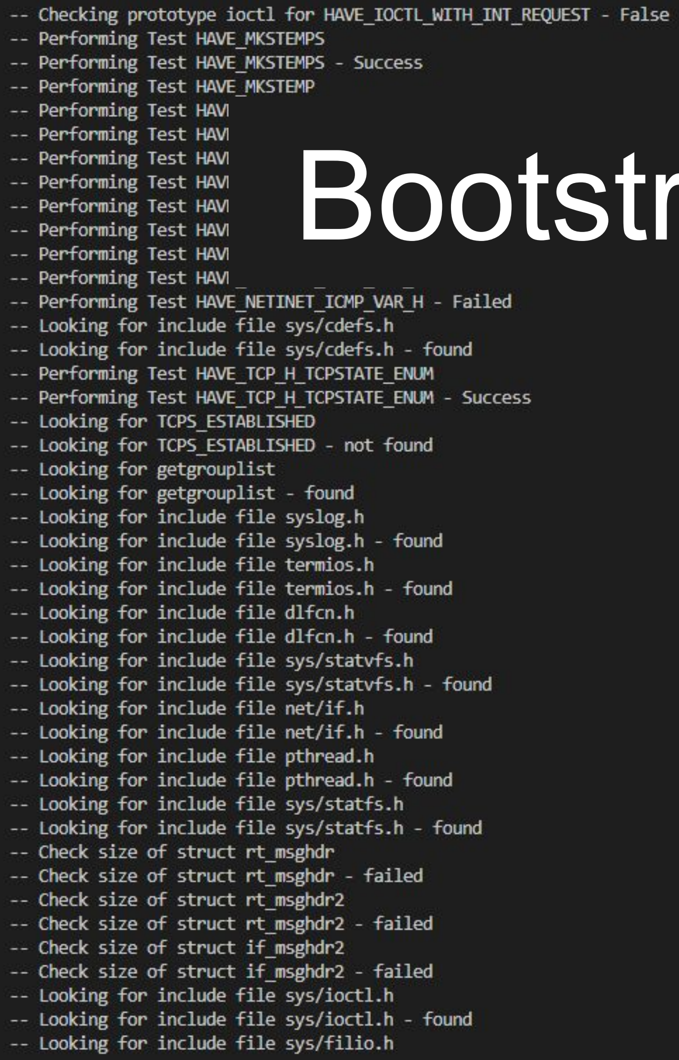
Bootstrapping .NET 8 SDK

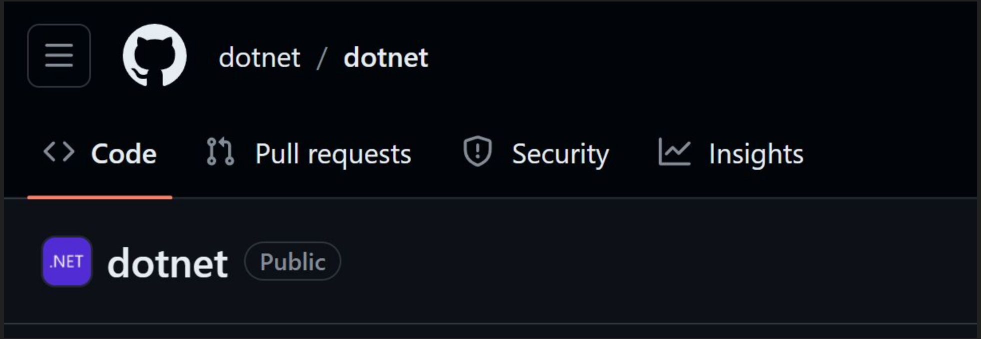
Собираем дотнет из исходников

Руслан Каменский
SpbDotNet 2024



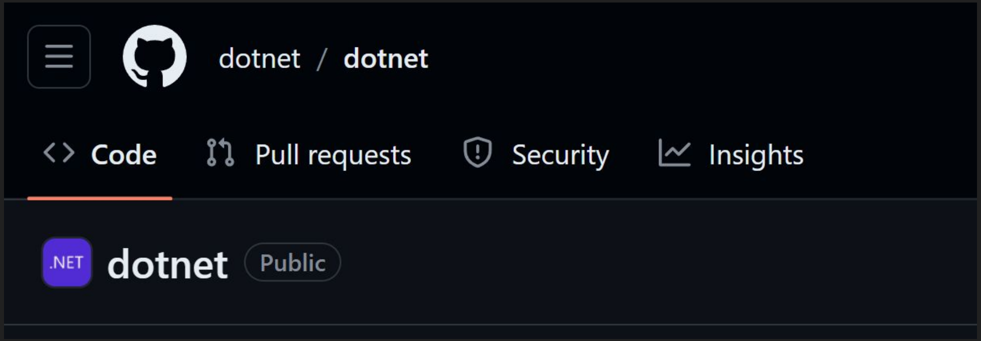
```
-- Checking prototype ioctl for HAVE_IOCTL_WITH_INT_REQUEST - False
-- Performing Test HAVE_MKSTEMPS
-- Performing Test HAVE_MKSTEMPS - Success
-- Performing Test HAVE_MKSTEMP
-- Performing Test HAV
-- Performing Test HAV
-- Performing Test HAV
-- Performing Test HAV
-- Performing Test HAV
-- Performing Test HAV
-- Performing Test HAV
-- Performing Test HAV
-- Performing Test HAVE_NETINET_ICMP_VAR_H - Failed
-- Looking for include file sys/cdefs.h
-- Looking for include file sys/cdefs.h - found
-- Performing Test HAVE_TCP_H_TCPSTATE_ENUM
-- Performing Test HAVE_TCP_H_TCPSTATE_ENUM - Success
-- Looking for TCPS_ESTABLISHED
-- Looking for TCPS_ESTABLISHED - not found
-- Looking for getgrouplist
-- Looking for getgrouplist - found
-- Looking for include file syslog.h
-- Looking for include file syslog.h - found
-- Looking for include file termios.h
-- Looking for include file termios.h - found
-- Looking for include file dlfcn.h
-- Looking for include file dlfcn.h - found
-- Looking for include file sys/statvfs.h
-- Looking for include file sys/statvfs.h - found
-- Looking for include file net/if.h
-- Looking for include file net/if.h - found
-- Looking for include file pthread.h
-- Looking for include file pthread.h - found
-- Looking for include file sys/statfs.h
-- Looking for include file sys/statfs.h - found
-- Check size of struct rt_msghdr
-- Check size of struct rt_msghdr - failed
-- Check size of struct rt_msghdr2
-- Check size of struct rt_msghdr2 - failed
-- Check size of struct if_msghdr2
-- Check size of struct if_msghdr2 - failed
-- Looking for include file sys/ioctl.h
-- Looking for include file sys/ioctl.h - found
-- Looking for include file sys/filio.h
```

Репозиторий



https://github.com/dotnet/dotnet/

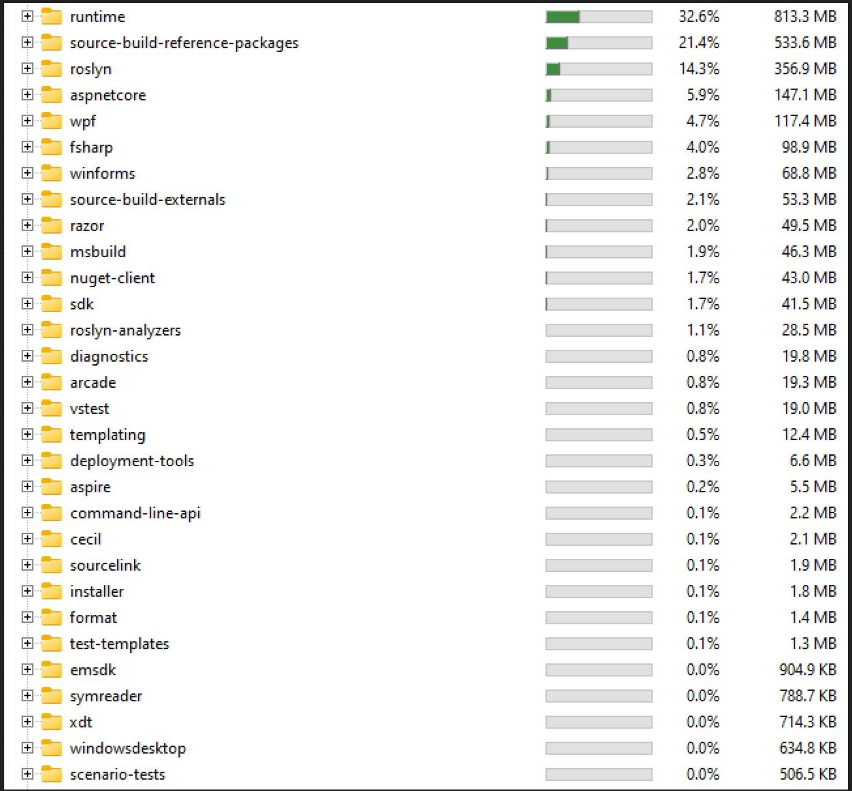
Репозиторий



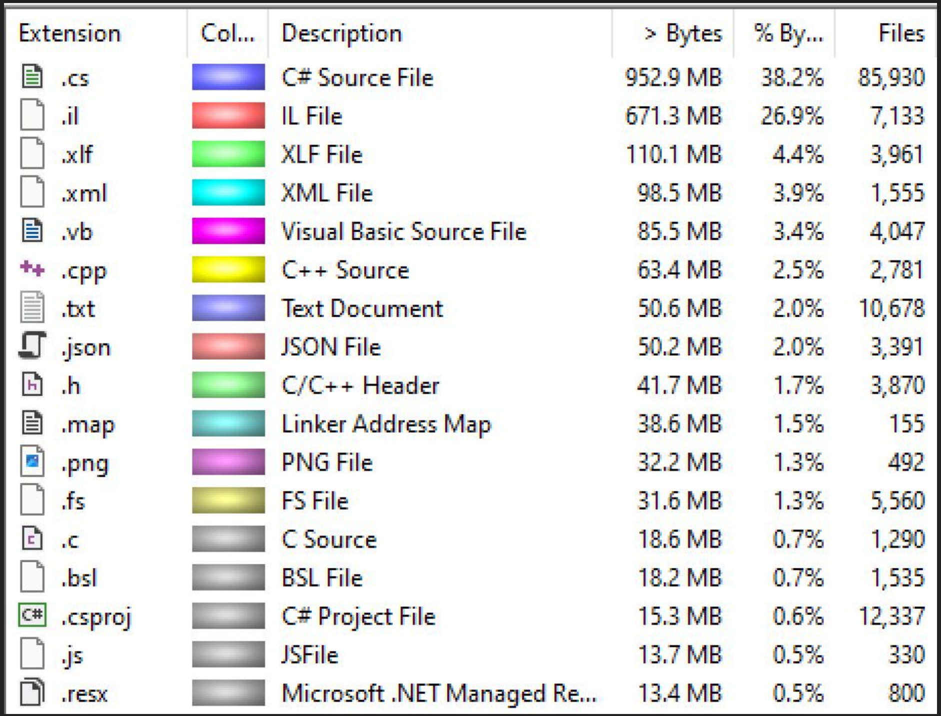
https://github.com/dotnet/dotnet/

- Home of the .NET VMR
- Первый коммит был год назад

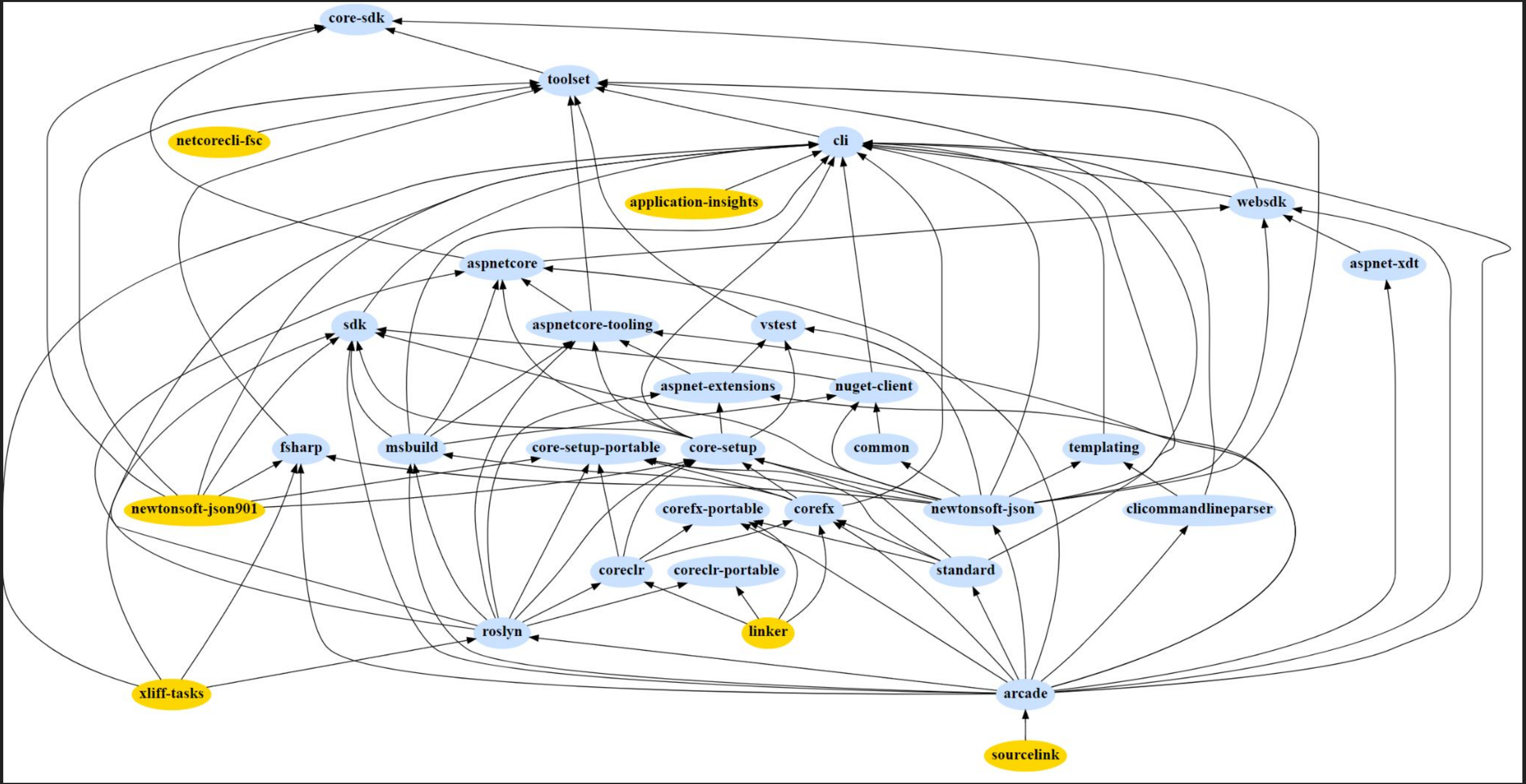
Содержимое репозитория



runtime	32.6%	813.3 MB
source-build-reference-packages	21.4%	533.6 MB
roslyn	14.3%	356.9 MB
aspnetcore	5.9%	147.1 MB
wpf	4.7%	117.4 MB
fsharp	4.0%	98.9 MB
winforms	2.8%	68.8 MB
source-build-externals	2.1%	53.3 MB
razor	2.0%	49.5 MB
msbuild	1.9%	46.3 MB
nuget-client	1.7%	43.0 MB
sdk	1.7%	41.5 MB
roslyn-analyzers	1.1%	28.5 MB
diagnostics	0.8%	19.8 MB
arcade	0.8%	19.3 MB
vstest	0.8%	19.0 MB
templating	0.5%	12.4 MB
deployment-tools	0.3%	6.6 MB
aspire	0.2%	5.5 MB
command-line-api	0.1%	2.2 MB
cecil	0.1%	2.1 MB
sourcelink	0.1%	1.9 MB
installer	0.1%	1.8 MB
format	0.1%	1.4 MB
test-templates	0.1%	1.3 MB
emsdk	0.0%	904.9 KB
symreader	0.0%	788.7 KB
xdt	0.0%	714.3 KB
windowsdesktop	0.0%	634.8 KB
scenario-tests	0.0%	506.5 KB



Extension	Col...	Description	> Bytes	% By...	Files
.cs	🟦	C# Source File	952.9 MB	38.2%	85,930
.il	🟥	IL File	671.3 MB	26.9%	7,133
.xlf	🟩	XLF File	110.1 MB	4.4%	3,961
.xml	🟦	XML File	98.5 MB	3.9%	1,555
.vb	🟪	Visual Basic Source File	85.5 MB	3.4%	4,047
.cpp	🟨	C++ Source	63.4 MB	2.5%	2,781
.txt	🟦	Text Document	50.6 MB	2.0%	10,678
.json	🟥	JSON File	50.2 MB	2.0%	3,391
.h	🟩	C/C++ Header	41.7 MB	1.7%	3,870
.map	🟦	Linker Address Map	38.6 MB	1.5%	155
.png	🟪	PNG File	32.2 MB	1.3%	492
.fs	🟨	FS File	31.6 MB	1.3%	5,560
.c	⬜	C Source	18.6 MB	0.7%	1,290
.bsl	⬜	BSL File	18.2 MB	0.7%	1,535
.csproj	⬜	C# Project File	15.3 MB	0.6%	12,337
.js	⬜	JSFile	13.7 MB	0.5%	330
.resx	⬜	Microsoft .NET Managed Re...	13.4 MB	0.5%	800


core-sdk
toolset
netcorecli-fsc
cli
application-insights
websdk
aspnetcore
aspnet-xdt
sdk
aspnetcore-tooling
vstest
aspnet-extensions
nuget-client
fsharp
msbuild
core-setup-portable
core-setup
common
templating
newtonsoft-json901
corefx-portable
corefx
newtonsoft-json
clicommandlineparser
coreclr
coreclr-portable
standard
roslyn
linker
xliff-tasks
arcade
sourcelink

Как было до .NET 8

https://github.com/dotnet/source-build

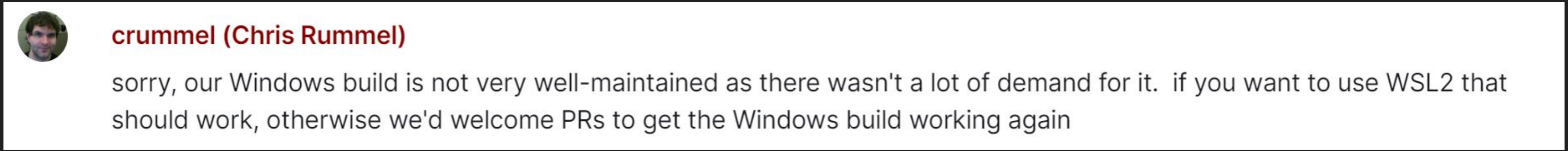
- Документация по сборке
- Toolset
- Скрипт, который делает *git clone* всех репозиториев
- Скрипт сборки

SOURCE-BUILD
~~Why cant~~ you just be normal?
ROSLYN
Screams
MemeCenter.com
imgflip.com

Системные требования

Системные требования

- Не Windows и не MacOS



crummel (Chris Rummel)

sorry, our Windows build is not very well-maintained as there wasn't a lot of demand for it. if you want to use WSL2 that should work, otherwise we'd welcome PRs to get the Windows build working again

Способы сборки

- Linux + Toolchain
- Docker

Сборка

1. Toolchain setup (CMake, llvm, lld, clang, build-essential, python-is-python3, curl, git, lldb, libicu-dev, liblttng-ust-dev, libssl-dev, libkrb5-dev, zlib1g-dev)

2. Clone the VMR
```
git clone https://github.com/dotnet/dotnet
```

3. Prep the source to build on your distro (downloads a .NET SDK and a number of .NET packages)
```
./prep.sh
```

4. Build the .NET SDK
```
./build.sh --clean-while-building --online
```

Кому это может понадобиться?

- Вы мейнтейнеры дистрибутива GNU/Linux
- Вы хотите запустить дотнет на не поддерживаемой архитектуре
- Вы хотите управлять атомной электростанцией на dotnet
- Вы хотите запатчить dotnet

Кому это может понадобиться?

- **Вы мейнтейнеры дистрибутива GNU/Linux**
- Вы хотите запустить дотнет на не поддерживаемой архитектуре
- Вы хотите управлять атомной электростанцией на dotnet
- Вы хотите запатчить dotnet

Ожидания от пакета с открытым исходным кодом

- Пакет должен иметь полный исходный код всего
- Пакет должен собираться без доступа в интернет на одной машине
- Пакет должен иметь "Consistent reproducibility", то есть сборка должна быть воспроизводима

Ожидания от пакета с открытым исходным кодом

- Пакет должен иметь полный исходный код всего
- Пакет должен собираться без доступа в интернет на одной машине
- Пакет должен иметь "Consistent reproducibility", то есть сборка должна быть воспроизводима

Курица или яйцо?

Bootstrapping

- Downloading source-built artifacts

  https://dotnetcli.azureedge.net/source-built-artifacts/assets/Private.SourceBuilt.$archiveType.$archiveVersion.$archiveRid.tar.gz

- Run restore on project to initiate download of bootstrap packages

Кому это может понадобиться?

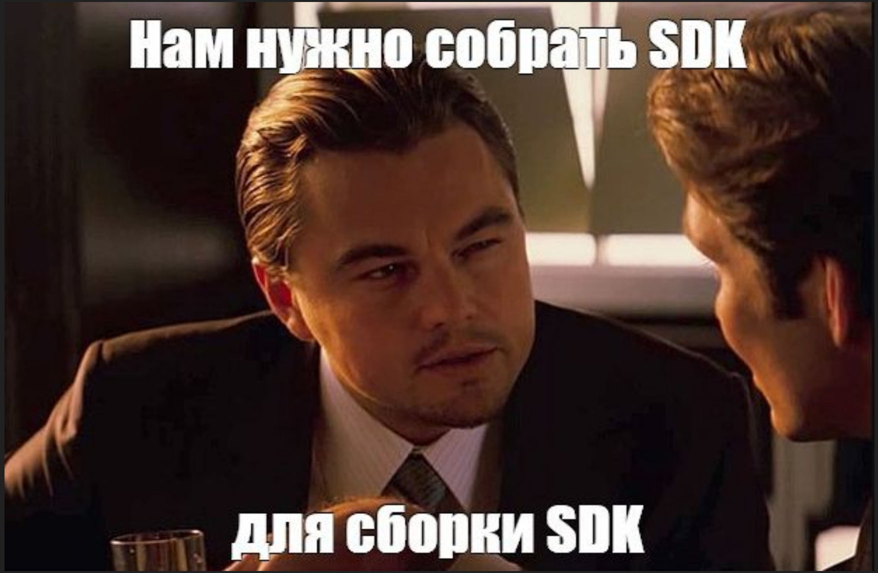
- Вы мейнтейнеры дистрибутива GNU/Linux
- **Вы хотите запустить дотнет на неподдерживаемой архитектуре**
- Вы хотите управлять атомной электростанцией на dotnet
- Вы хотите запатчить dotnet

Bootstrapping Scenarios

- Building for New OS (Using a RID unknown to .NET)
- Building for New Architecture (Using a RID unknown to .NET)

Bootstrapping Scenarios

- Building for New OS (Using a RID unknown to .NET)
- Building for New Architecture (Using a RID unknown to .NET)



Bootstrapping Scenarios

- Building for New OS (Using a RID unknown to .NET)
    - Stage 0:
      1. Get Microsoft portable SDK.
      2. Update the RID graph (runtime.json) in the Microsoft-built portable SDK
    - Stage 1:
      1. Update the RID graph in source with the same changes made in Stage 0
      2. Build with Stage 0 SDK using --with-sdk with your modified portable SDK
    - Stage 2:
      1. Now you have a RID-specific SDK that knows about your new RID, build with Stage 1 SDK

- Building for New Architecture (Using a RID unknown to .NET)

Bootstrapping Scenarios

- Building for New OS (Using a RID unknown to .NET)

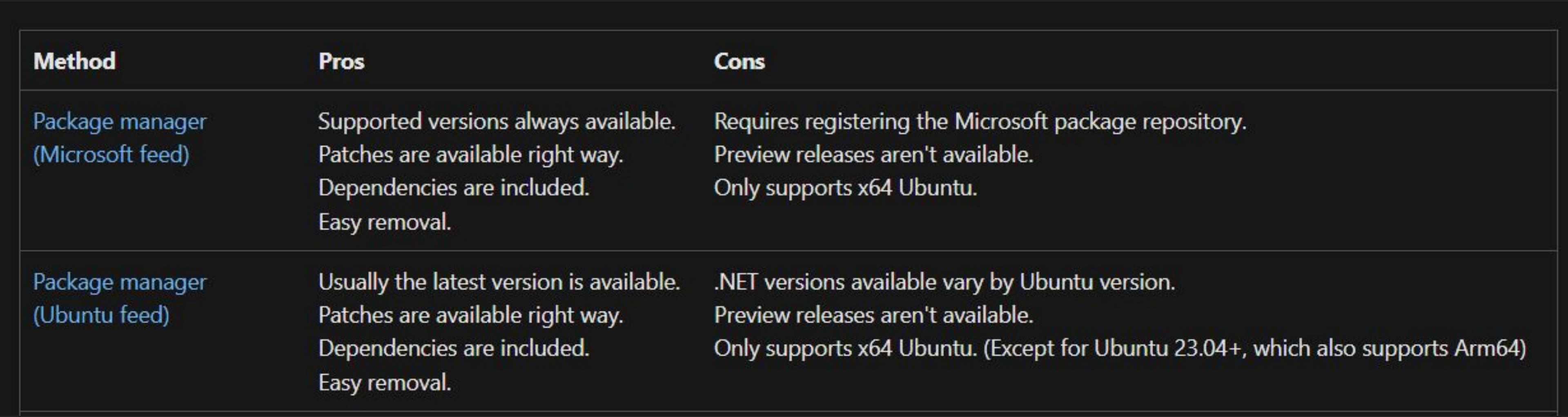
- <u>Building for New Architecture (Using a RID unknown to .NET)</u>
  - Stage 0:
    1. Cross compile an SDK on x64 for target platform
    2. Cross compile the runtime repo (on x64 for target platform, generally done as part of previous step) and save the nuget packages, use these to augment the Microsoft-built previously-source-built archive.
  - Stage 1:
    1. Use the cross-compiled SDK and augmented previously-source-built-archive to build a stage 1 SDK
  - Stage 2:
    1. Use your stage 1 SDK to build a stage 2 SDK, pointing it to the SDK and previously-source-built archives from stage 1

Кому это может понадобиться?

- Вы мейнтейнеры дистрибутива GNU/Linux
- Вы хотите запустить дотнет на не поддерживаемой архитектуре
- **Вы хотите управлять атомной электростанцией на dotnet**
- Вы хотите запатчить dotnet

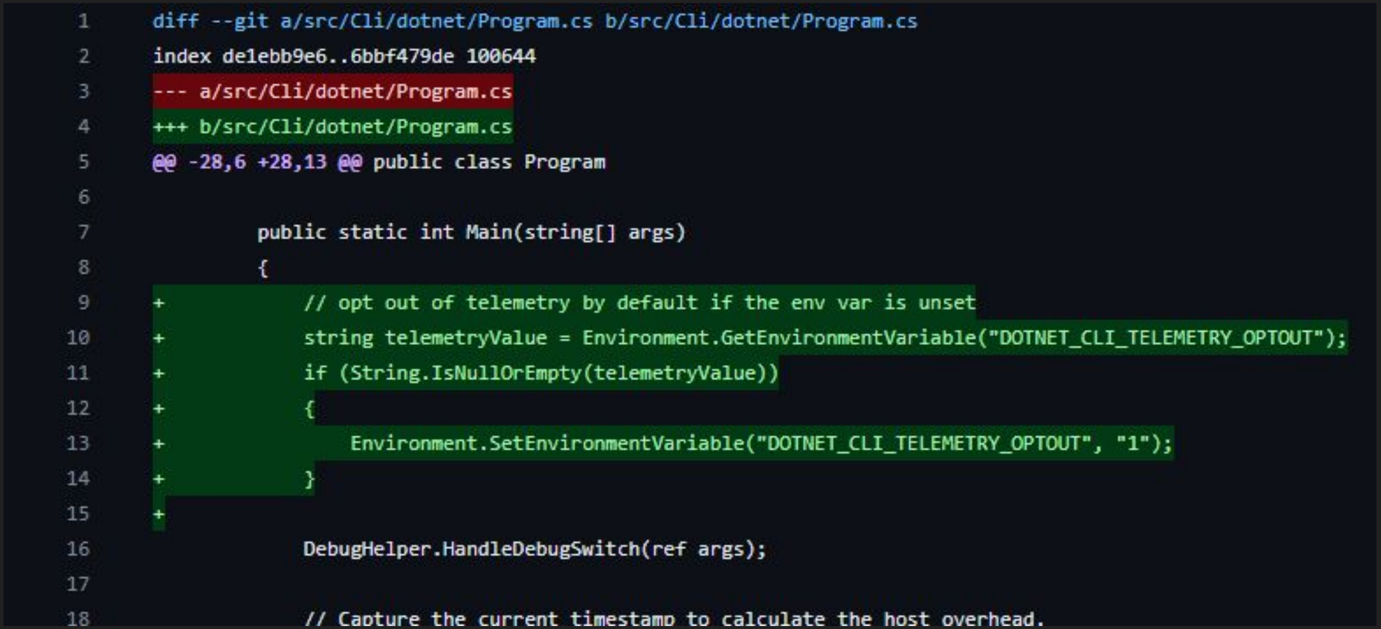
Кому это может понадобиться?

- **Вы мейнтейнеры дистрибутива GNU/Linux**
- Вы хотите запустить дотнет на не поддерживаемой архитектуре
- Вы хотите управлять атомной электростанцией на dotnet
- **Вы хотите запатчить dotnet**



Method	Pros	Cons
Package manager (Microsoft feed)	Supported versions always available. Patches are available right way. Dependencies are included. Easy removal.	Requires registering the Microsoft package repository. Preview releases aren't available. Only supports x64 Ubuntu.
Package manager (Ubuntu feed)	Usually the latest version is available. Patches are available right way. Dependencies are included. Easy removal.	.NET versions available vary by Ubuntu version. Preview releases aren't available. Only supports x64 Ubuntu. (Except for Ubuntu 23.04+, which also supports Arm64)

DotNet patch



```
diff --git a/src/Cli/dotnet/Program.cs b/src/Cli/dotnet/Program.cs
index de1ebb9e6..6bbf479de 100644
--- a/src/Cli/dotnet/Program.cs
+++ b/src/Cli/dotnet/Program.cs
@@ -28,6 +28,13 @@ public class Program

         public static int Main(string[] args)
         {
+            // opt out of telemetry by default if the env var is unset
+            string telemetryValue = Environment.GetEnvironmentVariable("DOTNET_CLI_TELEMETRY_OPTOUT");
+            if (String.IsNullOrEmpty(telemetryValue))
+            {
+                Environment.SetEnvironmentVariable("DOTNET_CLI_TELEMETRY_OPTOUT", "1");
+            }
+
             DebugHelper.HandleDebugSwitch(ref args);

             // Capture the current timestamp to calculate the host overhead.
```

.NET 8 Milestones

- Deliver .NET source-build to Linux partners via a “VMR-lite” repo (MVP).
- Improve current source-build infrastructure to support sustainability and reduce cost.
- Redesign .NET’s build to reduce complexity and align it with “vertical” requirements for source-build.
- Design and create E2E testing against installed products. Prioritize creation of tests that benefit our source-build partners first.
- Enable an experimental macOS source-build variant.
- Enable Linux portable source-build.

.NET 9 Milestones

- Enable .NET repo tests to run against full source-build.
- Build infrastructure to support full VMR source-code flow (forward and backward to individual repos).
- Expand source-build to support Windows and macOS (officially).
- Turn off existing official build.