

Создание script-файла для установки ПО

Фролов А.А., 2-к, ИВТ-2

Теоретическая часть

В ходе выполнения задания по автоматизации установки и настройки прикладного программного обеспечения и утилит для компьютерных аудиторий кафедры информационных технологий и электронного обучения я изучил несколько способов автоматизации установки ПО на операционной системе Windows.

Сначала я рассматривал классические методы: использование **.bat-файлов** и **.msi-инсталляторов**. Принцип их работы достаточно прост: скрипт содержит последовательность команд для запуска установщиков с необходимыми параметрами. Например, для MSI-файлов можно использовать команду `msiexec /i путь_к_файлу.msi /quiet`, что позволяет установить программу в тихом режиме без вмешательства пользователя. Bat-скрипты позволяют запускать несколько установок последовательно, но при этом требуют ручного управления ссылками на установочные файлы и параметрами запуска.

Однако такой подход имеет несколько существенных недостатков:

1. **Сложность поддержки.** Каждый новый пакет требует ручного добавления команд в скрипт, а при обновлении версий ПО скрипт нужно править.
2. **Отсутствие централизованного управления зависимостями.** Например, если одна программа требует установки другой, в bat-файле это нужно прописывать вручную.
3. **Риск ошибок при скачивании и установке с разных сайтов**, особенно если ссылки устаревают или файлы перемещаются.

Для упрощения процесса я решил использовать **Chocolatey (choco)** — пакетный менеджер для Windows. Этот инструмент позволяет автоматизировать установку большого числа приложений с минимальной настройкой. Основные причины выбора Chocolatey:

- **Автоматическая установка и обновление.** Choco скачивает и устанавливает пакеты напрямую с официальных источников или проверенных репозиториев.
- **Поддержка пакетного подхода.** Можно создать CSV или текстовый файл со списком нужного ПО, и скрипт автоматически установит всё в нужной последовательности.
- **Простая интеграция с PowerShell.** Скрипты легко читаются и поддерживаются, можно добавлять условия, проверку аудитории и обработку

ссылок.

- **Минимизация ручной работы.** Не нужно контролировать каждый скачанный установщик или вручную прописывать параметры для тихой установки.
- **Поддержка бэкапа и проверок.** Можно заранее сохранять список установленных пакетов и создавать резервные копии системы, что важно для работы в компьютерных аудиториях.

Таким образом, использование Chocolatey позволяет реализовать **эффективную, безопасную и легко поддерживаемую систему автоматизации установки прикладного ПО**. Это особенно важно для работы с большим количеством компьютеров, где ручная установка каждой программы занимает много времени и подвержена ошибкам.

В дальнейшем в проекте я создал скрипт на PowerShell, который:

- Проверяет и при необходимости устанавливает Chocolatey
- Импортирует список программ из CSV
- Проверяет аудиторию и устанавливает ПО только для соответствующих пользователей
- Создаёт резервную копию системы и данных перед установкой
- Устанавливает приложения через choco или по ссылкам с официальных сайтов, включая обработку ссылок на Google Drive с подтверждением

CSV - файл

```
1  AppName;Source;Extensions;Audience
2  Visual Studio Code;choco:vscode;ms-python.python, ms-vscode.cpptools, ms-
  azuretools.vscode-docker, ecmel.vscode-html-css, eamodio.gitlens, ms-
  vscode.vscode-typescript-next;all
3  Docker;choco:docker;;all
4  PyCharm Community Edition;choco:pycharm-community;;all
5  Git;choco:git;;all
6  GitHub Desktop;choco:github-desktop;;all
7  Maxima;choco:maxima;;all
8  KNIME Analytics Platform;choco:knime.install;;all
9  GIMP;choco:gimp;;all
10 Julia (+ VSCode);choco:julia;;all
11 Python (+ VSCode);choco:python;;all
12 Rust (+ VSCode);choco:rust;;all
13 MSYS2 UCRT64 Shell;choco:msys2;;all
14 Zettlr;choco:zettlr;;all
15 MiKTeX;choco:miktex;;all
16 TeXstudio;choco:texstudio;;all
17 Anaconda;choco:anaconda3;;all
18 Far Manager;choco:far-3;;all
19 SumatraPDF;choco:sumatrapdf;;all
20 Chrome;choco:googlechrome;;all
```

```

21 Flameshot;choco:flameshot;;all
22 Windows для Linux – WSL 2 (Ubuntu 22.04 и 24.04);choco:wsl-ubuntu-
23 2204;;all
24 Qalculate!;choco:qalculate;;all
25 Yandex.Telemost;https://telemost.yandex.ru/download-desktop;;all
26 Sber Jazz;https://dl.salutejazz.ru/desktop/latest/jazz.exe;;all
27 Arc;https://arc.net/downloaded;;all
28 7Zip;choco:7zip;;all
29 Firefox;choco:firefox;;all
30 Yandex Browser;choco:yandex-browser;;all
31 Edge;choco:microsoft-edge;;all
32 Ramus Educational;https://drive.usercontent.google.com/download?
33 id=1ygbqxneypuixRfr0bqT53raBlCUB6Kxz&export=download&authuser=0;;313
34 ARIS EXPRESS;https://drive.usercontent.google.com/u/0/uc?id=1sP0ZvSiLDDYx-
35 RYqluLbVIfr1Vy3Qvzk&export=download;;313
36 Archi;choco:Archi;;313

```

PowerShell скрипт

```

1  # === 0. Настройка переменных ===
2  $timestamp = (Get-Date).ToString('yyyyMMdd_HH:mm:ss')
3  $backupFolder = Join-Path -Path $env:USERPROFILE -ChildPath
4  "Desktop\Backup_$timestamp"
5  New-Item -Path $backupFolder -ItemType Directory -Force | Out-Null
6  Write-Host "Создана папка для бэкапа: $backupFolder"
7
8  # === 1. Общий бэкап ===
9  try {
10     if (Get-Command choco -ErrorAction SilentlyContinue) {
11         choco list --local-only | Out-File (Join-Path $backupFolder
12         'InstalledPackages.txt') -ErrorAction Stop
13         Write-Host "Список установленных пакетов сохранен"
14     }
15 } catch {
16     Write-Warning "Не удалось получить список пакетов choco: $($_.Exception)
17 }
18
19 $foldersToBackup = @("$env:USERPROFILE\AppData\Roaming",
20 "$env:USERPROFILE\AppData\Local", "$env:USERPROFILE\Documents")
21
22 foreach ($folder in $foldersToBackup) {
23     if (Test-Path $folder) {
24         $folderName = Split-Path $folder -Leaf
25         $dest = Join-Path $backupFolder $folderName
26         try {
27             Write-Host "Копируем $folderName..."
28             Copy-Item -Path $folder -Destination $dest -Recurse -Force -Er
29             Stop
30         } catch {

```

```

27         Write-Warning ("Не удалось скопировать " + $folderName + ". Ош
$_.Exception.Message)
28     }
29 }
30 }
31
32 # === 2. Установка Chocolatey ===
33 if (-not (Get-Command choco -ErrorAction SilentlyContinue)) {
34     Write-Host "Устанавливаем Chocolatey..."
35     try {
36         Set-ExecutionPolicy Bypass -Scope Process -Force
37         [System.Net.ServicePointManager]::SecurityProtocol =
[System.Net.ServicePointManager]::SecurityProtocol -bor 3072
38         Invoke-Expression ((New-Object
System.Net.WebClient).DownloadString('https://community.chocolatey.org/ins
39         $env:PATH += ";C:\ProgramData\chocolatey\bin"
40     } catch {
41         Write-Warning ("Ошибка установки Chocolatey. Сообщение: " +
$_.Exception.Message)
42     }
43 } else {
44     Write-Host "Chocolatey уже установлен"
45 }
46
47 # === 3. Запрос аудитории ===
48 $currentUserAudience = Read-Host "Введите аудиторию (313 или пусто для все
49 $currentUserAudience = $currentUserAudience.Trim()
50
51 # === 4. Импорт CSV ===
52 $csvPath = ".\apps.csv"
53 if (-not (Test-Path $csvPath)) {
54     Write-Error "Файл $csvPath не найден"
55     exit 1
56 }
57
58 try {
59     $apps = Import-Csv -Path $csvPath -Delimiter ';'
60 } catch {
61     Write-Error "Ошибка загрузки CSV: $($_.Exception.Message)"
62     exit 1
63 }
64
65 function Test-ChocoInstalled {
66     param([string]$package)
67     try {
68         $result = choco list --local-only --limit-output | Where-Object {
"$package*" }
69         return [bool]$result
70     } catch {
71         return $false

```

```

72     }
73 }
74
75 # === 5. Установка приложений ===
76 foreach ($app in $apps) {
77     $appName = if ($app.AppName) { $app.AppName.Trim() } else { $null }
78     $source = if ($app.Source) { $app.Source.Trim() } else { $null }
79     $audience = if ($app.Audience) { $app.Audience.Trim() } else { "all" }
80
81     if (-not $appName -or -not $source) { continue }
82
83     if ($audience -ne "all" -and $audience -ne $currentUserAudience) {
84         Write-Host "Пропускаем $appName (аудитория: $audience)"
85         continue
86     }
87
88     Write-Host "Обрабатываем: $appName"
89
90     if ($source -like 'choco:*') {
91         $pkg = $source -replace '^choco:\s*', ''
92         if (Test-ChocoInstalled $pkg) {
93             Write-Host "$appName уже установлен"
94         } else {
95             Write-Host "Устанавливаем $pkg..."
96             choco install $pkg -y
97         }
98     }
99     elseif ($source -like 'http*') {
100         $safeName = $appName -replace '[\\\/:*?"<>|]', '_'
101         $uriExt = [IO.Path]::GetExtension($source)
102         if (-not $uriExt) { $uriExt = '.exe' }
103         $fileName = Join-Path $env:TEMP "$safeName$uriExt"
104         $url = $source
105
106         if ($url -match 'drive\.google\.com' -and $url -notmatch 'confirm=')
107             $url = if ($url -match '\?') { "$url&confirm=t" } else { "$url
108     }
109
110     Write-Host "Скачиваем: $url"
111     try {
112         Invoke-WebRequest -Uri $url -OutFile $fileName -ErrorAction St
113         Start-Process -FilePath $fileName -Wait
114         Remove-Item $fileName -Force -ErrorAction SilentlyContinue
115     } catch {
116         Write-Warning ("Ошибка загрузки " + ${appName} + ". Сообщение:
117     $_.Exception.Message)
118 }
119 }

```

120

121 Write-Host "Установка завершена. Бэкап: \$backupFolder"

Описание процесса

Для начала я установил choco на свой компьютер, после чего занялся созданием таблицы в Excel. Для каждой программы из списка я искал package через `choco find "name"`. Для некоторых программ не было найдено пакетов, но была ссылка с подтверждением на Гугл диск, что потребовало добавления флага `confirm=t`. Добавил столбец с аудиторией, так как есть программы, нужные лишь в 313 аудитории. Заполнив таблицу, я сохранил ее как csv-файл.

Далее я занялся написанием скрипта. Сначала я решил, что нужно сделать общий бэкап системы, для того чтобы в случае неисправностей можно было легко все решить. Далее, так как я выбрал choco, нужно его установить.

Указал разделитель как ';'.

После чего скрипт проверяет package, если это choco, то устанавливает программу через него, если ссылка, то устанавливает через нее с подтверждением. Отдельно прописана установка с Google Drive с подтверждением. Так же добавил установку расширений для VS Code.

После написания скрипта, я скачал Virtual Box и создал виртуальную машину на Windows 10, для тестирования скрипта. Пару раз столкнулся с ошибками в синтаксисе, но в итоге исправил и все заработало (в отчете представлена финальная версия).

```
Chocolatey v2.5.1
Installing the following packages:
miktex
By installing, you accept licenses for the packages.
Downloading package from source 'https://community.chocolatey.org/api/v2/'
Progress: Downloading miktex.install 25.3.0... 100%

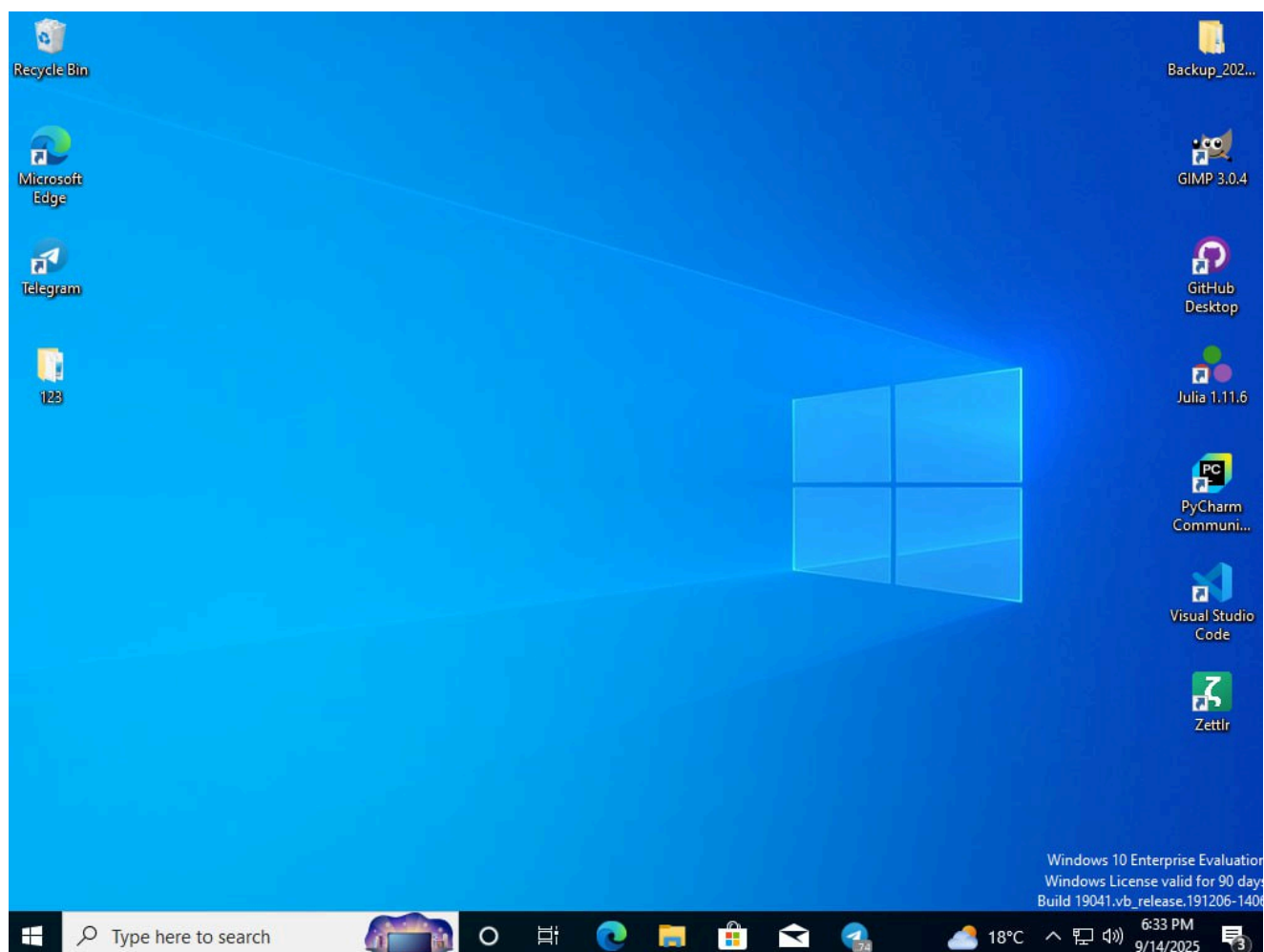
miktex.install v25.3.0 [Approved]
miktex.install package files install completed. Performing other installation steps.
Extracting 64-bit C:\ProgramData\chocolatey\lib\miktex.install\tools\miktexsetup-5.5.0+1763023-x64.zip to C:\ProgramData\chocolatey\lib\miktex.install\tools...
Downloading a "Basic" package set to install.
WARNING: No registry key found based on 'miktex*'
Creating a temporary repository at 'C:\Users\Andrew\AppData\Local\Temp\chocolatey\MiKTeX-repository'.
Installing from temporary MiKTeX repository for all users.
MiKTeX milestone 25.4 installed
miktex.install may be able to be automatically uninstalled.
Environment Vars (like PATH) have changed. Close/reopen your shell to
see the changes (or in powershell/cmd.exe just type 'refreshenv').
The install of miktex.install was successful.
  Deployed to 'C:\Program Files\MiKTeX'
Downloading package from source 'https://community.chocolatey.org/api/v2/'
Progress: Downloading miktex 25.3.0... 100%

miktex v25.3.0 [Approved] - Possibly broken
miktex package files install completed. Performing other installation steps.
The install of miktex was successful.
  Software install location not explicitly set, it could be in package or
  default install location of installer.

Chocolatey installed 2/2 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).

Enjoy using Chocolatey? Explore more amazing features to take your
experience to the next level at
https://chocolatey.org/compare
DEB+NED+D+N,N<D%D%D%: TeXstudio
DEB+N,D%D%D%D%D,D%D%D%D% texstudio...
Chocolatey v2.5.1
Installing the following packages:
texstudio
By installing, you accept licenses for the packages.
Downloading package from source 'https://community.chocolatey.org/api/v2/'
Progress: Downloading texstudio.install 4.8.9... 77%
```

Процесс установки



Промежуточный итог