

если в повершл ввести clean
что будет?

Думал на протяжении 9s >

Если прямо в PowerShell (в обычной строке PS C:\...>) набрать:

powershell

clean

 Копировать код

то в 99% случаев будет... ничего полезного 😊

PowerShell просто скажет ошибку типа:

```
clean : The term 'clean' is not recognized as the name of a cmdlet, function,
script file, or operable program...
```

Потому что встроенной команды clean в PowerShell нет.

Но есть важный “подвох”, где clean — очень опасно

Команда clean существует в DiskPart (утилита для работы с дисками). И там это означает:

- clean — удаляет информацию о разделах на выбранном диске (диск станет “пустым/неразмеченным”)
- clean all — ещё хуже: затирает весь диск нулями (долго и почти без шансов на восстановление)

То есть если ты сделаешь что-то вроде:

powershell

diskpart

 Копировать код

а потом внутри diskpart введёшь clean — вот тогда можно “очень красиво” уничтожить разметку диска.

Как проверить, что именно выполнится у тебя

В PowerShell безопасно уточнить:

powershell

`Get-Command clean`

Копировать код

- Если ничего не найдёт — значит `clean` не определён.
- Если найдёт (alias/функция/скрипт) — покажет, что именно запустится.

Если цель была “очистить экран”

Тогда нужно:

powershell

`cls`

Копировать код

или

powershell

`Clear-Host`

Копировать код

Если скажешь, что именно ты хотел “очистить” (экран? временные файлы? корзину? диск?), подскажу правильную команду без режима “случайно снёс себе жизнь”.



Windows PowerShell

(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

Попробуйте новую кроссплатформенную оболочку
PowerShell (<https://aka.ms/pscore6>)

```
PS C:\Windows\system32> cd $env:USERPROFILE\Desktop
>> python ./win11_readiness_report_v2.py --outdir .
>>
==== Windows 11 Readiness / System Inventory Report
(v2) ====
Generated: 2025-12-27T14:15:56
Host: DESKTOP-1OAT63D | User: Славиус | SystemDrive: C:
```

--- Windows ---

Product: Windows 10 Home

Version: 2009 | Build: 19045 | Edition: Core

Firmware (from ComputerInfo): 2

Manufacturer/Model: MSI MS-7817

--- CPU / RAM ---

CPU #1: Intel(R) Core(TM) i3-4170 CPU @ 3.70GHz | Cores:

2 | Threads: 4 | AddrWidth: 64

RAM: 11.95 GB

--- Firmware / Security ---

Boot mode: Uefi

Secure Boot: NO

TPM info: {'gettpm': {'TpmPresent': False, 'TpmReady': False, 'TpmEnabled': False, 'TpmActivated': False, 'TpmOwned': False, 'RestartPending': False, 'ManufacturerId': 0, 'ManufacturerIdTxt': None, 'ManufacturerVersion': None, 'ManufacturerVersionFull20': None, 'ManagedAuthLevel': 4, 'OwnerAuth': None, 'OwnerClearDisabled': True, 'AutoProvisioning': -1, 'LockedOut': False, 'LockoutHealTime': None, 'LockoutCount': None, 'LockoutMax': None, 'SelfTest': None}, 'win32tpm': []}, 'spec_versions': [], 'tpm2_supported': None, 'error': None}

--- Disks ---

Disk 0: P4-240 | 223.57 GB | Style=GPT | Bus=SATA |

IsSystem=True IsBoot=True

Disk 1: SAMSUNG HD161GJ | 149.05 GB | Style=MBR |

Bus=SATA | IsSystem=False IsBoot=False

--- Volumes ---

G: | FS=NTFS | Label= | Size=24.04 GB | Free=17.57 GB |
Health=Healthy

E: | FS=NTFS | Label=Win 7 | Size=30.01 GB | Free=29.93
GB | Health=Healthy

F: | FS=NTFS | Label= | Size=95.0 GB | Free=49.87 GB |
Health=Healthy

C: | FS=NTFS | Label= | Size=111.4 GB | Free=6.78 GB |
Health=Healthy

D: | FS=NTFS | Label=Новый том | Size=111.4 GB |
Free=106.38 GB | Health=Healthy

H: | FS=Unknown | Label= | Size=0.0 GB | Free=0.0 GB |
Health=Healthy

==== Windows 11 Readiness Checks ====

Verdict: NOT READY (some requirements failed)

- [PASS] CPU 64-bit: 64-bit (required: Yes)
- [PASS] RAM: 11.95 GB (required: >= 4 GB)
- [PASS] System drive capacity: 111.4 GB (required: >= 64 GB (Windows 11 requirement))
- [FAIL] System drive free space: 6.78 GB free (required: >= 25 GB (recommended for upgrade headroom))
- [PASS] Boot mode: Uefi (required: UEFI)
- [FAIL] Secure Boot: NO (required: Enabled)
- [FAIL] TPM: Present=NO, Ready=NO, TPM2=UNKNOWN (required: TPM 2.0 present & ready)
- [PASS] System disk partition style: Disk 0: GPT (required: GPT (recommended/typical for UEFI + Win11))

Next steps (based on blockers):

- * Free up space on C: aim for 25–30 GB free before upgrade.
- * Enable Secure Boot in UEFI settings.
- * Enable TPM 2.0 in BIOS/UEFI (Intel PTT / AMD fTPM).

Saved:

-
C:\Users\Славиус\Desktop\win11_readiness_v2_DESKTOP-1OAT63D_20251227_141624.txt

-
C:\Users\Славиус\Desktop\win11_readiness_v2_DESKTOP-1OAT63D_20251227_141624.json
PS C:\Users\Славиус\Desktop>
PS C:\Users\Славиус\Desktop>
PS C:\Users\Славиус\Desktop>
PS C:\Users\Славиус\Desktop> powercfg -h off
>> DISM /Online /Cleanup-Image
/StartComponentCleanup
>> cleanmgr
>> clean

Deployment Image Servicing and Management tool
Version: 10.0.19041.3636

Image Version: 10.0.19045.6456

```
[=====          10.0%      ]
```

Думал на протяжении 39s >

Окей, вижу что ты сделал — и главное: **ты ничего не “снёс” командой clean**, не переживай 😊

Что сейчас происходит по твоему логу

1. powercfg -h off

Выключил гибернацию и удалил hiberfil.sys → обычно **освобождает несколько ГБ** на C:. Это норм и безопасно.

2. DISM /Online /Cleanup-Image /StartComponentCleanup

Сейчас у тебя идёт именно это (процент 10%). DISM чистит компонентное хранилище Windows (WinSxS), часто **реально отъедает 1–5 ГБ**, иногда больше. И да, прогресс у DISM может “подвисать” на одном проценте — это типичное поведение.

3. cleanmgr

Запустит “Очистку диска” (обычно открывается окно). Там лучше нажать “**Очистить системные файлы**” и отметить Windows Update Cleanup / временные файлы и т.д.

4. clean

В **обычном PowerShell** это почти всегда **просто ошибка “команда не найдена”**.

Опасно clean только **внутри diskpart**, но тогда приглашение было бы DISKPART> (а у тебя PS C:\...>), так что **это не тот случай**.

Почему у тебя перед командами было >>

>> — это не “выполняю следующую команду”, а **режим продолжения ввода** (PowerShell думает, что команда не закончена: где-то открыта скобка/кавычка/ {} и т.п.).

Если после завершения DISM снова увидишь >> и не понимаешь почему — жми **Ctrl+C**, вернёт нормальный PS ...> .

Чтобы на 100% убедиться, что clean — не какая-то “левая” команда

После того как DISM закончит, выполнни:

powershell

```
Get-Command clean -All
Get-Alias clean -ErrorAction SilentlyContinue
```

Копировать код

Если ничего не найдёт — `clean` у тебя **не существует** (и значит не может “стереть диск”).

Если хочешь — кинь, сколько места стало на C: после `powercfg` и `DISM`, и я подскажу **самые жирные** ещё 2–3 безопасные чистки, чтобы добить твои **25–30 ГБ свободного** (у тебя сейчас прям “Windows на диете из воздуха”: 6.78 ГБ).

...

было 16 гб свободно
сейчас стало 29
что за

Это как раз то, ради чего ты всё и делал 😊

Было 16 ГБ, стало 29 ГБ — значит ты **освободил ~13 ГБ**. Нормально, даже жирно.

Откуда взялись эти гигабайты:

- `powercfg -h off` удалил `hiberfil.sys` (файл гибернации). Он обычно **3–8+ ГБ** (зависит от RAM и настроек).
- **DISM StartComponentCleanup** почистил **WinSxS / хвосты обновлений** — **ещё пару гигов**, иногда больше.
- Если параллельно `cleanmgr` что-то почистил — тоже мог добавить.

Так что “что за” = **всё по плану**, винда просто перестала держать у себя мусор “на всякий случай”.

Если хочешь добить **ещё свободного места** (без риска снести систему), вот **самые безопасные “следующие удары”**:

1. Папка загрузок обновлений (иногда раздутая):

powershell

```
net stop wuauserv
net stop bits
Remove-Item -Recurse -Force C:\Windows\SoftwareDistribution\Download\*
net start bits
net start wuauserv
```

Копировать код

2. Корзина и временные:

powershell

```
Clear-RecycleBin -Force
Remove-Item -Recurse -Force $env:TEMP\*
```

Копировать код

3. Проверить что жрёт С:

Самый быстрый вариант — открой **Параметры** → **Система** → **Память (Storage)** и там видно “кто жирный”.

Хочешь — скинь вывод твоего скрипта (или просто топ-10 самых больших папок с C:), и я скажу **что можно чистить, что нельзя, и что лучше перенести на D:**.



>> clean

Deployment Image Servicing and Management tool
Version: 10.0.19041.3636

Image Version: 10.0.19045.6456

```
[===== 10.0% ]
[=====100.0%=====
=====]
```

The operation completed successfully.

clean : Имя "clean" не распознано как имя командлета, функции, файла сценария или выполняемой программы. Проверьте правильность написания имени, а также наличие и правильность пути, после чего повторите попытку.

строка:4 знак:1

+ clean

+ ~~~~

+ CategoryInfo : ObjectNotFound: (clean:String) [],

```
CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException
```

```
PS C:\Users\Славиус\Desktop> net stop wuauserv
>> net stop bits
>> Remove-Item -Recurse -Force
C:\Windows\SoftwareDistribution\Download\*
>> net start bits
>> net start wuauserv
>> Clear-RecycleBin -Force
>> Remove-Item -Recurse -Force $env:TEMP\*
>>
```

Думал на протяжении 37s >

Всё отлично: **DISM отработал успешно**, а `clean` в обычном PowerShell — **правильно ругнулся** (“не распознано”). Значит ты **ничего не стирал** (и слава богу, диск сегодня без ампутаций 😊).

Почему у тебя снова >>

>> — это **режим продолжения ввода**. PowerShell думает, что ты *не закончил команду* (обычно где-то осталась **незакрытая кавычка**, скобка, `{}`, или строка закончилась на `| / ``).

И по твоему логу видно важное: после блока с `net stop...` у тебя **нет возврата к PS** `C:\...>`, значит **возможно эти команды вообще ещё не выполнились** — они просто “внутри незаконченной конструкции”.

Что сделать сейчас:

1. Нажми **Ctrl+C** — это сбросит режим `>>` и вернёт нормальный промпт `PS ...>`.
2. Запусти команды **по одной** (лучше в PowerShell от админа).

Правильный безопасный набор (и чтобы не падало на занятых файлах)

Открой PowerShell от **администратора**, потом:

powershell

```
net stop wuauserv
net stop bits
Remove-Item -Recurse -Force "$env:windir\SoftwareDistribution\Download\*" -ErrorAction SilentlyContinue
net start bits
net start wuauserv

Clear-RecycleBin -Force -ErrorAction SilentlyContinue
Get-ChildItem $env:TEMP -Force -ErrorAction SilentlyContinue | Remove-Item -Recurse
```

Копировать код

Если не админ — на net stop часто будет "Access denied / System error 5".

Проверить результат (сколько стало свободно на C)

powershell

 $[math]::Round((Get-PSDrive C).Free/1GB,2)$

Копировать код

Если хочешь — скинь, что покажет эта команда + что в "Параметры → Система → Память" самое жирное, и я подскажу, где ещё можно безопасно откусить ГБ (а где лучше вообще не трогать).

...

```
PS C:\Users\Славиус\Desktop> net stop wuauserv
The Центр обновления Windows service is not started.
```

More help is available by typing NET HELPMSG 3521.

```
PS C:\Users\Славиус\Desktop> net stop bits
>> ^C
PS C:\Users\Славиус\Desktop> Remove-Item -Recurse -
    Force "$env:windir\SoftwareDistribution\Download\*" -
    ErrorAction SilentlyContinue
>> net start bits
>> net start wuauserv
>> Clear-RecycleBin -Force -ErrorAction SilentlyContinue
>> Get-ChildItem $env:TEMP -Force -ErrorAction
    SilentlyContinue | Remove-Item -Recurse -Force -
    ErrorAction SilentlyContinue
```

>>

The Фоновая интеллектуальная служба передачи (BITS) service is starting.

The Фоновая интеллектуальная служба передачи (BITS) service was started successfully.

The Центр обновления Windows service is starting.

The Центр обновления Windows service was started successfully.

```
PS C:\Users\Славиус\Desktop> net stop wuauserv
```

>> net stop bits

>> Remove-Item -Recurse -Force

"\$env:windir\SoftwareDistribution\Download*" -

ErrorAction SilentlyContinue

>> net start bits

>> net start wuauserv

>>

>> Clear-RecycleBin -Force -ErrorAction SilentlyContinue

>> Get-ChildItem \$env:TEMP -Force -ErrorAction

SilentlyContinue | Remove-Item -Recurse -Force -

ErrorAction SilentlyContinue

>>

The Центр обновления Windows service is stopping.

The Центр обновления Windows service was stopped successfully.

The Фоновая интеллектуальная служба передачи (BITS) service is stopping..

The Фоновая интеллектуальная служба передачи (BITS) service was stopped successfully.

The Фоновая интеллектуальная служба передачи (BITS) service is starting.

The Фоновая интеллектуальная служба передачи (BITS) service was started successfully.

The Центр обновления Windows service is starting.

The Центр обновления Windows service was started successfully.

```
PS C:\Users\Славиус\Desktop>
```

```
PS C:\Users\Славиус\Desktop>
```

```
PS C:\Users\Славиус\Desktop>
```

```
PS C:\Users\Славиус\Desktop>
```

Отлично, ты всё сделал правильно. По логам видно:

- wuauserv сначала **не был запущен** — нормально.
- Потом ты корректно **остановил BITS и Центр обновления**, почистил SoftwareDistribution\Download , и **запустил службы обратно** — всё ок.
- Второй прогон тоже прошёл штатно.

А >> в начале у net stop bits — это был тот самый “режим продолжения”, ты его правильно **сбил Ctrl+C**, после чего всё поехало нормально.

Теперь давай зафиксируем результат — сколько свободно на C:

powershell

[math]::Round((Get-PSDrive C).Free/1GB,2)

 Копировать код

Если хочешь ещё +5–15 ГБ без боли, вот топовые и относительно безопасные шаги (по убыванию “выхлопа”):

1. Очистка “системных файлов” через cleanmgr

Нажми Win+R → cleanmgr → выбери C → “**Очистить системные файлы**” → отмечь:

- *Очистка обновлений Windows*
- *Временные файлы*
- *Файлы журнала обновления*
- *Корзина* (если не пустая)

(Старые установки/“Windows.old” — только если уверен, что откат не нужен.)

2. Проверить, не вырос ли pagefile.sys (файл подкачки)

Он может быть 4–16+ ГБ. Посмотреть:

powershell

dir C:\pagefile.sys -Force

 Копировать код

Если огромный — можно перенести/уменьшить, но это уже настройка, делать аккуратно.

3. Спящий режим уже выключен — красавчик (это был главный жирный кусок)

4. Самое частое: мусор в Downloads / рабочем столе / кэши браузеров

Но тут надо смотреть “кто жирный”.

Скинь вывод команды про свободное место — и если хочешь, дам “план добивания” до, скажем, 40–50 ГБ (чтобы Windows перестала жить в режиме “я дышу, значит я занимаю место”).



```
yContinue  
>> net start bits  
>> net start wuauserv  
>> Clear-RecycleBin -Force -ErrorAction SilentlyContinue  
>> Get-ChildItem $env:TEMP -Force -ErrorAction  
SilentlyContinue | Remove-Item -Recurse -Force -  
ErrorAction SilentlyContinue  
>>
```

The Фоновая интеллектуальная служба передачи (BITS) service is starting.

The Фоновая интеллектуальная служба передачи (BITS) service was started successfully.

The Центр обновления Windows service is starting.

The Центр обновления Windows service was started successfully.

```
PS C:\Users\Славиус\Desktop> net stop wuauserv  
>> net stop bits  
>> Remove-Item -Recurse -Force  
"$env:windir\SoftwareDistribution\Download\*" -  
ErrorAction SilentlyContinue  
>> net start bits  
>> net start wuauserv  
>>  
>> Clear-RecycleBin -Force -ErrorAction SilentlyContinue  
>> Get-ChildItem $env:TEMP -Force -ErrorAction  
SilentlyContinue | Remove-Item -Recurse -Force -  
ErrorAction SilentlyContinue  
>>  
The Центр обновления Windows service is stopping.
```

The Центр обновления Windows service was stopped successfully.

The Фоновая интеллектуальная служба передачи (BITS) service is stopping..

The Фоновая интеллектуальная служба передачи (BITS) service was stopped successfully.

The Фоновая интеллектуальная служба передачи (BITS) service is starting.

The Фоновая интеллектуальная служба передачи (BITS) service was started successfully.

The Центр обновления Windows service is starting.

The Центр обновления Windows service was started successfully.

```
PS C:\Users\Славиус\Desktop>
PS C:\Users\Славиус\Desktop>
PS C:\Users\Славиус\Desktop>
PS C:\Users\Славиус\Desktop> [math]::Round((Get-PSDrive C).Free/1GB,2)
>>
30.69
PS C:\Users\Славиус\Desktop>
```

О, кайф. **30.69 ГБ свободно** — это уже “Windows дышит, а не хрипит” 😊

И да, теперь у тебя почти выполнена рекомендация “25–30 ГБ headroom”.

А что за `yContinue` в начале

Это кусок строки из параметра `SilentlyContinue`, который ты, похоже, **случайно вставил с середины** (или терминал при копипасте “съел” начало строки). Это не команда, просто мусор в вводе.

Почему иногда опять появляется >>

`>>` = PowerShell считает, что ты **не закончил ввод** (чаще всего из-за незакрытой кавычки).

Лечится ровно так, как ты уже сделал: **Ctrl+C**.

Чтобы меньше ловить это — запускай команды **по одной** или вставляй **цельным блоком**, начиная с первой буквы, без обрезанных хвостов.

Что дальше: добиваем С: до 40–50 ГБ (без суицида и diskpart)

Если хочешь ещё место, вот следующий “безопасный топ”:

1) Узнать, кто реально жрёт С:

В PowerShell:

powershell

```
Get-ChildItem C:\ -Force -ErrorAction SilentlyContinue |  
Select-Object Name, @{n="GB";e={[math]::Round($_.Length/1GB,2)}} |  
Sort-Object GB -Descending | Select-Object -First 20
```

(Это грубо, но даст подсказку.)

Ещё лучше — встроенное:

Параметры → Система → Память → Временные файлы

Там часто прячутся “Очистка обновлений Windows”, Delivery Optimization, временные установщики.

2) Проверь “тяжёлых китов” в корне С:

powershell

```
dir C:\ -Force | sort Length -Descending | select -First 20
```

Смотри на:

- pagefile.sys (файл подкачки)
- Windows.old (если есть)
- C:\Windows\SoftwareDistribution (мы чистили Download, но папка может быть жирной)
- C:\Users\<ты>\AppData\Local\Temp (мы чистили, но иногда не всё удаляется)

3) Файл подкачки (если он огромный)

Посмотреть размер:

powershell

```
dir C:\pagefile.sys -Force
```

Если там 10–20+ ГБ, можно перенести на D: или ограничить, но скажу честно: это уже настройка, я дам конкретные шаги только после того как ты покажешь размер pagefile.sys и сколько у тебя RAM (у тебя ~12 ГБ).

Если хочешь, скинь вывод двух команд:

powershell

```
dir C:\ -Force | sort Length -Descending | select -First 15
dir C:\pagefile.sys -Force
```

 Копировать код

и я скажу: что можно резать/переносить, а к чему руки не тянуть.



Попробуйте новую кроссплатформенную оболочку
PowerShell (<https://aka.ms/pscore6>)

```
PS C:\Windows\system32> Get-ChildItem C:\ -Force -
ErrorAction SilentlyContinue |
>> Select-Object Name, @{n="GB";e=
{[math]::Round($_.Length/1GB,2)}} |
>> Sort-Object GB -Descending | Select-Object -First 20
>>
```

Name	GB
---	--
pagefile.sys	7.16
swapfile.sys	0.02
System Volume Information	0
Users	0
ProgramData	0
Recovery	0
DumpStack.log.tmp	0
\$Recycle.Bin	0
Windows	0
bootTel.dat	0
inetpub	0
MSOCache	0
\$WinREAgent	0

Documents and Settings	0
OneDriveTemp	0
Program Files	0
Program Files (x86)	0
PerfLogs	0
Program Box	0

PS C:\Windows\system32>

Каталог: C:\

Mode	LastWriteTime	Length	Name
---	-----	-----	-----
-a-hs-	12/27/2025 2:47 AM	7685099520	pagefile.sys
-a-hs-	12/27/2025 2:47 AM	16777216	swapfile.sys
-a-hs-	12/27/2025 2:47 AM	8192	DumpStack.log.tmp
--hs-	5/13/2025 10:44 PM	112	bootTel.dat
d--hs-	12/26/2025 11:23 PM		System Volume Information
d--hs-	1/3/2025 2:48 AM		Recovery
d--h--	12/27/2025 1:51 PM		ProgramData
d--hs-	1/3/2025 2:48 AM		\$Recycle.Bin
d----	10/25/2025 11:34 PM		Windows
d-r---	8/10/2025 11:50 PM		Users
d-----	4/13/2025 7:08 PM		inetpub
d-rh--	3/25/2025 2:25 PM		MSOCache
d--h--	6/11/2025 8:48 PM		\$WinREAgent
d--hsl	3/25/2023 10:12 PM		Documents and Settings
d--h--	1/4/2025 8:32 PM		OneDriveTemp
d-r---	9/9/2025 9:50 PM		Program Files
d-r---	12/27/2025 2:10 PM		Program Files (x86)
d-----	12/7/2019 12:14 PM		PerfLogs
d-----	3/25/2025 2:23 PM		Program Box

PS C:\Windows\system32>

Думал на протяжении 37s >

Всё норм — ты просто поймал “ложную диагностику” 😊

Почему у папок в твоём списке везде 0 GB

Потому что ты смотришь поле `Length`, а у папок оно **0** (размер есть только у файлов). Поэтому PowerShell честно показывает “папка = 0”, хотя внутри там может быть пол-винды.

Единственное, что реально видно из твоего вывода — это файлы в корне:

- `pagefile.sys` = 7.16 GB — файл подкачки, **нормально**.
 - `swapfile.sys` — мелочь.
-

Что делать дальше (по делу)

1) Посчитать, какие папки реально самые жирные (нормальная команда)

Сначала лучше по пользовательскому профилю (быстрее и полезнее):

powershell

```
$root = $env:USERPROFILE
Get-ChildItem $root -Directory -Force -ErrorAction SilentlyContinue |
ForEach-Object {
    $sum = (Get-ChildItem $_.FullName -File -Recurse -Force -ErrorAction Silently
Measure-Object Length -Sum).Sum
    [pscustomobject]@{ Path = $_.FullName; GB = [math]::Round($sum/1GB,2) }
} | Sort-Object GB -Descending | Select-Object -First 20
```

Копировать код

Если надо именно корень C:\ (будет дольше и местами “Access denied” — это ок):

powershell

```
$root = "C:\"
Get-ChildItem $root -Directory -Force -ErrorAction SilentlyContinue |
```

```
ForEach-Object {
    $sum = (Get-ChildItem $_.FullName -File -Recurse -Force -ErrorAction Silently
Measure-Object Length -Sum).Sum
    [pscustomobject]@{ Path = $_.FullName; GB = [math]::Round($sum/1GB,2) }
} | Sort-Object GB -Descending | Select-Object -First 20
```

Copировать код

2) Про pagefile.sys 7.16 GB — это “можно выиграть место, но аккуратно”

Удалять его нельзя — он управляет системой. Но если прям хочешь **быстро +5–7 ГБ на С:** можно **перенести/уменьшить** файл подкачки.

Самый безопасный вариант:

- оставить **небольшой pagefile на С:** (например 1–2 ГБ) — чтобы Windows могла писать дампы
- основной pagefile сделать на **D:** (у тебя там море места)

Как сделать (GUI, надёжнее всего):

1. Win+R → sysdm.cpl
2. вкладка Дополнительно → Быстродействие → Параметры
3. вкладка Дополнительно → Виртуальная память → Изменить
4. Снять галочку “Автоматически...”
5. С: → “Особый размер” (например 1024–2048 MB) → Задать
6. D: → “Системный размер” (или особый 4096–8192 MB) → Задать
7. OK → перезагрузка

Если сомневаешься — оставь как есть. **30.69 ГБ свободно уже отлично.**

3) Что ещё обычно жрёт место (и куда смотреть)

Раз уж С ожила, следующие “киты” обычно такие:

- C:\Users\...\AppData\Local (кэши, Telegram/Discord, браузеры)
- Downloads
- C:\Windows\Installer (часто большой, **вручную не трогать**)
- точки восстановления (System Volume Information — размер скрыт)

Проверка точек восстановления:

powershell

```
vssadmin list shadowstorage
```

Copilot Копировать код

Если хочешь, пришли результат команды **по профилю** (пункт 1) — я скажу, где самые жирные папки, что можно снести/почистить, а что лучше переносом на D решить (без игры в "удали System32 — и места много, и проблем тоже").

🔗 🤝 🤔 ⏚ ⏴ ...

Windows PowerShell

(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

Попробуйте новую кроссплатформенную оболочку
PowerShell (<https://aka.ms/pscore6>)

```
PS C:\Windows\system32> Get-ChildItem C:\ -Force -  
ErrorAction SilentlyContinue |  
    >> Select-Object Name, @{n="GB";e=  
    {[math]::Round($_.Length/1GB,2)}} |  
    >> Sort-Object GB -Descending | Select-Object -First 20  
    >>
```

Name	GB
pagefile.sys	7.16
swapfile.sys	0.02
System Volume Information	0
Users	0
ProgramData	0
Recovery	0
DumpStack.log.tmp	0
\$Recycle.Bin	0
Windows	0
bootTel.dat	0
inetpub	0
MSOCache	0
\$WinREAgent	0
Documents and Settings	0

OneDriveTemp	0
Program Files	0
Program Files (x86)	0
PerfLogs	0
Program Box	0

```
PS C:\Windows\system32> dir C:\ -Force | sort Length -
Descending | select -First 20
>>
```

Каталог: C:\

Mode	LastWriteTime	Length	Name
---	-----	-----	-----
-a-hs-	12/27/2025 2:47 AM	7685099520	
	pagefile.sys		
-a-hs-	12/27/2025 2:47 AM	16777216	
	swapfile.sys		
-a-hs-	12/27/2025 2:47 AM	8192	
	DumpStack.log.tmp		
--hs-	5/13/2025 10:44 PM	112	bootTel.dat
d--hs-	12/26/2025 11:23 PM		System Volume
	Information		
d--hs-	1/3/2025 2:48 AM		Recovery
d--h--	12/27/2025 1:51 PM		ProgramData
d--hs-	1/3/2025 2:48 AM		\$Recycle.Bin
d----	10/25/2025 11:34 PM		Windows
d-r---	8/10/2025 11:50 PM		Users
d-----	4/13/2025 7:08 PM		inetpub
d-rh--	3/25/2025 2:25 PM		MSOCache
d--h--	6/11/2025 8:48 PM		\$WinREAgent
d--hsl	3/25/2023 10:12 PM		Documents and
	Settings		
d--h--	1/4/2025 8:32 PM		OneDriveTemp
d-r---	9/9/2025 9:50 PM		Program Files
d-r---	12/27/2025 2:10 PM		Program Files
	(x86)		
d-----	12/7/2019 12:14 PM		PerfLogs
d-----	3/25/2025 2:23 PM		Program Box

```
PS C:\Windows\system32> $root = $env:USERPROFILE
>> Get-ChildItem $root -Directory -Force -ErrorAction
SilentlyContinue |
>> ForEach-Object {
>>   $sum = (Get-ChildItem $_.FullName -File -Recurse -
Force -ErrorAction SilentlyContinue |
>>     Measure-Object Length -Sum).Sum
>>   [pscustomobject]@{ Path = $_.FullName; GB =
[math]::Round($sum/1GB,2) }
>> } | Sort-Object GB -Descending | Select-Object -First
20
>>
```

Path	GB
---	--
C:\Users\Славиус\AppData	17.66
C:\Users\Славиус\OneDrive	7.69
C:\Users\Славиус\Microsoft	0.81
C:\Users\Славиус\.windsurf	0.67
C:\Users\Славиус\.jdks	0.65
C:\Users\Славиус\.rustup	0.55
C:\Users\Славиус\.codeium	0.52
C:\Users\Славиус\.git	0.39
C:\Users\Славиус\IdeaProjects	0.28
C:\Users\Славиус\.cache	0.09
C:\Users\Славиус\.cursor	0.07
C:\Users\Славиус\Desktop	0.06
C:\Users\Славиус\Fooocus	0.05
C:\Users\Славиус\Recent	0
C:\Users\Славиус\PrintHood	0
C:\Users\Славиус\Music	0
C:\Users\Славиус\NetHood	0
C:\Users\Славиус\.codex	0
C:\Users\Славиус\Saved Games	0
C:\Users\Славиус\главное меню	0

```
PS C:\Windows\system32> $root = "C:\"
>> Get-ChildItem $root -Directory -Force -ErrorAction
SilentlyContinue |
>> ForEach-Object {
>>   $sum = (Get-ChildItem $_.FullName -File -Recurse -
```

```

Force -ErrorAction SilentlyContinue |
>>      Measure-Object Length -Sum).Sum
>>  [pscustomobject]@{ Path = $_.FullName; GB =
[math]::Round($sum/1GB,2) }
>> } | Sort-Object GB -Descending | Select-Object -First
20
>>
Get-ChildItem : Не удается найти указанный файл
строка:4 знак:13
+ $sum = (Get-ChildItem $_.FullName -File -Recurse -
Force -ErrorAct ...
+
~~~~~
~~~~~
+ CategoryInfo          : NotSpecified: () [Get-ChildItem],
Win32Exception
+ FullyQualifiedErrorId :
System.ComponentModel.Win32Exception,Microsoft.PowerShell.Commands.GetChildItemCommand

```

Path	GB
---	--
C:\Windows	19.81
C:\Program Files	15.34
C:\ProgramData	6.43
C:\Program Files (x86)	2.97
C:\Program Box	0.8
C:\MSOCache	0.74
C:\System Volume Information	0
C:\Recovery	0
C:\Users	0
C:\\$Recycle.Bin	0
C:\\$WinREAgent	0
C:\PerfLogs	0
C:\OneDriveTemp	0
C:\Documents and Settings	0
C:\inetpub	0

```

PS C:\Windows\system32> vssadmin list shadowstorage
>>
vssadmin 1.1 - Программа командной строки для

```

администрирования службы теневого копирования
томов
(C) Корпорация Майкрософт (Microsoft Corporation),
2001-2013.

Отсутствуют элементы, соответствующие запросу.
PS C:\Windows\system32>

Вот теперь картина ясная — и она такая: **твой C: забит не “виндой”, а твоими кэшами/инструментами и софтом**. Хорошая новость: тут можно отжать ещё место без “переустановки Windows”.

Что реально самое жирное у тебя

По профилю пользователя:

- **AppData** — 17.66 ГБ (вот тут главный мусор и кэши)
- **OneDrive** — 7.69 ГБ (возможно можно сделать “только онлайн”)
- дальше мелочи: .windsurf, .jdks, .rustup, .codeium, .git и т.п.

По корню C:

- **Windows** — 19.81 ГБ (это норм, руками не лезем)
- **Program Files** — 15.34 ГБ
- **ProgramData** — 6.43 ГБ (часто тут кэши установщиков/антивирусов/обновлений)

Точки восстановления у тебя **не жрут** (vssadmin пусто) — ок.

План “быстро + много” (безопасно)

1) Распотрошим AppData (самый жирный кит)

Дай топ-20 самых больших папок внутри AppData:

powershell

```
$root = "$env:USERPROFILE\AppData"
Get-ChildItem $root -Directory -Force -ErrorAction SilentlyContinue |
ForEach-Object {
    $sum = (Get-ChildItem $_.FullName -File -Recurse -Force -ErrorAction Silently
Measure-Object Length -Sum).Sum
```

```
[pscobject]@{ Path = $_.FullName; GB = [math]::Round($sum/1GB, 2) } | Sort-Object GB -Descending | Select-Object -First 20
```

Копировать код

С вероятностью 80% там вылезет что-то типа:

- AppData\Local\Google\Chrome\User Data\Default\Cache
- AppData\Roaming\Telegram Desktop\tdata (если ты с телегой)
- кэши VS Code / JetBrains / npm / pip / cargo
- кэши драйверов/GeForce Experience

Руками “удалить всю AppData” нельзя, но точечно чистить кэши — можно и нужно.

2) OneDrive 7.69 ГБ — можно освободить почти всё без удаления файлов

Если тебе не нужно держать эти файлы физически на диске:

- ПКМ по папкам в OneDrive → “Освободить место” (Files On-Demand)
Это оставит “ярлыки” в облако, а на диске будет почти 0.
-

3) ProgramData 6.43 ГБ — часто там мусор установщиков

Нужно понять кто именно. Топ папок:

powershell

```
$root = "C:\ProgramData"
Get-ChildItem $root -Directory -Force -ErrorAction SilentlyContinue |
ForEach-Object {
    $sum = (Get-ChildItem $_.FullName -File -Recurse -Force -ErrorAction Silently
Measure-Object Length -Sum).Sum
    [pscobject]@{ Path = $_.FullName; GB = [math]::Round($sum/1GB, 2) }
} | Sort-Object GB -Descending | Select-Object -First 20
```

Копировать код

Про pagefile.sys 7.16 ГБ

Это нормально для 12 ГБ RAM. **Можно** выиграть 3–6 ГБ, но это уже “тюнинг”, а не уборка. Я бы трогал **только если тебе критично**.

Почему у тебя была ошибка “Не удается найти указанный файл”

Это из-за того, что по C:\ ты рекурсивно лезешь в системные каталоги — там бывают:

- “битые” ссылки/переадресации,
- папки с особыми правами,
- временные точки монтирования.

Это ок, результат у тебя всё равно получился.

Дальше делаем по-взрослому

Скинь вывод двух команд (AppData и ProgramData). По ним я скажу:

- что можно удалять сразу,
- что чистить через настройки программы,
- что лучше перенести на D: (JetBrains SDK/Android SDK/кэши билдов),
- и где есть риск “сломать” программы.



Windows PowerShell

(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

Попробуйте новую кроссплатформенную оболочку
PowerShell (<https://aka.ms/pscore6>)

```
PS C:\Windows\system32> Get-ChildItem C:\ -Force -  
ErrorAction SilentlyContinue |  
>> Select-Object Name, @{n="GB";e=  
{[math]::Round($_.Length/1GB,2)}} |  
>> Sort-Object GB -Descending | Select-Object -First 20  
>>
```

Name	GB
pagefile.sys	7.16
swapfile.sys	0.02
System Volume Information	0
Users	0
ProgramData	0

Recovery	0
DumpStack.log.tmp	0
\$Recycle.Bin	0
Windows	0
bootTel.dat	0
inetpub	0
MSOCache	0
\$WinREAgent	0
Documents and Settings	0
OneDriveTemp	0
Program Files	0
Program Files (x86)	0
PerfLogs	0
Program Box	0

```
PS C:\Windows\system32> dir C:\ -Force | sort Length -Descending | select -First 20
>>
```

Каталог: C:\

Mode	LastWriteTime	Length	Name
---	-----	-----	-----
-a-hs-	12/27/2025 2:47 AM	7685099520	pagefile.sys
-a-hs-	12/27/2025 2:47 AM	16777216	swapfile.sys
-a-hs-	12/27/2025 2:47 AM	8192	DumpStack.log.tmp
---hs-	5/13/2025 10:44 PM	112	bootTel.dat
d--hs-	12/26/2025 11:23 PM		System Volume Information
d--hs-	1/3/2025 2:48 AM		Recovery
d--h--	12/27/2025 1:51 PM		ProgramData
d--hs-	1/3/2025 2:48 AM		\$Recycle.Bin
d-----	10/25/2025 11:34 PM		Windows
d-r---	8/10/2025 11:50 PM		Users
d-----	4/13/2025 7:08 PM		inetpub
d-rh--	3/25/2025 2:25 PM		MSOCache
d--h--	6/11/2025 8:48 PM		\$WinREAgent

d--hs1	3/25/2023 10:12 PM	Documents and Settings
d--h--	1/4/2025 8:32 PM	OneDriveTemp
d-r---	9/9/2025 9:50 PM	Program Files
d-r---	12/27/2025 2:10 PM	Program Files
(x86)		
d-----	12/7/2019 12:14 PM	PerfLogs
d-----	3/25/2025 2:23 PM	Program Box

```
PS C:\Windows\system32> $root = $env:USERPROFILE
>> Get-ChildItem $root -Directory -Force -ErrorAction
SilentlyContinue |
>> ForEach-Object {
>>   $sum = (Get-ChildItem $_.FullName -File -Recurse -
Force -ErrorAction SilentlyContinue |
>>     Measure-Object Length -Sum).Sum
>>   [pscustomobject]@{ Path = $_.FullName; GB =
[math]::Round($sum/1GB,2) }
>> } | Sort-Object GB -Descending | Select-Object -First
20
>>
```

Path	GB
---	--
C:\Users\Славиус\AppData	17.66
C:\Users\Славиус\OneDrive	7.69
C:\Users\Славиус\Microsoft	0.81
C:\Users\Славиус\.windsurf	0.67
C:\Users\Славиус\.jdks	0.65
C:\Users\Славиус\.rustup	0.55
C:\Users\Славиус\.codeium	0.52
C:\Users\Славиус\.git	0.39
C:\Users\Славиус\IdeaProjects	0.28
C:\Users\Славиус\.cache	0.09
C:\Users\Славиус\.cursor	0.07
C:\Users\Славиус\Desktop	0.06
C:\Users\Славиус\Fooocus	0.05
C:\Users\Славиус\Recent	0
C:\Users\Славиус\PrintHood	0
C:\Users\Славиус\Music	0
C:\Users\Славиус\NetHood	0
C:\Users\Славиус\codex	0

```
C:\Users\Славиус\Saved Games    0
C:\Users\Славиус\главное меню   0
```

```
PS C:\Windows\system32> $root = "C:\"
>> Get-ChildItem $root -Directory -Force -ErrorAction
SilentlyContinue |
>>  ForEach-Object {
>>      $sum = (Get-ChildItem $_.FullName -File -Recurse -
Force -ErrorAction SilentlyContinue |
>>          Measure-Object Length -Sum).Sum
>>      [pscustomobject]@{ Path = $_.FullName; GB =
[math]::Round($sum/1GB,2) }
>> } | Sort-Object GB -Descending | Select-Object -First
20
>>
Get-ChildItem : Не удается найти указанный файл
строка:4 знак:13
+      $sum = (Get-ChildItem $_.FullName -File -Recurse -
Force -ErrorAct ...
+
~~~~~
~~~~~
+ CategoryInfo          : NotSpecified: () [Get-ChildItem], Win32Exception
+ FullyQualifiedErrorId :
System.ComponentModel.Win32Exception,Microsoft.PowerShell.Commands.GetChildItemCommand
```

Path	GB
---	--
C:\Windows	19.81
C:\Program Files	15.34
C:\ProgramData	6.43
C:\Program Files (x86)	2.97
C:\Program Box	0.8
C:\MSOCache	0.74
C:\System Volume Information	0
C:\Recovery	0
C:\Users	0
C:\\$Recycle.Bin	0
C:\\$WinREAgent	0

```
C:\PerfLogs          0
C:\OneDriveTemp      0
C:\Documents and Settings   0
C:\inetpub           0
```

```
PS C:\Windows\system32> vssadmin list shadowstorage
>>
vssadmin 1.1 - Программа командной строки для
администрирования службы теневого копирования
томов
(С) Корпорация Майкрософт (Microsoft Corporation),
2001-2013.
```

Отсутствуют элементы, соответствующие запросу.
PS C:\Windows\system32>

Windows PowerShell
(С) Корпорация Майкрософт (Microsoft Corporation). Все
права защищены.

Попробуйте новую кроссплатформенную оболочку
PowerShell (<https://aka.ms/pscore6>)

```
PS C:\Windows\system32> cd $env:USERPROFILE\Desktop
>> python ./win11_readiness_report_v2.py --outdir .
>>
==== Windows 11 Readiness / System Inventory Report
(v2) ====
Generated: 2025-12-27T14:15:56
Host: DESKTOP-1OAT63D | User: Славиус | SystemDrive: C:
```

--- Windows ---
Product: Windows 10 Home
Version: 2009 | Build: 19045 | Edition: Core
Firmware (from ComputerInfo): 2
Manufacturer/Model: MSI MS-7817

--- CPU / RAM ---
CPU #1: Intel(R) Core(TM) i3-4170 CPU @ 3.70GHz | Cores:
2 | Threads: 4 | AddrWidth: 64

RAM: 11.95 GB

--- Firmware / Security ---

Boot mode: Uefi

Secure Boot: NO

TPM info: {'gettpm': {'TpmPresent': False, 'TpmReady': False, 'TpmEnabled': False, 'TpmActivated': False, 'TpmOwned': False, 'RestartPending': False, 'ManufacturerId': 0, 'ManufacturerIdTxt': None, 'ManufacturerVersion': None, 'ManufacturerVersionFull20': None, 'ManagedAuthLevel': 4, 'OwnerAuth': None, 'OwnerClearDisabled': True, 'AutoProvisioning': -1, 'LockedOut': False, 'LockoutHealTime': None, 'LockoutCount': None, 'LockoutMax': None, 'SelfTest': None}, 'win32tpm': []}, 'spec_versions': [], 'tpm2_supported': None, 'error': None}

--- Disks ---

Disk 0: P4-240 | 223.57 GB | Style=GPT | Bus=SATA | IsSystem=True IsBoot=True

Disk 1: SAMSUNG HD161GJ | 149.05 GB | Style=MBR | Bus=SATA | IsSystem=False IsBoot=False

--- Volumes ---

G: | FS=NTFS | Label= | Size=24.04 GB | Free=17.57 GB | Health=Healthy

E: | FS=NTFS | Label=Win 7 | Size=30.01 GB | Free=29.93 GB | Health=Healthy

F: | FS=NTFS | Label= | Size=95.0 GB | Free=49.87 GB | Health=Healthy

C: | FS=NTFS | Label= | Size=111.4 GB | Free=6.78 GB | Health=Healthy

D: | FS=NTFS | Label=Новый том | Size=111.4 GB | Free=106.38 GB | Health=Healthy

H: | FS=Unknown | Label= | Size=0.0 GB | Free=0.0 GB | Health=Healthy

==== Windows 11 Readiness Checks ===

Verdict: NOT READY (some requirements failed)

- [PASS] CPU 64-bit: 64-bit (required: Yes)
- [PASS] RAM: 11.95 GB (required: >= 4 GB)
- [PASS] System drive capacity: 111.4 GB (required: >= 64 GB (Windows 11 requirement))

- [FAIL] System drive free space: 6.78 GB free (required: >= 25 GB (recommended for upgrade headroom))
- [PASS] Boot mode: Uefi (required: UEFI)
- [FAIL] Secure Boot: NO (required: Enabled)
- [FAIL] TPM: Present=NO, Ready=NO, TPM2=UNKNOWN (required: TPM 2.0 present & ready)
- [PASS] System disk partition style: Disk 0: GPT (required: GPT (recommended/typical for UEFI + Win11))

Next steps (based on blockers):

- * Free up space on C: aim for 25–30 GB free before upgrade.
- * Enable Secure Boot in UEFI settings.
- * Enable TPM 2.0 in BIOS/UEFI (Intel PTT / AMD fTPM).

Saved:

```
- C:\Users\Славиус\Desktop\win11_readiness_v2_DESKTOP-1OAT63D_20251227_141624.txt -  
C:\Users\Славиус\Desktop\win11_readiness_v2_DESKTOP-1OAT63D_20251227_141624.json  
PS C:\Users\Славиус\Desktop>  
PS C:\Users\Славиус\Desktop>  
PS C:\Users\Славиус\Desktop>  
PS C:\Users\Славиус\Desktop> powercfg -h off  
>> DISM /Online /Cleanup-Image  
/StartComponentCleanup  
>> cleanmgr  
>> clean
```

Deployment Image Servicing and Management tool
Version: 10.0.19041.3636

Image Version: 10.0.19045.6456

```
[===== 10.0% ]  
[===== 100.0% =====]  
===== ]
```

The operation completed successfully.

clean : Имя "clean" не распознано как имя командлета, функции, файла сценария или выполняемой программы. Проверьте правильность написания имени, а также наличие и

```
правильность пути, после чего повторите попытку.  
строка:4 знак:1  
+ clean  
+ ~~~~~  
+ CategoryInfo          : ObjectNotFound: (clean:String) [],  
CommandNotFoundException +  
FullyQualifiedErrorId : CommandNotFoundException  
PS C:\Users\Славиус\Desktop> net stop wuauserv  
The Центр обновления Windows service is not started.
```

More help is available by typing NET HELPMSG 3521.

```
PS C:\Users\Славиус\Desktop> net stop bits  
>> ^C  
PS C:\Users\Славиус\Desktop> Remove-Item -Recurse -  
Force "$env:windir\SoftwareDistribution\Download\*" -  
ErrorAction SilentlyContinue  
>> net start bits  
>> net start wuauserv  
>> Clear-RecycleBin -Force -ErrorAction SilentlyContinue  
>> Get-ChildItem $env:TEMP -Force -ErrorAction  
SilentlyContinue | Remove-Item -Recurse -Force -  
ErrorAction SilentlyContinue  
>>  
The Фоновая интеллектуальная служба передачи (BITS)  
service is starting.  
The Фоновая интеллектуальная служба передачи (BITS)  
service was started successfully.
```

The Центр обновления Windows service is starting.
The Центр обновления Windows service was started
successfully.

```
PS C:\Users\Славиус\Desktop> net stop wuauserv  
>> net stop bits  
>> Remove-Item -Recurse -Force  
"$env:windir\SoftwareDistribution\Download\*" -  
ErrorAction SilentlyContinue  
>> net start bits  
>> net start wuauserv  
>>  
>> Clear-RecycleBin -Force -ErrorAction SilentlyContinue  
>> Get-ChildItem $env:TEMP -Force -ErrorAction
```

```
SilentlyContinue | Remove-Item -Recurse -Force -  
ErrorAction SilentlyContinue
```

>>

The Центр обновления Windows service is stopping.
The Центр обновления Windows service was stopped
successfully.

The Фоновая интеллектуальная служба передачи (BITS)
service is stopping..

The Фоновая интеллектуальная служба передачи (BITS)
service was stopped successfully.

The Фоновая интеллектуальная служба передачи (BITS)
service is starting.

The Фоновая интеллектуальная служба передачи (BITS)
service was started successfully.

The Центр обновления Windows service is starting.

The Центр обновления Windows service was started
successfully.

```
PS C:\Users\Славиус\Desktop>
```

```
PS C:\Users\Славиус\Desktop>
```

```
PS C:\Users\Славиус\Desktop>
```

```
PS C:\Users\Славиус\Desktop> [math]::Round((Get-  
PSDrive C).Free/1GB,2)
```

>>

30.69

```
PS C:\Users\Славиус\Desktop>
```

```
PS C:\Windows\system32> vssadmin list shadowstorage
```

>>

vssadmin 1.1 - Программа командной строки для
администрирования службы теневого копирования
томов

(C) Корпорация Майкрософт (Microsoft Corporation),
2001-2013.

Отсутствуют элементы, соответствующие запросу.

```
PS C:\Windows\system32> $root =  
"$env:USERPROFILE\AppData"  
>> Get-ChildItem $root -Directory -Force -ErrorAction  
SilentlyContinue |
```

```
>> ForEach-Object {
>>     $sum = (Get-ChildItem $_.FullName -File -Recurse -Force -ErrorAction SilentlyContinue |
>>             Measure-Object Length -Sum).Sum
>>     [pscustomobject]@{ Path = $_.FullName; GB =
>>         [math]::Round($sum/1GB,2) }
>> } | Sort-Object GB -Descending | Select-Object -First 20
>>
```

Path	GB
---	--
C:\Users\Славиус\AppData\Local	14.5
C:\Users\Славиус\AppData\Roaming	2.97
C:\Users\Славиус\AppData\LocalLow	0.2

```
PS C:\Windows\system32> $root = "C:\ProgramData"
>> Get-ChildItem $root -Directory -Force -ErrorAction SilentlyContinue |
>>     ForEach-Object {
>>         $sum = (Get-ChildItem $_.FullName -File -Recurse -Force -ErrorAction SilentlyContinue |
>>             Measure-Object Length -Sum).Sum
>>         [pscustomobject]@{ Path = $_.FullName; GB =
>>             [math]::Round($sum/1GB,2) }
>> } | Sort-Object GB -Descending | Select-Object -First 20
>>
```

Path	GB
---	--
C:\ProgramData\NVIDIA Corporation	3.59
C:\ProgramData\Microsoft	1.89
C:\ProgramData\Package Cache	0.64
C:\ProgramData\NVIDIA	0.21
C:\ProgramData\Outbyte	0.06
C:\ProgramData\USOShared	0.01
C:\ProgramData\USOPrivate	0.01
C:\ProgramData\TP-LINK	0
C:\ProgramData\WindowsHolographicDevices	0
C:\ProgramData\Yandex	0
C:\ProgramData\Документы	0

C:\ProgramData\Рабочий стол	0
C:\ProgramData\Шаблоны	0
C:\ProgramData\главное меню	0
C:\ProgramData\ssh	0
C:\ProgramData\Microsoft Help	0
C:\ProgramData\Microsoft OneDrive	0
C:\ProgramData\Application Data	0
C:\ProgramData\FLEXnet	0
C:\ProgramData\Packages	0

PS C:\Windows\system32>

Думал на протяжении 44s >

Найс, ты уже сделал **самое полезное**: С: теперь **30.69 ГБ свободно**. Винда перестала жить «на ингаляторе». 😊

А твой старый отчёт про **6.78 ГБ** — просто был сгенерен **до** чистки. Если сейчас перезапустишь `win11_readiness_report_v2.py`, там покажет новые цифры.

Теперь по твоим цифрам видно, **кто реально жрёт место**:

Главные пожиратели

1) AppData = 17.66 ГБ

Из них:

- `AppData\Local` = **14.5 ГБ** (обычно кэши браузеров/мессенджеров/IDE/апдейтеров)
- `AppData\Roaming` = **2.97 ГБ** (Telegram/Discord/плагины/профили)

2) OneDrive = 7.69 ГБ

Это почти гарантированно можно “скать” без удаления файлов — сделать их **только онлайн**.

3) C:\ProgramData\NVIDIA Corporation = 3.59 ГБ

Часто это **кэш/загрузчик драйверов** — можно чистить выборочно.

Что делать дальше (быстро и безопасно)

Шаг 1 — покажи “топ виновников” внутри `AppData\Local` и `Roaming`

Выполните по очереди:

`powershell`

```
# ТОП папок в Local (кто жрёт 14.5 ГБ)
$root = "$env:USERPROFILE\AppData\Local"
Get-ChildItem $root -Directory -Force -ErrorAction SilentlyContinue |
ForEach-Object {
    $sum = (Get-ChildItem $_.FullName -File -Recurse -Force -ErrorAction Silently
Measure-Object Length -Sum).Sum
```

↗ Копировать код

```
[pscobject]@{ Path = $_.FullName; GB = [math]::Round($sum/1GB,2) }
} | Sort-Object GB -Descending | Select-Object -First 25
```

powershell

```
# ТОП папок в Roaming (кто жрёт 3 ГБ)
$root = "$env:USERPROFILE\AppData\Roaming"
Get-ChildItem $root -Directory -Force -ErrorAction SilentlyContinue |
ForEach-Object {
    $sum = (Get-ChildItem $_.FullName -File -Recurse -Force -ErrorAction Silently
Measure-Object Length -Sum).Sum
    [pscobject]@{ Path = $_.FullName; GB = [math]::Round($sum/1GB,2) }
} | Sort-Object GB -Descending | Select-Object -First 25
```

Копировать код

Скинешь результаты — я скажу **точечно**, что можно чистить прямо сейчас, а что лучше не трогать.

Шаг 2 — OneDrive: быстрый “бесплатный” прирост места

Если ок хранить в облаке:

- в проводнике → OneDrive → ПКМ по жирным папкам → “**Освободить место**” (файлы останутся, но станут “только онлайн”).

Шаг 3 — NVIDIA: чистим кэш драйверов (обычно безопасно)

Проверь, есть ли эти папки, и почисти **их содержимое** (не всю “NVIDIA Corporation” целиком):

powershell

```
rm -Recurse -Force "C:\ProgramData\NVIDIA Corporation\Downloader\*" -ErrorAction SilentlyContinue
rm -Recurse -Force "C:\ProgramData\NVIDIA Corporation\NV_Cache\*" -ErrorAction SilentlyContinue
```

Копировать код

Если папок нет — команда просто молча ничего не сделает (это ок).

Важное про Windows 11 (чтобы не было иллюзий)

У тебя в отчёте:

- Secure Boot = NO
- TPM отсутствует

И ещё: i3-4170 (4-е поколение) официально не поддерживается Windows 11.

То есть даже если место на С уже норм, "официальная готовность" всё равно будет падать на TPM/CPU (есть обходы, но это отдельная история).

Кидай сюда вывод топ-25 по AppData\Local и AppData\Roaming — и я дам конкретный список "удалить/почистить/перенести", чтобы добить С: до 40–60 ГБ без риска сломать систему.



Windows PowerShell

(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

Попробуйте новую кроссплатформенную оболочку PowerShell (<https://aka.ms/pscore6>)

```
PS C:\Windows\system32> # Т папок в Local (кто жрёт
14.5 )
PS C:\Windows\system32> $root =
"$env:USERPROFILE\AppData\Local"
PS C:\Windows\system32> Get-ChildItem $root -Directory
-Force -ErrorAction SilentlyContinue |
>> ForEach-Object {
>>   $sum = (Get-ChildItem $_.FullName -File -Recurse -
Force -ErrorAction SilentlyContinue |
>>     Measure-Object Length -Sum).Sum
>>   [pscustomobject]@{ Path = $_.FullName; GB =
[math]::Round($sum/1GB,2) }
>> } | Sort-Object GB -Descending | Select-Object -First
25
```

Path	GB
---	--
C:\Users\Славиус\AppData\Local\Yandex	3.11
C:\Users\Славиус\AppData\Local\Programs	2.48

C:\Users\Славиус\AppData\Local\Microsoft	2.43
C:\Users\Славиус\AppData\Local\Packages	1.92
C:\Users\Славиус\AppData\Local\JetBrains	1.2
C:\Users\Славиус\AppData\Local\GitHubDesktop	
1.01	
C:\Users\Славиус\AppData\Local\SquirrelTemp	
0.65	
C:\Users\Славиус\AppData\Local\Opera Software	
0.45	
C:\Users\Славиус\AppData\Local\Steam	0.29
C:\Users\Славиус\AppData\Local\puccinialin	0.19
C:\Users\Славиус\AppData\Local\pip	0.18
C:\Users\Славиус\AppData\Local\pachca-updater	
0.15	
C:\Users\Славиус\AppData\Local\Temp	0.14
C:\Users\Славиус\AppData\Local\ms-playwright-go	
0.09	
C:\Users\Славиус\AppData\Local\CrashDumps	
0.07	
C:\Users\Славиус\AppData\Local\NVIDIA Corporation	
0.04	
C:\Users\Славиус\AppData\Local\Package Cache	
0.03	
C:\Users\Славиус\AppData\Local\Comms	0.03
C:\Users\Славиус\AppData\Local\CEF	0.02
C:\Users\Славиус\AppData\Local\pupa	0.01
C:\Users\Славиус\AppData\Local\PlaceholderTileLogoFold	
er 0	
C:\Users\Славиус\AppData\Local\Rockstar Games	
0	
C:\Users\Славиус\AppData\Local\PAYDAY 2	0
C:\Users\Славиус\AppData\Local\seaborn	0
C:\Users\Славиус\AppData\Local\TunnelBear	0

```
PS C:\Windows\system32>
PS C:\Windows\system32> # ТОП папок в Roaming (кто
жрёт 3 ГБ)
>> $root = "$env:USERPROFILE\AppData\Roaming"
>> Get-ChildItem $root -Directory -Force -ErrorAction
SilentlyContinue |
>> ForEach-Object {
>>     $sum = (Get-ChildItem $_.FullName -File -Recurse -

```

```
Force -ErrorAction SilentlyContinue |  

>>      Measure-Object Length -Sum).Sum  

>>  [pscustomobject]@{ Path = $_.FullName; GB =  

[math]::Round($sum/1GB,2) }  

>> } | Sort-Object GB -Descending | Select-Object -First  

25  

>>
```

Path	GB
---	--
C:\Users\Славиус\AppData\Roaming\JetBrains	1.24
C:\Users\Славиус\AppData\Roaming\Windsurf	0.58
C:\Users\Славиус\AppData\Roaming\Opera Software	0.56
C:\Users\Славиус\AppData\Roaming\TunnelBear	0.18
C:\Users\Славиус\AppData\Roaming\Microsoft	0.14
C:\Users\Славиус\AppData\Roaming\Python	0.09
C:\Users\Славиус\AppData\Roaming\Cursor	0.08
C:\Users\Славиус\AppData\Roaming\Pachca	0.03
C:\Users\Славиус\AppData\Roaming\utorrent	0.03
C:\Users\Славиус\AppData\Roaming\WinRAR	0.02
C:\Users\Славиус\AppData\Roaming\GitHub Desktop	
0.01	
C:\Users\Славиус\AppData\Roaming\dbeaverData	0.01
C:\Users\Славиус\AppData\Roaming\Yandex	0
C:\Users\Славиус\AppData\Roaming\VK	0
C:\Users\Славиус\AppData\Roaming\VK Messenger	0
C:\Users\Славиус\AppData\Roaming\jupyter	0
C:\Users\Славиус\AppData\Roaming\Electronic Arts	0
C:\Users\Славиус\AppData\Roaming\AnyDesk	0
C:\Users\Славиус\AppData\Roaming\postgresql	0
C:\Users\Славиус\AppData\Roaming\PolarBear	0
C:\Users\Славиус\AppData\Roaming\Adobe	0

PS C:\Windows\system32>

```
PS C:\Windows\system32> Remove-Item -Recurse -Force  

"C:\ProgramData\NVIDIA Corporation\Downloader\*" -  

ErrorAction SilentlyContinue  

>> Remove-Item -Recurse -Force  

"C:\ProgramData\NVIDIA Corporation\NV_Cache\*" -
```

```
ErrorAction SilentlyContinue
```

```
>>
```

```
PS C:\Windows\system32>
```

Думал на протяжении 56s >

О, вот теперь ты принёс **вкусное мясо**, а не "0 GB у папок" 😊

По факту у тебя С: жрут не "мистические системные демоны", а очень конкретные ребята:

- Local\Yandex — 3.11 ГБ (кэш/профили браузера)
- Local\Programs — 2.48 ГБ (проги, установленные "в профиль")
- Local\Microsoft — 2.43 ГБ (Edge/Teams/кэши/разное)
- Local\Packages — 1.92 ГБ (Store/UWP приложения)
- JetBrains: Local 1.2 + Roaming 1.24 ≈ 2.44 ГБ (кэши IDE)
- GitHubDesktop — 1.01 ГБ
- SquirrelTemp — 0.65 ГБ (почти всегда мусор)

Ты уже почистил NVIDIA Downloader/NV_Cache — ок.

Ниже — что можно чистить прямо сейчас, и что лучше делать аккуратно.

0) Перед чисткой

Закрой: Yandex/Opera, JetBrains IDE, GitHub Desktop, Windsurf/Cursor, TunnelBear.

Иначе часть файлов не удалится.

1) Быстро и безопасно (почти без риска)

SquirrelTemp (0.65 ГБ) — можно вымести полностью

powershell

```
Remove-Item -Recurse -Force "$env:LOCALAPPDATA\SquirrelTemp\*" -ErrorAction SilentlyContinue
```

Копировать код

CrashDumps — можно вымести

powershell

```
Remove-Item -Recurse -Force "$env:LOCALAPPDATA\CrashDumps\*" -ErrorAction SilentlyContinue
```

Копировать код

pip cache (0.18 ГБ) — чистится штатно

powershell

```
python -m pip cache purge
```

Копировать код

playwright-go (0.09 ГБ) — можно удалить, потом сам докачает

powershell

```
Remove-Item -Recurse -Force "$env:LOCALAPPDATA\ms-playwright-go\*" -ErrorAction SilentlyContinue
```

Копировать код

2) Самый жирный выигрыш: Yandex + Opera (кэши)

Yandex (3.11 ГБ)

Сначала покажем, что именно там жирное (чтобы не чистить "наугад"):

powershell

```
$root = "$env:LOCALAPPDATA\Yandex"
Get-ChildItem $root -Directory -Force -ErrorAction SilentlyContinue |
ForEach-Object {
    $sum = (Get-ChildItem $_.FullName -File -Recurse -Force -ErrorAction SilentlyContinue |
        Measure-Object Length -Sum).Sum
```

Копировать код

```
[pscustomobject]@{ Path=$_.FullName; GB=[math]::Round($sum/1GB,2) }
} | Sort-Object GB -Descending | Select-Object -First 20
```

Если это именно **Yandex Browser**, то можно безопасно удалить кэши (браузер закрыт!):

powershell

```
$paths = @(
"$env:LOCALAPPDATA\Yandex\YandexBrowser\User Data\Default\Cache",
"$env:LOCALAPPDATA\Yandex\YandexBrowser\User Data\Default\Code Cache",
"$env:LOCALAPPDATA\Yandex\YandexBrowser\User Data\Default\GPUCache",
"$env:LOCALAPPDATA\Yandex\YandexBrowser\User Data\ShaderCache"
)
$paths | ForEach-Object { Remove-Item -Recurse -Force $_ -ErrorAction SilentlyContinue }
```

Opera (Local 0.45 ГБ + Roaming 0.56 ГБ)

Тоже чистим кэши (Opera закрыт):

powershell

```
$paths = @(
"$env:LOCALAPPDATA\Opera Software\Opera Stable\Cache",
"$env:LOCALAPPDATA\Opera Software\Opera Stable\Code Cache",
"$env:LOCALAPPDATA\Opera Software\Opera Stable\GPUCache",
"$env:APPDATA\Opera Software\Opera Stable\Cache",
"$env:APPDATA\Opera Software\Opera Stable\Code Cache",
"$env:APPDATA\Opera Software\Opera Stable\GPUCache"
)
$paths | ForEach-Object { Remove-Item -Recurse -Force $_ -ErrorAction SilentlyContinue }
```

Обычно это даёт **несколько ГБ**, особенно если браузер активно живёт.

3) JetBrains (\approx 2.44 ГБ) — чистим кэши IDE, не настройки

JetBrains хранит много кэшей/индексов. **Настройки/плагины** трогать не надо, а вот **caches/log** — можно.

Сначала выведем “что внутри жирнее”:

powershell

```
$root = "$env:LOCALAPPDATA\JetBrains"
Get-ChildItem $root -Directory -Force -ErrorAction SilentlyContinue |
ForEach-Object {
    $sum = (Get-ChildItem $_.FullName -File -Recurse -Force -ErrorAction Silently
Measure-Object Length -Sum).Sum
    [pscustomobject]@{ Path=$_.FullName; GB=[math]::Round($sum/1GB,2) }
} | Sort-Object GB -Descending | Select-Object -First 20
```

Копировать код

Если хочешь “прибрать” руками (IDE закрыта):

- удаляй **только** caches и log в папках продуктов.

Универсально (аккуратно, но рабоче):

powershell

```
Get-ChildItem "$env:LOCALAPPDATA\JetBrains" -Directory -Force -ErrorAction Silent
ForEach-Object {
    Remove-Item -Recurse -Force (Join-Path $_.FullName "caches") -ErrorAction Sil
    Remove-Item -Recurse -Force (Join-Path $_.FullName "log") -ErrorAction Sil
}
```

Копировать код

После запуска IDE заново индексы пересоберутся (первый старт может быть чуть дольше).

4) Local\Programs (2.48 ГБ) — это “установленные в профиль” приложения

Тут лучше **не удалять** папки вручную, а:

- Параметры → Приложения → Установленные приложения → сортировка по размеру
и удалить ненужное, или переустановить на D: (если установщик позволяет выбрать путь).

Хочешь — можно быстро посмотреть, что именно там жирное:

powershell

```
$root = "$env:LOCALAPPDATA\Programs"
Get-ChildItem $root -Directory -Force -ErrorAction SilentlyContinue |
```

```
ForEach-Object {
    $sum = (Get-ChildItem $_.FullName -File -Recurse -Force -ErrorAction SilentlyContinue | Measure-Object Length -Sum).Sum
    [pscustomobject]@{ Path=$_.FullName; GB=[math]::Round($sum/1GB, 2) }
} | Sort-Object GB -Descending | Select-Object -First 30
```

Копировать код

5) ProgramData\NVIDIA Corporation (3.59 ГБ)

Ты уже сделал правильные две чистки. Ещё 2 места, которые часто жирные:

1. C:\NVIDIA (папка распаковки драйверов) — можно удалить, если есть:

powershell

```
Remove-Item -Recurse -Force "C:\NVIDIA" -ErrorAction SilentlyContinue
```

Копировать код

2. Installer2 (может быть большой). Тут аккуратно: удаление лишает "отката" некоторых компонентов.

Я бы сначала посмотрел размер:

powershell

```
$root = "C:\ProgramData\NVIDIA Corporation"
Get-ChildItem $root -Directory -Force -ErrorAction SilentlyContinue |
ForEach-Object {
    $sum = (Get-ChildItem $_.FullName -File -Recurse -Force -ErrorAction SilentlyContinue | Measure-Object Length -Sum).Sum
    [pscustomobject]@{ Path=$_.FullName; GB=[math]::Round($sum/1GB, 2) }
} | Sort-Object GB -Descending | Select-Object -First 20
```

Копировать код

Проверка результата

После пары шагов:

powershell

```
[math]::Round((Get-PSDrive C).Free/1GB, 2)
```

Копировать код

Мини-пояснение про Win11 readiness (чтобы не путаться)

Твоё “6.78 GB свободно” в отчёте — это **до чистки**. Сейчас у тебя уже **30.69**, т.е. этот блокер снят.

Но останутся: **Secure Boot = NO**, **TPM нет**, и **CPU 4-го поколения** (официально мимо Win11).

Если хочешь — сделаем максимально эффективно:

Сделай **ещё три вывода** (они короткие), и я скажу “удалить вот это / не трогать вот это” без гаданий:

1. Топ внутри Local\Yandex (ты уже готов)
2. Топ внутри Local\Microsoft
3. Топ внутри Local\Packages

Команды такие же, как я давал (подставлю пути, если надо).



```
PS C:\Windows\system32> $paths = @(
>> "$env:LOCALAPPDATA\Yandex\YandexBrowser\User
Data\Default\Cache",
>> "$env:LOCALAPPDATA\Yandex\YandexBrowser\User
Data\Default\Code Cache",
>> "$env:LOCALAPPDATA\Yandex\YandexBrowser\User
Data\Default\GPUCache",
>> "$env:LOCALAPPDATA\Yandex\YandexBrowser\User
Data\ShaderCache"
>> )
>> $paths | ForEach-Object { Remove-Item -Recurse -
Force $_ -ErrorAction SilentlyContinue }
>>
PS C:\Windows\system32> $paths = @(
>> "$env:LOCALAPPDATA\Opera Software\Opera
Stable\Cache",
>> "$env:LOCALAPPDATA\Opera Software\Opera
Stable\Code Cache",
>> "$env:LOCALAPPDATA\Opera Software\Opera
Stable\GPUCache",
```

```

>> "$env:APPDATA\Opera Software\Opera Stable\Cache",
>> "$env:APPDATA\Opera Software\Opera Stable\Code
Cache",
>> "$env:APPDATA\Opera Software\Opera
Stable\GPUCache"
>> )
>> $paths | ForEach-Object { Remove-Item -Recurse -
Force $_ -ErrorAction SilentlyContinue }
>>
PS C:\Windows\system32> $root =
"$env:LOCALAPPDATA\JetBrains"
>> Get-ChildItem $root -Directory -Force -ErrorAction
SilentlyContinue |
>> ForEach-Object {
>>   $sum = (Get-ChildItem $_.FullName -File -Recurse -
Force -ErrorAction SilentlyContinue |
>>     Measure-Object Length -Sum).Sum
>>   [pscustomobject]@{ Path=$_.FullName; GB=
[math]::Round($sum/1GB,2) }
>> } | Sort-Object GB -Descending | Select-Object -First
20
>>

```

Path	GB
---	--
C:\Users\Славиус\AppData\Local\JetBrains\IdeaIC2024.3	
1.2	

PS C:\Windows\system32>

```

PS C:\Windows\system32> Remove-Item -Recurse -Force
"C:\NVIDIA" -ErrorAction SilentlyContinue
>>
PS C:\Windows\system32> $root =
"C:\ProgramData\NVIDIA Corporation"
>> Get-ChildItem $root -Directory -Force -ErrorAction
SilentlyContinue |
>> ForEach-Object {
>>   $sum = (Get-ChildItem $_.FullName -File -Recurse -
Force -ErrorAction SilentlyContinue |
>>     Measure-Object Length -Sum).Sum

```

```
>> [pscustomobject]@{ Path=$_._FullName; GB=
[math]::Round($sum/1GB,2) }
>> } | Sort-Object GB -Descending | Select-Object -First
20
>>
```

Path	GB
---	--
C:\ProgramData\NVIDIA Corporation\NVIDIA app	
3.57	
C:\ProgramData\NVIDIA Corporation\ShadowPlay	
0.01	
C:\ProgramData\NVIDIA	
Corporation\NvProfileUpdaterPlugin 0.01	
C:\ProgramData\NVIDIA Corporation\nvtopp	0
C:\ProgramData\NVIDIA Corporation\umdlogs	
0	
C:\ProgramData\NVIDIA Corporation\NvVAD	
0	
C:\ProgramData\NVIDIA Corporation\NvTelemetry	
0	
C:\ProgramData\NVIDIA Corporation\FrameViewSDK	
0	
C:\ProgramData\NVIDIA Corporation\Dr	0
C:\ProgramData\NVIDIA Corporation\DisplayDriverRAS	
0	
C:\ProgramData\NVIDIA	
Corporation\GameSessionTelemetry	0

PS C:\Windows\system32>

Красиво работаешь 😊 Тебя “жир” почти весь **в одном месте**:

- **ProgramData\NVIDIA Corporation\NVIDIA app = 3.57 ГБ** — это и есть тот самый слон.
- **JetBrains Local: IdeaIC2024.3 = 1.2 ГБ**
- Кэши Yandex/Opera ты уже прибрал (это хорошо, даже если эффект не сразу видно — иногда размер пересчитывается через минуту/после перезапуска).

Дальше — два направления: (A) безопасно освободить ещё место, (B) не убить себе драйвера/IDE.

A) NVIDIA app 3.57 ГБ — что это и что с ним делать

C:\ProgramData\NVIDIA Corporation\NVIDIA app — это данные нового “NVIDIA App” (замена/наследник GeForce Experience): кэш, телеметрия, возможно скачанные компоненты/контент.

Самый безопасный способ освободить реально много

Удалить/переустановить NVIDIA App через “Приложения”:

- Параметры → Приложения → Установленные приложения
- найди NVIDIA App / GeForce Experience (если есть)
- если не пользуешься оверлеем/ShadowPlay — удаляй.
- драйвер видеокарты при этом обычно остаётся.

Это почти всегда даёт **несколько ГБ** и не ломает работу видеокарты.

Если хочешь оставить NVIDIA App, но чистить кэши

Нужно посмотреть, что внутри этих 3.57 ГБ. Дай топ-30 подпапок:

powershell

```
$root = "C:\ProgramData\NVIDIA Corporation\NVIDIA app"
Get-ChildItem $root -Directory -Force -ErrorAction SilentlyContinue |
ForEach-Object {
    $sum = (Get-ChildItem $_.FullName -File -Recurse -Force -ErrorAction Silently
Measure-Object Length -Sum).Sum
    [pscustomobject]@{ Path= $_.FullName; GB=[math]::Round($sum/1GB,2) }
} | Sort-Object GB -Descending | Select-Object -First 30
```

Copировать код

После этого я скажу “вот эти 1–2 папки можно вычистить”, а вот “эти не трогай”.

Важное: я не буду предлагать “удали всю папку NVIDIA app” вслепую — можно словить поломку обновлятора/оверлея.

B) JetBrains (IdeaIC2024.3 = 1.2 ГБ) — можно уменьшить без боли

Если IDE закрыта, безопасно прибрать кэши:

powershell

```
Remove-Item -Recurse -Force "$env:LOCALAPPDATA\JetBrains\IdeaIC2024.3\caches" -Er
Remove-Item -Recurse -Force "$env:LOCALAPPDATA\JetBrains\IdeaIC2024.3\log" -Er
```

 Копировать код

Потом IDEA запустится и заново всё переиндексирует (первый старт может быть чуть медленнее). Обычно это минус **сотни МБ**, иногда больше.

C) Быстрый контроль: сколько стало свободно на С

Сейчас обязательно проверь (после всех действий):

powershell

```
[math]::Round((Get-PSDrive C).Free/1GB,2)
```

 Копировать код

D) Почему отчёт win11_readiness показывает 6.78 GB

Потому что он был запущен до чистки. Перезапусти скрипт сейчас — будет новое число (у тебя уже 30+).

Что я бы сделал на твоём месте (максимум эффекта за минимум риска)

1. Проверил свободное место (команда выше).
2. Сделал JetBrains caches/log (быстро и без крови).
3. Решил судьбу NVIDIA App: удалить как программу (самый чистый вариант) или показать мне топ подпапок внутри NVIDIA app , чтобы чистить точечно.

Кидай:

- результат свободного места на С после этих чисток
- и/или вывод топ-30 подпапок внутри C:\ProgramData\NVIDIA Corporation\NVIDIA app — и я скажу, где там лежит “жир”, который можно безопасно выкинуть.



Windows PowerShell

(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

Попробуйте новую кроссплатформенную оболочку
PowerShell (<https://aka.ms/psscure6>)

```
PS C:\Windows\system32> $root =  
"C:\ProgramData\NVIDIA Corporation\NVIDIA app"  
>> Get-ChildItem $root -Directory -Force -ErrorAction  
SilentlyContinue |  
>> ForEach-Object {  
>>     $sum = (Get-ChildItem $_.FullName -File -Recurse -  
Force -ErrorAction SilentlyContinue |  
>>         Measure-Object Length -Sum).Sum  
>>     [pscustomobject]@{ Path=$_.FullName; GB=  
[math]::Round($sum/1GB,2) }  
>> } | Sort-Object GB -Descending | Select-Object -First  
30  
>>
```

Path	GB
---	--
C:\ProgramData\NVIDIA Corporation\NVIDIA app\UpdateFramework	3.56
C:\ProgramData\NVIDIA Corporation\NVIDIA app\NvDriverDiagnostics	0
C:\ProgramData\NVIDIA Corporation\NVIDIA app\NvContainer	0
C:\ProgramData\NVIDIA Corporation\NVIDIA app\UXD	0
C:\ProgramData\NVIDIA Corporation\NVIDIA app\SessionLogs	0
C:\ProgramData\NVIDIA Corporation\NVIDIA app\Logs	0
C:\ProgramData\NVIDIA Corporation\NVIDIA app\Installer	0
C:\ProgramData\NVIDIA Corporation\NVIDIA app\MessageBus	0
C:\ProgramData\NVIDIA Corporation\NVIDIA app\NvConfig	0

```
C:\ProgramData\NVIDIA Corporation\NVIDIA
app\NvBackend      0
```

```
PS C:\Windows\system32> Remove-Item -Recurse -Force
"$env:LOCALAPPDATA\JetBrains\IdealC2024.3\caches" -
ErrorAction SilentlyContinue
>> Remove-Item -Recurse -Force
"$env:LOCALAPPDATA\JetBrains\IdealC2024.3\log" -
ErrorAction SilentlyContinue
>>
PS C:\Windows\system32> [math]::Round((Get-PSDrive
C).Free/1GB,2)
>>
31.42
PS C:\Windows\system32>
```

Окей, теперь всё предельно конкретно:

- **Почти весь жир NVIDIA app сидит в**
C:\ProgramData\NVIDIA Corporation\NVIDIA app\UpdateFramework = 3.56 ГБ
- JetBrains caches/log ты снёс — красавчик.
- Свободно стало **31.42 ГБ** (чуть подросло — значит кеши реально съелись).

Теперь важное: **UpdateFramework** — это не “кэш картинок”, а **фреймворк/модуль обновлений NVIDIA App** + его данные. Если снести его “в лоб”, приложение может:

- начать качать всё заново (вернётся обратно),
- или сломать апдейтер / NVIDIA App,
- или восстановиться при следующем запуске.

Так что даю 2 варианта: “чисто” и “быстро”.

Вариант 1 (рекомендую): удалить/отключить NVIDIA App как программу

Если тебе не нужен оверлей, ShadowPlay, авто-обновлялка, оптимизация игр — это самый безопасный способ вернуть эти 3.5 ГБ навсегда.

Параметры → Приложения → Установленные приложения:

- NVIDIA App (или GeForce Experience) → Удалить

Драйвер видеокарты при этом обычно остаётся. Максимум потеряешь "фишки" типа оверлея/записи/автообновлений.

Вариант 2 (быстро, но чуть рискованнее): почистить именно UpdateFramework

Если хочешь оставить NVIDIA App — можно попробовать "почистить кэш" UpdateFramework, но надо действовать аккуратно.

2.1 Сначала остановим службы/процессы NVIDIA (чтобы файлы не были заняты)

Запусти PowerShell от администратора, затем:

```
powershell
```

```
Stop-Process -Name "NVIDIA*" -Force -ErrorAction SilentlyContinue
Stop-Service -Name "NVDisplay.ContainerLocalSystem" -Force -ErrorAction SilentlyContinue
```

Копировать код

2.2 Переименуем папку (это безопаснее, чем удалять)

Так ты всегда сможешь откатить:

```
powershell
```

```
Rename-Item "C:\ProgramData\NVIDIA Corporation\NVIDIA app\UpdateFramework" "Updat
```

Копировать код

2.3 Перезагрузка

После ребута:

- если всё ок — NVIDIA App создаст новую папку, обычно значительно меньше.
- если что-то сломалось — просто вернёшь назад имя папки.

Откат:

```
powershell
```

```
Rename-Item "C:\ProgramData\NVIDIA Corporation\NVIDIA app\UpdateFramework.bak" "Updat
```

Копировать код

2.4 Если всё ок — удалить .bak окончательно

powershell

```
Remove-Item -Recurse -Force "C:\ProgramData\NVIDIA Corporation\NVIDIA App\*" <input>
```

Копировать код

Как понять, стоит ли вообще заморачиваться

У тебя сейчас 31.42 ГБ свободно — это уже хорошо для апдейта/жизни системы.

Если цель была “снять блокер по месту” — ты уже победил.

Мой честный совет

- Если NVIDIA App тебе не нужен → удаляй через “Приложения” (вариант 1).
- Если нужен → делай переименование в .bak (вариант 2), это самый безопасный хак без “удалил и молись”.

Скажи одно: ты пользуешься записью экрана/оверлеем (ShadowPlay),
автоподбором настроек игр или автообновлением драйверов?

Если нет — вообще не стесняйся выкидывать NVIDIA App, драйверу пофиг.



нет

но запиши всю историю, скрипты, историю запросов
результатов и историю чатов и тд в лог или несколько
файлов или архив

подготовь скрипты для очистки

напиши промт мб даже для другой беседы чтобы
интегрировать или объединить логику

тот же чат гпт

только в другом чате (беседе)

постараюсь сделать одну сборку файлов скриптов

запусков батников и тд

и по сути все что мы делали

по шагам или вместе

Думал на протяжении 27s >

Проанализировано ^

python

```
import os, textwrap, zipfile, da Всегда показывать подробности
```

Копировать код

```
base = "/mnt/data/win11_cleanup_bundle_2025-12-27"
```

```
if os.path.exists(base):
```

```
    shutil.rmtree(base)
```

```
os.makedirs(base, exist_ok=True)
```

```
scripts_dir = os.path.join(base, "scripts")
```

```
logs_dir = os.path.join(base, "logs")
```

```
docs_dir = os.path.join(base, "docs")
```

```
os.makedirs(scripts_dir, exist_ok=True)
```

```
os.makedirs(logs_dir, exist_ok=True)
```

```
os.makedirs(docs_dir, exist_ok=True)
```

```
# Build transcript from user-provided content (best-effort)
```

```
transcript = r"""
```

```
== Session transcript (user-provided) ==
```

```
Date: 2025-12-27
```

```
PS C:\Users\Славиус\Desktop> python .\win11_readiness_report_v2.py --outdir .
```

```
== Windows 11 Readiness / System Inventory Report (v2) ==
```

```
Generated: 2025-12-27T14:15:56
```

```
Host: DESKTOP-10AT63D | User: Славиус | SystemDrive: C:
```

```
...
```

```
C: | Size=111.4 GB | Free=6.78 GB | Health=Healthy
```

```
D: | Size=111.4 GB | Free=106.38 GB | Health=Healthy
```

```
Verdict: NOT READY
```

```
- [FAIL] System drive free space: 6.78 GB free (required: >= 25 GB)
```

```
- [FAIL] Secure Boot: NO
```

```
- [FAIL] TPM: Present=NO
```

```
Commands executed:
```

```
- powercfg -h off
```

```
- DISM /Online /Cleanup-Image /StartComponentCleanup
```

```
- cleanmgr
```

```
- (attempted) clean -> CommandNotFoundException in PowerShell (NOT diskpart)
```

```
DISM result:
```

```
The operation completed successfully.
```

Windows Update cleanup (stop services, clear downloads, start services):

- net stop wuauserv
- net stop bits
- Remove-Item -Recurse -Force C:\Windows\SoftwareDistribution\Download*
- net start bits
- net start wuauserv

Recycle bin + TEMP cleanup:

- Clear-RecycleBin -Force
- Get-ChildItem \$env:TEMP -Force | Remove-Item -Recurse -Force

Free space progress:

- Before: ~16 GB free
- After: ~29 GB free
- After additional steps: 30.69 GB free
- After JetBrains cleanup: 31.42 GB free

Disk usage highlights (selected):

- pagefile.sys 7.16 GB
- AppData total 17.66 GB (Local 14.5 GB, Roaming 2.97 GB)
- OneDrive 7.69 GB
- ProgramData 6.43 GB
- ProgramData\NVIDIA Corporation 3.59 GB (mostly NVIDIA app UpdateFramework 3.56 !)

Browser caches removed (best-effort):

- Yandex Browser caches (Cache / Code Cache / GPU Cache / Shader Cache)
- Opera caches (Cache / Code Cache / GPU Cache)

JetBrains caches/log removed:

- %LOCALAPPDATA%\JetBrains\IdeaIC2024.3\caches
- %LOCALAPPDATA%\JetBrains\IdeaIC2024.3\log

NVIDIA caches removed (best-effort):

- C:\ProgramData\NVIDIA Corporation\Downloader*
- C:\ProgramData\NVIDIA Corporation\NV_Cache*
- C:\NVIDIA (if present)

Notes:

- ">>" prompt indicated PowerShell continuation mode; resolved with Ctrl+C.
 - vssadmin list shadowstorage returned none (no shadow storage entries).
- ==== End transcript ====
""".strip() + "\n"

```

with open(os.path.join(logs_dir, "session_transcript_2025-12-27.txt"), "w", encoding='utf-8') as f:
    f.write(transcript)

# Docs: steps
steps_md = textwrap.dedent("""
# Windows cleanup + Win11 readiness – session pack (2025-12-27)

This bundle captures what we did, plus reusable scripts to repeat the cleanup safely.

## What changed
- Free space on **C:** increased to **~31.42 GB**.
- Win11 readiness blockers still remain:
  - **Secure Boot:** disabled
  - **TPM:** not present/ready
  - CPU is 4th-gen Intel (often not officially supported for Win11)

## Safety notes
- **PowerShell `clean` is NOT a thing**; the dangerous command is `clean` **inside** PowerShell
- Most scripts here require **Administrator** PowerShell to stop services and clean up
- Some cleanups (JetBrains caches, browser caches) require the apps to be **closed** before running

## Recommended run order
1. `01_cleanup_system.ps1` – hibernation off, DISM component cleanup, opens Disk Management
2. `02_cleanup_updates_temp.ps1` – clears Windows Update download cache + recycle bin
3. `03_cleanup_apps_caches.ps1` – browser caches, JetBrains caches/log, pip cache
4. `04_audit_sizes.ps1` – shows what's still large.

## NVIDIA App (optional)
Your `C:\ProgramData\NVIDIA Corporation\NVIDIA App\UpdateFramework` was ~3.56 GB
If you don't use NVIDIA App features (overlay/recording/auto-updates), the cleaner
- **Uninstall NVIDIA App** from *Settings → Apps → Installed apps*.

A safer script alternative is to **rename** `UpdateFramework` to `UpdateFrameworkSafe`.
See `05_nvidia_updateframework_safe_rename.ps1`.

## What's included
- `logs/` – transcript of the session (best-effort from pasted output).
- `scripts/` – PowerShell scripts + batch runner.
- `docs/` – this guide + a prompt to continue in another chat.

""").strip() + "\n"

```

```
with open(os.path.join(docs_dir, "STEPS.md"), "w", encoding="utf-8") as f:  
    f.write(steps_md)  
  
# Prompt for another chat  
prompt_md = textwrap.dedent("""  
# Prompt to continue in a new ChatGPT chat
```

Paste the text below into a new chat to restore context quickly:

We worked on freeing space on Windows 10 (C: was nearly full) and checking Window

System highlights:

- Windows 10 Home, build 19045
- CPU: Intel Core i3-4170
- RAM: ~12 GB
- Boot mode: UEFI
- Secure Boot: NO
- TPM: not present/ready
- Disk 0 GPT (system), Disk 1 MBR

Actions performed:

- 1) `powercfg -h off` (hibernation off)
- 2) `DISM /Online /Cleanup-Image /StartComponentCleanup`
- 3) Disk Cleanup (`cleanmgr`)
- 4) Cleaned Windows Update downloads:
 - stop wuauserv/bits
 - delete `C:\\Windows\\SoftwareDistribution\\Download*`
 - start bits/wuauserv
- 5) Cleared recycle bin + `%TEMP%`
- 6) Cleaned browser caches (Yandex + Opera)
- 7) Cleaned JetBrains IDEA caches/log (`%LOCALAPPDATA%\\JetBrains\\IdeaIC2024.3\\c`
- 8) Cleaned NVIDIA caches and removed `C:\\NVIDIA` if present

Result:

- Free space on C: improved from ~16 GB to ~31.42 GB.

Next goals:

- Keep C: above 25-30 GB free long-term.
- Optional: reduce bloat in AppData (Yandex/Opera/JetBrains) and consider uninsta
- Win11 readiness still blocked by Secure Boot/TPM/CPU; provide options with hone

Please:

- Review the included PowerShell cleanup scripts and suggest improvements.
- Provide a long-term maintenance plan (what to clean weekly/monthly, what to move)
- Provide a safe way to handle NVIDIA App's large UpdateFramework (prefer uninsta

```
"""
    """").strip() + "\n"

with open(os.path.join(docs_dir, "PROMPT_FOR_NEW_CHAT.md"), "w", encoding="utf-8") as f:
    f.write(prompt_md)

# README
readme = textwrap.dedent("""
# Win11 readiness + cleanup pack

## Quick start
1. Extract this zip somewhere, e.g. `C:\\\\tools\\\\win11_cleanup_pack` .
2. Right-click **PowerShell** → **Run as administrator**.
3. Run:

```powershell
Set-Location C:\\\\tools\\\\win11_cleanup_pack\\\\scripts
powershell -ExecutionPolicy Bypass -File .\\\\run_all.ps1
```
""")
```

Or run scripts one by one in order (see docs/STEPS.md).

Files

- docs/STEPS.md
- docs/PROMPT_FOR_NEW_CHAT.md
- logs/session_transcript_2025-12-27.txt
- scripts/*.ps1
- scripts/run_all_as_admin.bat

```
"""
    """").strip() + "\n"
```

with open(os.path.join(base, "README.md"), "w", encoding="utf-8") as f:
f.write(readme)

Scripts content

```
ps_common_header = textwrap.dedent(r"""
#Requires -Version 5.1
```

```

Set-StrictMode -Version Latest
$ErrorActionPreference = "Continue"

function Write-Section($title) {
    Write-Host ""
    Write-Host ("=" * 70)
    Write-Host $title
    Write-Host ("=" * 70)
}

function Assert-Admin {
    $isAdmin = ([Security.Principal.WindowsPrincipal]
    [Security.Principal.WindowsIdentity]::GetCurrent()
    ).IsInRole([Security.Principal.WindowsBuiltinRole]::Administrator)
    if (-not $isAdmin) {
        Write-Warning "Run this script in an elevated (Administrator) PowerShell for best results."
    }
}
""").lstrip()

```

01: system cleanup

```

cleanup_system = ps_common_header + textwrap.dedent(r"""
Assert-Admin
Write-Section "01 — System cleanup (hibernation off + DISM component cleanup + Disk
Cleanup UI)"

Write-Host "Disabling hibernation (frees hiberfil.sys if present)..."
try { powercfg -h off | Out-Null } catch { Write-Warning $_ }

Write-Host "Running DISM component cleanup (may take a while)..."
try {
    DISM /Online /Cleanup-Image /StartComponentCleanup
} catch {
    Write-Warning $_
}

Write-Host "Launching Disk Cleanup (cleanmgr). Use 'Clean up system files' inside the UI
for best effect."
try { Start-Process cleanmgr } catch { Write-Warning $_ }

```

```

Write-Host "Done."
""").strip() + "\n"

with open(os.path.join/scripts_dir, "01_cleanup_system.ps1"), "w", encoding="utf-8") as f:
f.write(cleanup_system)

```

02: updates temp

```

cleanup_updates = ps_common_header + textwrap.dedent(r"""
Assert-Admin

Write-Section "02 — Windows Update downloads + Recycle Bin + TEMP"

```

Stop services (ignore errors)

```

Write-Host "Stopping Windows Update (wuauserv) and BITS..."
try { net stop wuauserv | Out-Host } catch { }
try { net stop bits | Out-Host } catch { }

```

Clear update download cache

```

$dl = Join-Path $env:windir "SoftwareDistribution\Download"
Write-Host "Clearing: $dl*"
try { Remove-Item -Recurse -Force (Join-Path $dl "*") -ErrorAction SilentlyContinue } catch
{ Write-Warning $_ }

```

Restart services

```

Write-Host "Starting BITS and Windows Update..."
try { net start bits | Out-Host } catch { }
try { net start wuauserv | Out-Host } catch { }

```

Recycle bin

```

Write-Host "Clearing Recycle Bin..."
try { Clear-RecycleBin -Force -ErrorAction SilentlyContinue } catch { }

```

TEMP

```

Write-Host "Clearing TEMP folder..."
try {
Get-ChildItem $env:TEMP -Force -ErrorAction SilentlyContinue | Remove-Item -Recurse -
Force -ErrorAction SilentlyContinue
} catch { Write-Warning $_ }

Write-Host "Done."
""").strip() + "\n"

```

```
with open(os.path.join(scripts_dir, "02_cleanup_updates_temp.ps1"), "w", encoding="utf-8") as f:
    f.write(cleanup_updates)
```

03: apps caches

```
cleanup_apps = ps_common_header + textwrap.dedent(r"""
Write-Section "03 — App caches (browsers, JetBrains, pip, CrashDumps)"

Write-Host "Tip: close Yandex/Opera/JetBrains/Windsurf/Cursor/GitHub Desktop before
running."
```

Yandex Browser cache paths (best-effort)

```
$yandexPaths = @(
    "$env:LOCALAPPDATA\Yandex\YandexBrowser\User Data\Default\Cache",
    "$env:LOCALAPPDATA\Yandex\YandexBrowser\User Data\Default\Code Cache",
    "$env:LOCALAPPDATA\Yandex\YandexBrowser\User Data\Default\GPUCache",
    "$env:LOCALAPPDATA\Yandex\YandexBrowser\User Data\ShaderCache"
)
Write-Host "Cleaning Yandex Browser caches..."
```

ChatGPT 5.2 Thinking ▾



SilentlyContinue }

Opera caches (best-effort)

```
$operaPaths = @(
    "$env:LOCALAPPDATA\Opera Software\Opera Stable\Cache",
    "$env:LOCALAPPDATA\Opera Software\Opera Stable\Code Cache",
    "$env:LOCALAPPDATA\Opera Software\Opera Stable\GPUCache",
    "$env:APPDATA\Opera Software\Opera Stable\Cache",
    "$env:APPDATA\Opera Software\Opera Stable\Code Cache",
    "$env:APPDATA\Opera Software\Opera Stable\GPUCache"
)
Write-Host "Cleaning Opera caches..."
```

```
$operaPaths | ForEach-Object { Remove-Item -Recurse -Force $_ -ErrorAction
SilentlyContinue }
```

JetBrains IDEA caches/logs (only caches/log; leaves config/plugins)

```
$ideaLocal = Join-Path $env:LOCALAPPDATA "JetBrains\IdeaIC2024.3"
if (Test-Path $ideaLocal) {
    Write-Host "Cleaning JetBrains IDEA caches/log in: $ideaLocal"
    Remove-Item -Recurse -Force (Join-Path $ideaLocal "caches") -ErrorAction
    SilentlyContinue
    Remove-Item -Recurse -Force (Join-Path $ideaLocal "log") -ErrorAction SilentlyContinue
} else {
    Write-Host "JetBrains IDEA folder not found at $ideaLocal (skip)."
}
```

Crash dumps

```
Write-Host "Cleaning CrashDumps..."
Remove-Item -Recurse -Force "$env:LOCALAPPDATA\CrashDumps*" -ErrorAction
SilentlyContinue
```

SquirrelTemp

```
Write-Host "Cleaning SquirrelTemp..."
Remove-Item -Recurse -Force "$env:LOCALAPPDATA\SquirrelTemp*" -ErrorAction
SilentlyContinue
```

pip cache purge (if python exists)

```
Write-Host "Purging pip cache (if Python is available)..."
try { python -m pip cache purge | Out-Host } catch { Write-Host "pip cache purge skipped
(python/pip not available)." }

Write-Host "Done."
""").strip() + "\n"
```

with open(os.path.join(scripts_dir, "03_cleanup_apps_caches.ps1"), "w", encoding="utf-8")
as f:
f.write(cleanup_apps)

04 audit sizes

```
audit_sizes = ps_common_header + textwrap.dedent(r"""
Write-Section "04 — Size audit (top folders in profile and ProgramData)"

function Get-TopDirs($rootPath, $top = 20) {
```

```
    Write-Host ""
    Write-Host "Top $top directories under: $rootPath"
    try {
```

```

Get-ChildItem $rootPath -Directory -Force -ErrorAction SilentlyContinue |
ForEach-Object {
$sum = 0
try {
$sum = (Get-ChildItem $.FullName -File -Recurse -Force -ErrorAction SilentlyContinue |
Measure-Object Length -Sum).Sum
} catch {}  

[pscustomobject]@{ Path = $.FullName; GB = [math]::Round($sum/1GB,2) }
} | Sort-Object GB -Descending | Select-Object -First $top
} catch {
Write-Warning $_
}
}

```

```

Get-TopDirs $env:USERPROFILE 25
Get-TopDirs "$env:USERPROFILE\AppData\Local" 25
Get-TopDirs "$env:USERPROFILE\AppData\Roaming" 25
Get-TopDirs "C:\ProgramData" 25

```

```

Write-Host ""
Write-Host "Free space on C: (GB): " -NoNewline
Write-Host ([math]::Round((Get-PSDrive C).Free/1GB,2))
""").strip() + "\n"

```

```

with open(os.path.join(scripts_dir, "04_audit_sizes.ps1"), "w", encoding="utf-8") as f:
f.write(audit_sizes)

```

05 Nvidia safe rename

```

nvidia_rename = ps_common_header + textwrap.dedent(r"""
Assert-Admin
Write-Section "05 — NVIDIA UpdateFramework safe rename (optional)"

```

<#

This script is OPTIONAL.

It renames the heavy folder:

```

C:\ProgramData\NVIDIA Corporation\NVIDIA app\UpdateFramework
to:
...\\UpdateFramework.bak

```

Why rename instead of delete?

- Easy rollback if NVIDIA App breaks.
- Lets you see if the folder is recreated smaller.

Steps:

1. Run as Administrator.
2. Reboot after rename.
3. If everything works, you may delete the .bak later.

Rollback:

```
Rename-Item UpdateFramework.bak UpdateFramework
```

```
#>
```

```
$src = "C:\ProgramData\NVIDIA Corporation\NVIDIA app\UpdateFramework"
$bak = "C:\ProgramData\NVIDIA Corporation\NVIDIA app\UpdateFramework.bak"
```

```
if (-not (Test-Path $src)) {
    Write-Host "Source folder not found: $src"
    exit 0
}
```

```
Write-Host "Stopping NVIDIA processes/services (best-effort)..."
try { Stop-Process -Name "NVIDIA*" -Force -ErrorAction SilentlyContinue } catch { }
try { Stop-Service -Name "NVDISPLAY.ContainerLocalSystem" -Force -ErrorAction
    SilentlyContinue } catch { }
```

```
Write-Host "Renaming:"
Write-Host "$src"
Write-Host "to"
Write-Host "$bak"
try {
    if (Test-Path $bak) {
        Write-Warning "Backup already exists ($bak). Delete/rename it first if you want to
proceed."
        exit 1
    }
}
```

```
Rename-Item $src (Split-Path $bak -Leaf) -ErrorAction Stop
Write-Host "Done. Reboot recommended."
} catch {
```

```
Write-Warning $_
Write-Host "Nothing changed."
exit 1
}
""").strip() + "\n"

with open(os.path.join/scripts_dir, "05_nvidia_updateframework_safe_rename.ps1"), "w",
encoding="utf-8") as f:
f.write(nvidia_rename)
```

run_all.ps1

```
run_all = ps_common_header + textwrap.dedent(r"""
Write-Section "RUN ALL — orchestrator"

$here = Split-Path -Parent $MyInvocation.MyCommand.Path
Set-Location $here

$files = @(
"01_cleanup_system.ps1",
"02_cleanup_updates_temp.ps1",
"03_cleanup_apps_caches.ps1",
"04_audit_sizes.ps1"
)

foreach ($f in $files) {
Write-Section "Running $f"
try {
powershell -ExecutionPolicy Bypass -File (Join-Path $here $f)
} catch {
Write-Warning $_
}
}

Write-Host """
Write-Host "Optional: run 05_nvidia_updateframework_safe_rename.ps1 if you want to try
shrinking NVIDIA App footprint."
""").strip() + "\n"

with open(os.path.join/scripts_dir, "run_all.ps1"), "w", encoding="utf-8") as f:
f.write(run_all)
```

Batch runner

```
bat = textwrap.dedent(r"""
@echo off
setlocal
set SCRIPT_DIR=%~dp0
echo Running cleanup scripts from: %SCRIPT_DIR%
echo.
echo IMPORTANT: Right-click this .bat and choose "Run as administrator".
echo.

powershell -NoProfile -ExecutionPolicy Bypass -File "%SCRIPT_DIR%run_all.ps1"

echo.
echo Done. Press any key to exit.
pause >nul
""").strip() + "\n"

with open(os.path.join(scripts_dir, "run_all_as_admin.bat"), "w", encoding="utf-8") as f:
    f.write(bat)
```

Create zip

```
zip_path = "/mnt/data/win11_cleanup_bundle_2025-12-27.zip"
if os.path.exists(zip_path):
    os.remove(zip_path)

with zipfile.ZipFile(zip_path, "w", compression=zipfile.ZIP_DEFLATED) as z:
    for root, dirs, files in os.walk(base):
        for fn in files:
            full = os.path.join(root, fn)
            rel = os.path.relpath(full, os.path.dirname(base))
            z.write(full, rel)

zip_path
```

Всегда показывать подробности

Копировать код

Результат

'/mnt/data/win11_cleanup_bundle_2025-12-27.zip'

Готово — я собрал **полный пакет**: лог сессии + пошаговая инструкция + набор PowerShell-скриптов + батник для запуска + промт для переноса в новый чат.

[Скачать архив](#)

Что внутри:

- README.md — как быстро запустить
- docs/STEPS.md — **вся история по шагам** (что делали и зачем)
- docs/PROMPT_FOR_NEW_CHAT.md — промт, чтобы в новом чате сразу восстановить контекст
- logs/session_transcript_2025-12-27.txt — **лог всей истории** (из твоих вставок)
- scripts/
 - 01_cleanup_system.ps1 — powercfg -h off + DISM cleanup + открывает cleanmgr
 - 02_cleanup_updates_temp.ps1 — чистка Windows Update Download + TEMP + корзина
 - 03_cleanup_apps_caches.ps1 — кэши Yandex/Opera + JetBrains caches/log + CrashDumps + SquirrelTemp + pip cache purge
 - 04_audit_sizes.ps1 — аудит размеров (что ещё жрёт место)
 - 05_nvidia_updateframework_safe_rename.ps1 — **опционально**: безопасное переименование UpdateFramework в .bak с возможностью отката
 - run_all.ps1 — прогон 01→04
 - run_all_as_admin.bat — запуск “как человек” (ПКМ → Run as administrator)

Если хочешь, я ещё могу сделать “сборку под тебя”:

- добавить логирование прямо в скрипты (чтобы они сами писали logs\run_YYYYMMDD_HHMM.txt)
- сделать “тихий режим” без cleanmgr (чтобы полностью без UI)
- добавить отдельный скрипт “OneDrive: сделать папки online-only” (без удаления файлов)

Скажи только: хочешь чистку NVIDIA App через удаление приложения (самый чистый вариант) или оставляем “rename UpdateFramework” как опциональный шаг?

</>



