оболочки. Дополнительно контролировалось отсутствие критических ошибок при старте среды и при обращении к файловой системе. Оставшиеся риски связаны с тем, что часть библиотек взята из другой версии Windows: возможны несовпадения зависимостей СОМ-компонентов, отсутствие отдельных ресурсов локализации или некорректная работа программ, требующих полноценной установленной ОС. В рамках данной сборки эти риски считаются приемлемыми, поскольку образ используется как сервисная среда для диагностики и восстановления, а не как универсальная замена полноценной Windows.

Для корректной работы Windows PE необходимо извлечь из скачанных образов обязательные для работы образа файлы. Для этого через редактор iso файлов открываем архив образа. В архиве версии 21H2 переходим в папку install.esd\1\Windows\System32\, в архиве версии 1507 — в install.wim\Windows\System32\ из которого нужно выгрузить только определенные файлы.

Извлекаем файлы, указанные на рисунках 1–4 из образов в папку C:\BuildPE\AddFiles\x32\Windows\System32. Все файлы, кроме указанных на рисунке 4 извлечены из директории ru-RU образа 1507.

 explorerframe.dll.mui	07.12.2019 15:08	Файл "MUI"	27 КБ
 ole32.dll.mui	06.10.2021 16:00	Файл "MUI"	32 КБ
 oledlg.dll.mui	07.12.2019 15:08	Файл "MUI"	30 КБ
 shell32.dll.mui	06.10.2021 16:00	Файл "MUI"	466 КБ
 taskkill.exe.mui	07.12.2019 15:08	Файл "MUI"	18 КБ
 twinapi.appcore.dll.mui	07.12.2019 15:08	Файл "MUI"	7 КБ
 twinapi.dll.mui	07.12.2019 15:08	Файл "MUI"	5 КБ
 twinui.appcore.dll.mui	07.12.2019 15:08	Файл "MUI"	52 КБ
 twinui.dll.mui	07.12.2019 15:08	Файл "MUI"	86 КБ

Рис. 1. Файлы, взятые из ru-RU

Файл shellstyle.dll.mui находится в дочерней папке en-US в 1507.

 Open-Shell	17.12.2025 22:08	Папка с файлами	
 History.txt	28.01.2024 17:49	Text Document	56 КБ
 License.txt	28.01.2024 17:49	Text Document	33 КБ
 Readme.txt	28.01.2024 17:49	Text Document	1 КБ
 config.xml	17.12.2025 19:39	xmlfile	16 КБ
 UI.bat	17.12.2025 21:47	Пакетный файл Win...	1 КБ
 Explorer++.exe	28.01.2024 17:53	Приложение	3 523 КБ
 taskkill.exe	10.07.2015 6:24	Приложение	95 КБ
 timeout.exe	10.07.2015 6:14	Приложение	30 КБ
 WinXShell.exe	17.12.2025 21:10	Приложение	3 104 КБ
 actxprxy.dll	10.09.2015 8:08	Расширение прилож...	4 504 КБ
 chartv.dll	10.07.2015 6:15	Расширение прилож...	132 КБ
 d3d11.dll	10.07.2015 7:41	Расширение прилож...	2 686 КБ
 DataExchange.dll	10.07.2015 6:19	Расширение прилож...	263 КБ
 dcomp.dll	10.07.2015 8:08	Расширение прилож...	820 КБ
 ddraw.dll	10.07.2015 6:17	Расширение прилож...	560 КБ
 DXCore.dll	06.10.2021 15:58	Расширение прилож...	160 КБ

Рис. 2. Извлечение файлов из образа

Рубрика 1. Информатика и информационные процессы

 dxgi.dll	10.09.2015 8:08	Расширение прилож...	618 КБ
 ExplorerFrame.dll	10.09.2015 8:08	Расширение прилож...	4 649 КБ
 glu32.dll	10.07.2015 6:16	Расширение прилож...	163 КБ
 ksuser.dll	10.07.2015 8:03	Расширение прилож...	22 КБ
 oledlg.dll	10.07.2015 6:20	Расширение прилож...	124 КБ
 opengl32.dll	10.07.2015 6:16	Расширение прилож...	1 055 КБ
 pdh.dll	10.07.2015 6:15	Расширение прилож...	292 КБ
 pdhui.dll	10.07.2015 6:14	Расширение прилож...	56 КБ
 shellstyle.dll	10.07.2015 6:19	Расширение прилож...	1 129 КБ
 shfolder.dll	10.07.2015 6:18	Расширение прилож...	11 КБ
 thumbcache.dll	10.07.2015 6:19	Расширение прилож...	285 КБ
 twinapi.appcore.dll	10.09.2015 8:08	Расширение прилож...	944 КБ
 twinapi.dll	10.07.2015 6:20	Расширение прилож...	720 КБ
 twinui.appcore.dll	10.09.2015 8:08	Расширение прилож...	2 076 КБ
 twinui.dll	10.09.2015 8:08	Расширение прилож...	11 287 КБ
 Explorer++ Help.chm	28.01.2024 17:56	Скомпилированный...	699 КБ
 startnet.cmd	17.12.2025 19:39	Сценарий Windows	2 КБ

Рис. 3. Извлечение файлов из образа

 WinXShell.jcfg	17.12.2025 21:10	Файл "JCFG"	3 КБ
 wallpaper.jpg	17.12.2025 21:10	Файл "JPG"	94 КБ
 WinXShell.lua	17.12.2025 21:10	Файл "LUA"	3 КБ
 explorerframe.dll.mui	07.12.2019 15:08	Файл "MUI"	27 КБ
 ole32.dll.mui	06.10.2021 16:00	Файл "MUI"	32 КБ
 oledlg.dll.mui	07.12.2019 15:08	Файл "MUI"	30 КБ
 shell32.dll.mui	06.10.2021 16:00	Файл "MUI"	466 КБ
 shellstyle.dll.mui	10.09.2015 8:04	Файл "MUI"	82 КБ
 taskkill.exe.mui	07.12.2019 15:08	Файл "MUI"	18 КБ
 twinapi.appcore.dll.mui	07.12.2019 15:08	Файл "MUI"	7 КБ
 twinapi.dll.mui	07.12.2019 15:08	Файл "MUI"	5 КБ
 twinui.appcore.dll.mui	07.12.2019 15:08	Файл "MUI"	52 КБ
 twinui.dll.mui	07.12.2019 15:08	Файл "MUI"	86 КБ
 spectraconfig.reg	17.12.2025 19:39	Файл реестра	82 КБ

Рис. 4. Извлечение файлов из образа

Помимо этих файлов нам необходим оригинальный образ, его извлекаем из папки sources. После чего удаляем всё содержимое папки. Это сделано для удобства тестирования, позже в эту папку будет добавлен измененный файл boot.wim.

Далее необходим набор файлов, включающий spectraconfig.reg — файл реестра, содержащий предварительно настроенные ветки реестра для корректной работы среды. В данном случае изменения относятся не к активной системе, а к автономному кусту реестра смонтированного образа Windows PE, то есть к файлу C:\BuildPE\Mount\Windows\System32\config\SOFTWARE. Через утилиту offlineereg в него импортируются параметры, необходимые для работы графической оболочки WinXShell, регистрации используемых библиотек оболочки и, при использовании подходящего разрядного образа, настроек WoW64. В рабочем bat-файле эта операция задается командой вида: offlineereg-win64 C:\BuildPE\Mount\Windows\System32\config\SOFTWARE "" run C:\BuildPE\util\regexport32.txt. По смыслу изменения затрагивают ветви HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion и связанные разделы, отвечающие за запуск компонентов оболочки, ассоциации системных библиотек и параметры

пользовательской среды. `startnet.cmd` — скрипт, который выполняется при старте WinPE и запускает необходимые службы, драйверы и графическую оболочку. В минимальном варианте он последовательно вызывает инициализацию среды, переходит в каталог `X:\Windows\System32` и запускает `WinXShell.exe` с параметром `-winpe`. Также используются ярлыки для портативных утилит, размещенные в меню и на рабочем столе, чтобы администратор мог быстро открыть файловый менеджер, архиватор, текстовый редактор и командную строку.

Фрагмент применяемой логики настройки можно описать следующим образом: сначала в смонтированном образе создается каталог `Desktop` для профиля `systemprofile`, затем из стандартного меню удаляются лишние ярлыки, после чего в автономный раздел `SOFTWARE` импортируются параметры из подготовленного `reg`-файла. Такой порядок нужен для воспроизводимости: запуск оболочки не зависит от ручных действий после загрузки, а необходимые настройки уже находятся внутри WIM-образа. При этом зависимость от сторонних скриптов остается ограниченной: `startnet.cmd` отвечает только за первичный запуск среды, а `spectraconfig.reg` фиксирует заранее подготовленные параметры оболочки и библиотек. Это не полностью устраняет риск ошибок в `reg`-файле, но делает процесс более проверяемым, так как все изменения вносятся до финальной сборки и могут быть повторно протестированы в виртуальной машине.

Для проверки работы образа установим виртуальную машину (в моем случае это VMware Workstation), и создадим новую машину. При указании объема лучше всего выбрать `Store virtual disk as a single file`, это облегчит задачу, когда мы будем извлекать из диска нужные файлы. Проверка в виртуальной среде также позволяет заранее оценить корректность работы накопителей, графической оболочки и базовых периферийных устройств, что важно при дальнейшем использовании сервисного диска [7].

Далее займемся созданием своих образов. Для этого открываем от имени администратора Среду средств развертывания и работы с образами. В ней нам необходимо экспортировать оригинальный файл `boot.wim` и смонтировать образ `edit.wim` для внесения изменений. Командами `Dism /Get-WimInfo /WimFile:"C:\BuildPE\originalWIM\x32\boot.wim"` и `DISM /Cleanup-Wim` мы можем отобразить редакции в `wim` файле и отключить `wim` файлы в случае, если они были ранее подключены.

```

Администратор: Среда средств развертывания и работы с образами
C:\Windows\System32>Dism /Get-WimInfo /WimFile:"C:\BuildPE\originalWIM\x64\boot.wim"

Система DISM
Версия: 10.0.26100.5074

Сведения об образе : C:\BuildPE\originalWIM\x64\boot.wim

Индекс: 1
Имя : Microsoft Windows PE (amd64)
Описание : Microsoft Windows PE (amd64)
Размер (байт): 2 485 369 083

Индекс: 2
Имя : Microsoft Windows Setup (amd64)
Описание : Microsoft Windows Setup (amd64)
Размер (байт): 2 595 260 019

Операция успешно завершена.

C:\Windows\System32>DISM /Cleanup-Wim

Система DISM
Версия: 10.0.26100.5074

Удалена устаревшая точка подключения, указывающая на D:\PB Download\x86\Mount77960250
Поиск устаревших файлов на диске C
Поиск устаревших файлов на диске D
Операция успешно завершена.

C:\Windows\System32>
    
```

Рис. 5. Отключение `wim` файлов

Рубрика 1. Информатика и информационные процессы

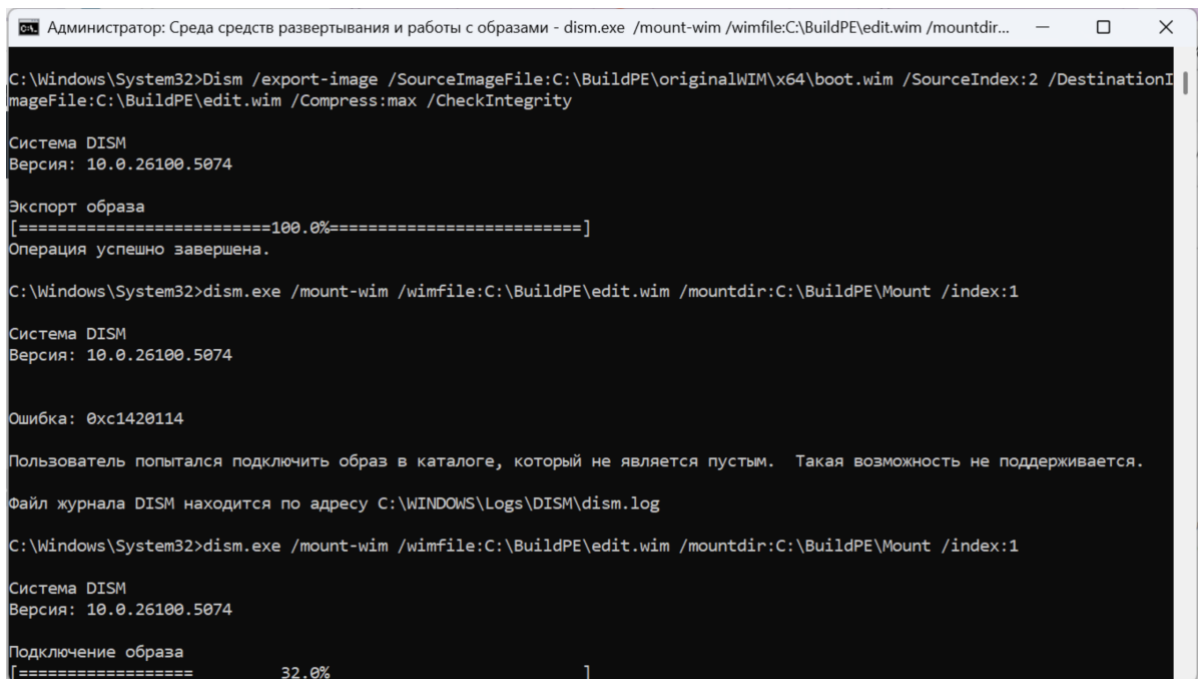


Рис. 6. Монтрование образа

Прав администратора недостаточно для удаления лишних файлов в образе, чтобы повысить себе права ещё выше, запустим от имени администратора Explorer++, откроем папку C:\BuildPE\ и находясь в ней нажмем на \, после чего откроем командную строку через значок черного квадрата вверху окна. В командной строке вводим следующие команды:

```
erase C:\BuildPE\Mount\Setup.exe
takeown /F C:\BuildPE\Mount\sources\*. * /A /R /D Y
icacls C:\BuildPE\Mount\sources /grant Администраторы:F /T /C /Q
rd C:\BuildPE\Mount\sources\ /s/q
```

Данные действия необходимы для удаления программы установки операционной системы.

```
rd C:\BuildPE\Mount\sources\ /s/q
```

Далее создаём каталог «Рабочий стол» для пользователя systemprofile без которого Explorer++ может выдавать ошибку. Удаляем ненужные ярлыки из меню Пуск чтобы не видеть их в меню WinXShell. Переходим в каталог, где хранится offlineereg и вносим изменения в реестр. Копируем весь набор своего софта+доп файлы для ОС. Для удобства можно вынести команды в файл с разрешением .bat, тогда при запуске файла из него будут проиграны все команды, прописанные внутри него.

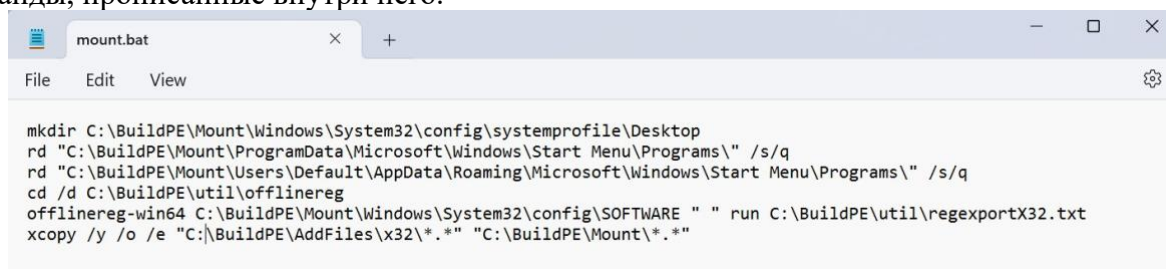


Рис. 7. Модификация смонтированного образа

Возвращаемся в «Среду средств развертывания и работы с образами» и размонтируем образ

```
Dism /Unmount-Image /MountDir:"C:\BuildPE\Mount" /commit
DISM /Cleanup-Wim
```

Далее необходимо экспортировать готовый образ в новый .wim файл с целью избавиться от папки [DELETED] который создаётся после редактирования .wim файла

```
DISM /export-image /sourceimagefile:C:\BuildPE\edit.wim /sourceindex:1
/destinationimagefile:C:\BuildPE\COMPLETE\32\boot.wim /compress:maximum
```

Следом, вернём измененный файл boot.wim в образ ISO. Для этого так же открываем редактор ISO файлов и в пустую папку sources переносим файл [5].

Сборка и настройка образа готова, протестируем образ.

Результаты исследования

Для того, чтобы виртуальная машина начала подгружаться с нашего ISO файла необходимо его указать в настройках машины (CD/DVD (SATA)); такой способ проверки соответствует логике предварительного тестирования загрузочного носителя перед записью на USB или применением на реальном компьютере [8].

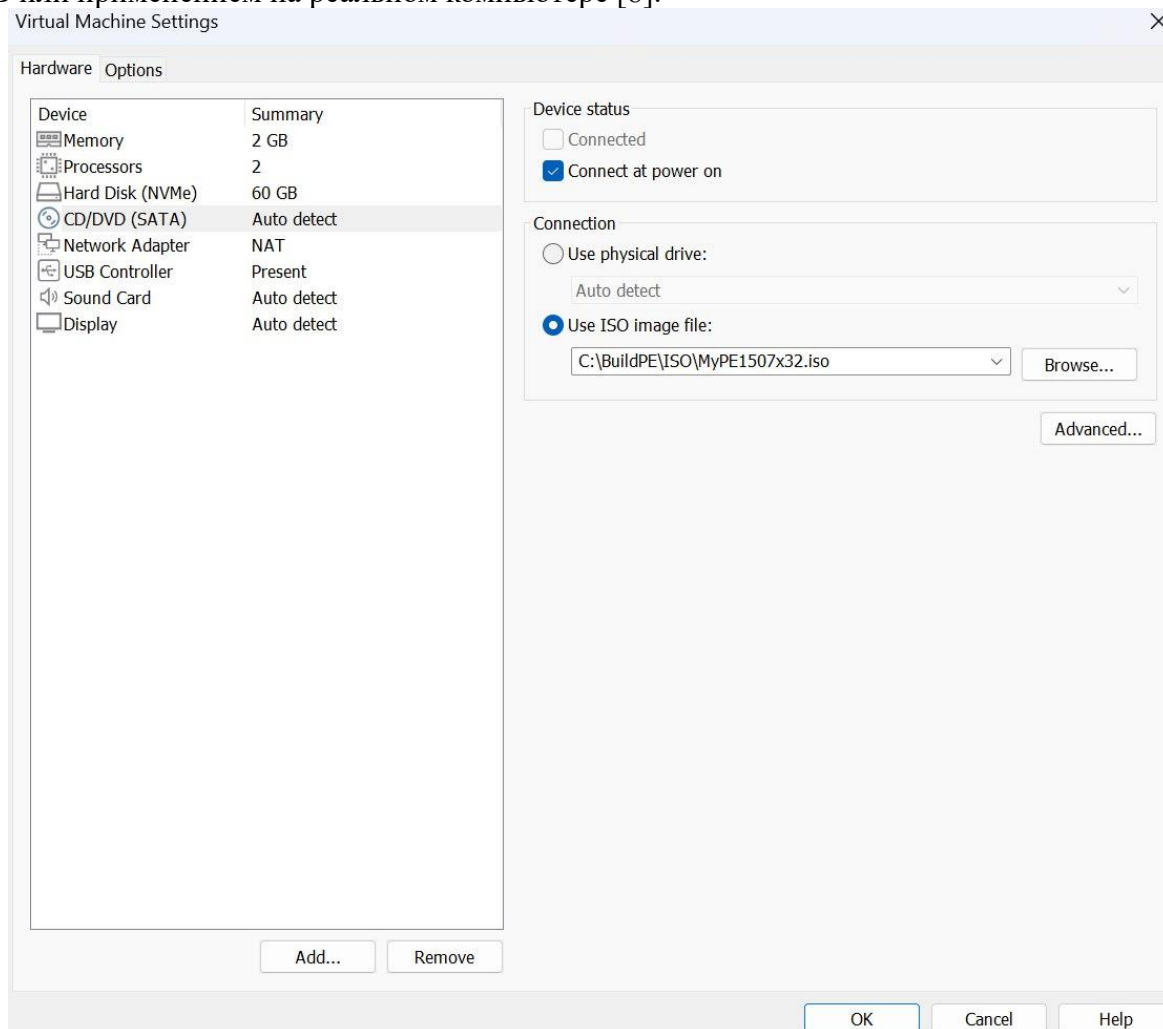


Рис. 8. Указание ISO файла

После включения машины появится командная строка, в которой необходимо выполнить имеющийся скачанный файл UI.bat. В нем лежит последовательность команд, запускающих WinXShell — графическую оболочку.

```
X:\Windows\System32>UI.bat

X:\Windows\System32>cd /d "X:\Windows\System32"

X:\Windows\System32>start WinXShell.exe -winpe

X:\Windows\System32>
```

Рис. 9. Запуск графического интерфейса

В результате получим рабочий стол Windows, где есть всё необходимое.

Рубрика 1. Информатика и информационные процессы

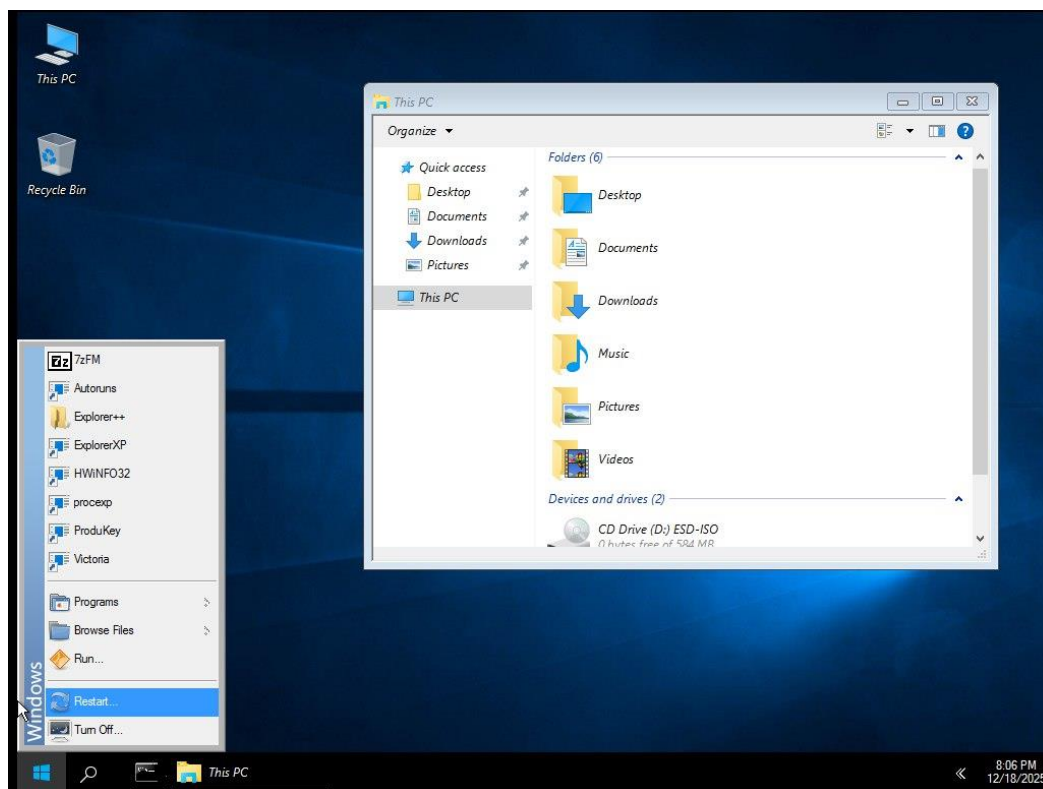


Рис. 10. Проверка работы программ

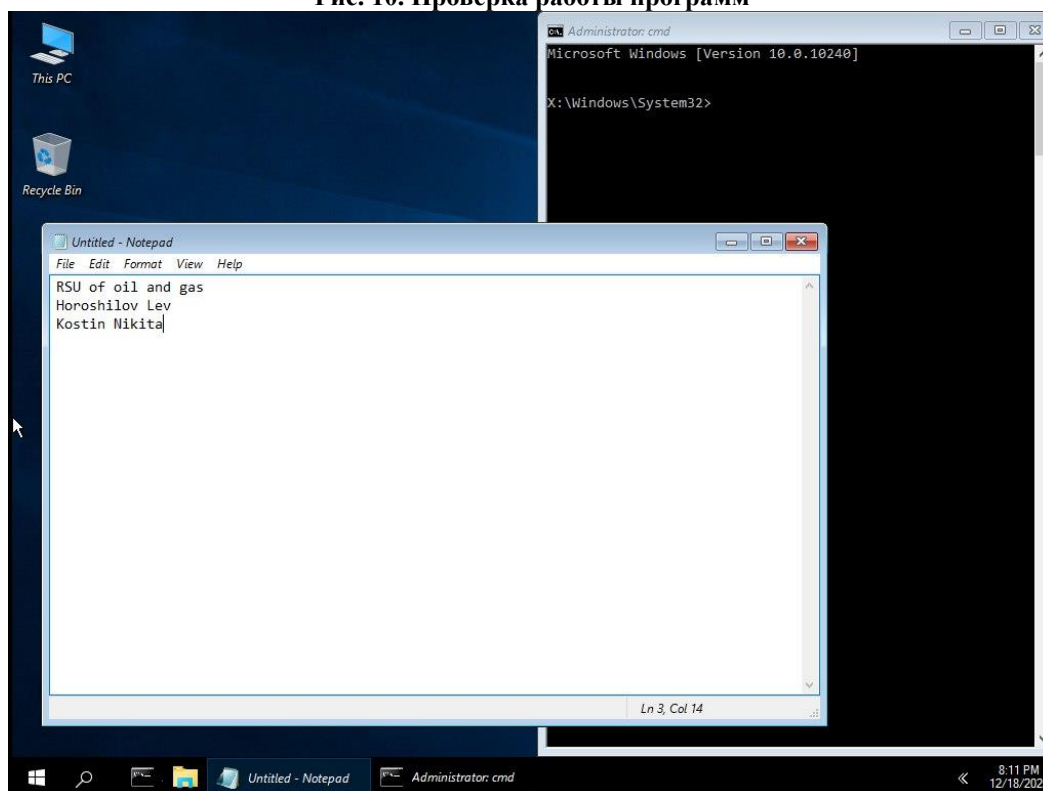


Рис. 11. Результат создания образа Windows PE

В рамках практического тестирования возможностей созданного образа Windows PE был успешно выполнен сценарий сброса пароля локальной учётной записи администратора на тестовой виртуальной машине. Доступ к системе был восстановлен без потери данных, что подтвердило готовность среды к выполнению реальных задач аварийного обслуживания.

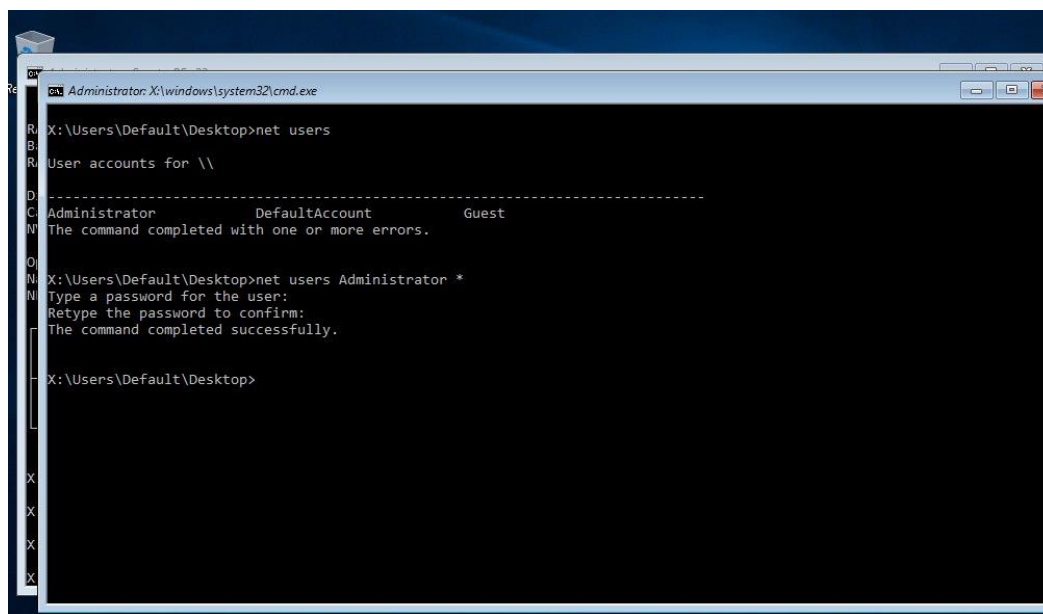


Рис. 12. Сброс пароля локальной учетной записи администратора

Заключение

В результате проведенного исследования разработана комплексная методика создания специализированного образа предустановленной среды Windows PE, ориентированного на задачи обслуживания, диагностики и аварийного восстановления компьютерных систем. Основным результатом работы является пошаговая технология, охватывающая все ключевые этапы сборки:

1. Подготовку инструментальной базы (Windows ADK) и структурированного рабочего пространства;
2. Расширение базового образа WinPE за счет интеграции недостающих системных библиотек и файлов, необходимых для работы графического интерфейса и базовых функций проводника;
3. Методы правки автономного реестра для регистрации системных компонентов и предварительной настройки среды;
4. Процедуру модификации WIM-образа с помощью DISM, его настройку и последующее создание загрузочного ISO-носителя с учетом правил работы с образами и носителями [8].

Результат проверки гипотез:

1. Гипотеза Н1 подтвердилась полностью: комбинация инструментов ADK/DISM и методов сообщества позволила создать рабочий образ. Образ стабилен, функционален (с графическим интерфейсом, диагностическими утилитами) и сохранил компактность, превосходя стандартную WinPE;
2. Гипотеза Н2 подтверждается тем, что разработанная методика предусматривает явное описание и обоснование каждого включаемого в образ элемента. Все системные файлы, библиотеки и настройки реестра добавлены целенаправленно, что исключает наличие скрытых или избыточных компонентов. Это обеспечивает как предсказуемость поведения среды, так и её пригодность для использования в ответственных сценариях обслуживания, где важны контроль и прозрачность.

Библиографический список

1. Microsoft. (2025). Windows PE (WinPE). URL: <https://learn.microsoft.com/ru-ru/windows-hardware/manufacture/desktop/winpe-intro?view=windows-11> (дата обращения: 10.12.2025).
2. Трасу. (2025). Два метода создания загрузочного диска WinPE в Windows 10 эффективно. AOMEI. URL: <https://www.diskpart.com/ru/articles/winpe-bootable-disk-windows-10-5740.html> (дата обращения: 12.12.2025).

Рубрика 1. Информатика и информационные процессы

3. spectralwhite. (2021). Загрузочный диск Windows PE своими руками. Хабр. URL: <https://habr.com/ru/articles/598001/> (дата обращения: 16.12.2025).
4. Стеркин, В. (2009). Создание базового загрузочного диска Windows PE 3.1. URL: <https://www.outsidethebox.ms/9950/> (дата обращения: 13.12.2025).
5. Верещака, Е. (2021). Как смонтировать ISO-образ и подключить его в системе Windows 10: 4 способа. IT-Техник. URL: <https://it-tehnik.ru/windows10/ispolzovanie/kak-smontirovat-obraz-iso.html> (дата обращения: 15.12.2025).
6. Соловьев, Д. (2024). Что такое Windows PE и как он работает. Wondershare Recoverit. URL: <https://recoverit.wondershare.com.ru/windows-pe/what-is-windows-pe-and-how-it-works.html> (дата обращения: 11.12.2025).
7. Уймин, А. Г. (2023). Периферийные устройства ЭВМ: практикум. Москва.
8. Microsoft. (2025). Создание загрузочного носителя Windows PE. URL: <https://learn.microsoft.com/ru-ru/windows-hardware/manufacture/desktop/winpe-create-usb-bootable-drive?view=windows-11> (дата обращения: 15.12.2025).

References

1. Microsoft. (2025). Windows PE (WinPE). <https://learn.microsoft.com/en-us/windows-hardware/manufacture/desktop/winpe-intro?view=windows-11> (accessed December 10, 2025).
2. Tracy. (2025). Two efficient methods to create a WinPE bootable disk in Windows 10. AOMEI. <https://www.diskpart.com/ru/articles/winpe-bootable-disk-windows-10-5740.html> (accessed December 12, 2025).
3. spectralwhite. (2021). Creating a Windows PE bootable disk by yourself. Habr. <https://habr.com/en/articles/598001/> (accessed December 16, 2025).
4. Sterkin, V. (2009). Creating a basic Windows PE 3.1 bootable disk. <https://www.outsidethebox.ms/9950/> (accessed December 13, 2025).
5. Vereshchaka, E. (2021). How to mount an ISO image in Windows 10: Four methods. IT-Tehnik. <https://it-tehnik.ru/windows10/ispolzovanie/kak-smontirovat-obraz-iso.html> (accessed December 15, 2025).
6. Solovev, D. (2024). What is Windows PE and how it works. Wondershare Recoverit. <https://recoverit.wondershare.com.ru/windows-pe/what-is-windows-pe-and-how-it-works.html> (accessed December 11, 2025).
7. Uimin, A. G. (2023). Computer peripherals: Practical guide. Moscow.
8. Microsoft. (2025). Create bootable Windows PE media. <https://learn.microsoft.com/en-us/windows-hardware/manufacture/desktop/winpe-create-usb-bootable-drive?view=windows-11> (accessed December 15, 2025).

Информация об авторах

Хорошилов Лев Геннадьевич — студент, ФГАОУ ВО «РГУ нефти и газа (НИУ) имени И.М. Губкина», г. Москва, e-mail: levyahorosh@yandex.ru

Костин Никита Андреевич — студент, ФГАОУ ВО «РГУ нефти и газа (НИУ) имени И.М. Губкина», г. Москва, e-mail: knikita200558@mail.ru

ASSEMBLY AND CONFIGURATION OF A WINDOWS PE IMAGE FOR SYSTEM MAINTENANCE

Khoroshilov L. G.¹, Kostin N. A.¹

¹National University of Oil and Gas «Gubkin University», Moscow, Russian Federation

Abstract. *The article examines the problem of creating and configuring a Windows PE image with a set of software tools for system maintenance. The relevance of the topic is determined by the need for an autonomous tool that can be used for diagnostics, recovery, and administration of computer systems without booting the main operating system. Special attention is paid to the practical stages of preparing the working environment, selecting portable programs, adding system files, and configuring components that ensure the launch of the graphical interface. The proposed method is based on editing the official Windows image and integrating the required programs and files into it. The paper describes the use of Windows ADK and DISM tools, mounting and modifying a WIM image, editing the offline registry, configuring startup scripts, and subsequently building a bootable ISO media. The testing of the created image in a virtual machine is considered separately, confirming its operability, stability of the graphical shell, and suitability for emergency maintenance tasks. The practical value of the study lies in the ability to flexibly modify the service environment, exclude unnecessary components, and add only those utilities required for specific operating scenarios. This approach improves the convenience of system maintenance and makes the Windows recovery process more controlled for administration. The developed methodology is applied in nature and allows system administrators to create lightweight, autonomous, and functionally rich service disks tailored to specific tasks.*

Keywords: *Windows PE, image assembly, pre-installed environment, system maintenance, system administration, bootable ISO media.*

Information about the authors

Khoroshilov Lev Gennadievich — student, Gubkin Russian State University of Oil and Gas (National Research University), Moscow, e-mail: levyahorosh@yandex.ru

Kostin Nikita Andreevich — student, Gubkin Russian State University of Oil and Gas (National Research University), Moscow, e-mail: knikita200558@mail.ru