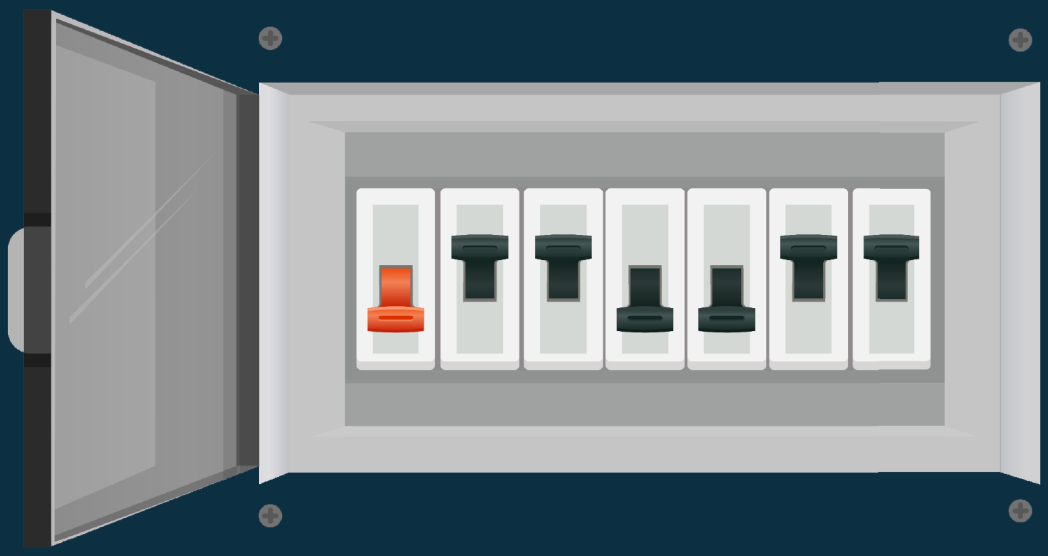
Electron Security

Fuse



NOP Team

Fuse | Electron 安全

0x01 简介

大家好，今天和大家讨论的是 fuse ， fuse 直译过来是保险丝，官方文档中翻译为包特性切换

`Electron` 开发的应用有很多特性，能够为一些场景提供帮助，但并不是所有的场景都会用到这些特性，因此对于普通开发者来说，你默认给我开发的程序带了一堆特性，我可能还用不到，甚至可能还不太安全，我是不是应该有禁用的选项，例如，99%的应用都没有使用 `ELECTRON_RUN_AS_NODE` ，开发者希望能够提供无法使用该功能的二进制文件。

这就是 fuse

0x02 当前可用的 fuse

fuse 还在随着版本不断增加，这篇文章只讨论目前(2024-05-10)的情况

fuse	作用	默认状态
`runAsNode`	`runAsNode` 是否考虑 `ELECTRON_RUN_AS_NODE` 环境变量。请注意，如果禁用此 `fuse` ，则主进程中的 `process.fork` 将无法按预期运行，因为它依赖于此环境变量来运行	`Enabled`
`cookieEncryption`	`cookieEncryption` 磁盘上的 `cookie` 存储是否使用操作系统级别的加密密钥进行加密。默认情况下， `Chromium` 用于存储 `cookie` 的 `sqlite` 数据库以明文形式存储值。如果您希望确保您的应用程序 `cookie` 以与 `Chrome` 相同	`Disabled`

	的方式加密，则应启用此 `fuse`	
`nodeOptions`	`nodeOptions` 是否考虑 `NODE_OPTIONS` 和 `NODE_EXTRA_CA_CERTS` 环境变量。此环境变量可用于将各种自定义选项传递到 `Node.js` 运行时，并且通常不被生产中的应用程序使用。大多数应用程序可以安全地禁用此 fuse	`Enabled`
`nodeCliInspect`	`nodeCliInspect` 是否遵守 `--inspect`、`--inspect-brk` 等标志。禁用时，它还确保 `SIGUSR1` 信号不会初始化主进程检查器。大多数应用程序可以安全地禁用此fuse。	`Enabled`
`embeddedAsarIntegrityValidation`	`embeddedAsarIntegrityValidation` 是 `macOS` 上的一项实验性功能，该功能在加载 `app.asar` 文件时验证其内容。此功能旨在将性能影响降至最低，但可能会略微降低从 `app.asar` 存档中读取文件的速度	`Disabled`
`onlyLoadAppFromAsar`	`onlyLoadAppFromAsar` 改变了 `Electron` 用来定位应用程序代码的搜索系统。 默认情况下，`Electron` 将按照以下顺序搜索 `app.asar -> app -> default_app.asar`。当这个fuse 被启用时，搜索顺	`Disabled`

	序变成了一个单一条目的 `app.asar`，从而确保当与 `embeddedAsarIntegrityValidation fuse` 结合使用时，不可能加载未经验证的代码。	
`loadBrowserProcessSpecificV8Snapshot`	`loadBrowserProcessSpecificV8Snapshot` 更改浏览器进程使用的 `V8` 快照文件。 默认情况下，`Electron` 的进程都将使用相同的 `V8` 快照文件。启用此fuse后，浏览器进程将使用名为 `browser_v8_context_snapshot.bin` 的文件作为其 `V8` 快照。其他进程将使用它们通常使用的 `V8` 快照文件	`Disabled`
`grantFileProtocolExtraPrivileges`	`grantFileProtocolExtraPrivileges` 从 `file://` 协议加载的页面是否被赋予超出它们在传统 `Web` 浏览器中所获得的权限的权限。在 `Electron` 的原始版本中，这种行为是 `Electron` 应用程序的核心，但不再需要，因为应用程序现在应该从自定义协议中提供本地文件。如果您不从 `file://` 中提供页面，则应禁用此fuse	`Enabled`

但是经过我的实际测试，发现 `Electron Forge` ，也就是官方推荐的打包工具默认的 Fuse 配置如下

forge.config.js

```
const { FusesPlugin } = require('@electron-forge/plugin-fuses');
const { FuseV1Options, FuseVersion } = require('@electron/fuses');

module.exports = {
  packagerConfig: {
    asar: true,
  },
  rebuildConfig: {},
  makers: [
    {
      name: '@electron-forge/maker-squirrel',
      config: {},
    },
    {
      name: '@electron-forge/maker-zip',
      platforms: ['darwin'],
    },
    {
      name: '@electron-forge/maker-deb',
      config: {},
    },
    {
      name: '@electron-forge/maker-rpm',
      config: {},
    },
  ],
  plugins: [
    {
      name: '@electron-forge/plugin-auto-unpack-natives',
      config: {},
    },
    // Fuses are used to enable/disable various Electron functionality
    // at package time, before code signing the application
    new FusesPlugin({
      version: FuseVersion.V1,
      [FuseV1Options.RunAsNode]: false,
      [FuseV1Options.EnableCookieEncryption]: true,
```

```
      [FuseV1Options.EnableNodeOptionsEnvironmentVariable]: false,
      [FuseV1Options.EnableNodeCliInspectArguments]: false,
      [FuseV1Options.EnableEmbeddedAsarIntegrityValidation]: true,
      [FuseV1Options.OnlyLoadAppFromAsar]: true,
    }),
  ],
};
```

可以看到，除了 `loadBrowserProcessSpecificV8Snapshot` 和 `grantFileProtocolExtraPrivileges` 其他的与官网标记的默认值完全相反，我们看一下实际打包出来的程序

```
➜  x64 git:(master) ✗ which sophisticated-bell-slow-t246g
/usr/bin/sophisticated-bell-slow-t246g
➜  x64 git:(master) ✗ npx @electron/fuses read --app /usr/bin/sophisticated-bell-slow-t246g
Analyzing app: sophisticated-bell-slow-t246g
Fuse Version: v1
  RunAsNode is Disabled
  EnableCookieEncryption is Enabled
  EnableNodeOptionsEnvironmentVariable is Disabled
  EnableNodeCliInspectArguments is Disabled
  EnableEmbeddedAsarIntegrityValidation is Enabled
  OnlyLoadAppFromAsar is Enabled
  LoadBrowserProcessSpecificV8Snapshot is Disabled
  GrantFileProtocolExtraPrivileges is Enabled
➜  x64 git:(master) ✗
```

和上面的配置一致

所以你说官方设置默认值不太符合默认即安全吧，它打包工具里给你自动重新设置了值，你说他默认即安全吧，还没有把安全的值设置为默认，奇奇怪怪

0x03 如何查看程序的 fuse

检查一个应用程序的 fuse 设置

> https://www.electronjs.org/zh/docs/latest/tutorial/fuses#how-do-i-flip-the-fuses

需要安装 `@electron/fuses` 依赖包

```
npm i @electron/fuses
```

检查应用程序的 fuse

```
npx @electron/fuses read --app /Applications/Foo.app
```

```
~ ❯❯❯ npx @electron/fuses read --app /Applications/Goby.app
Analyzing app: Goby.app
Fuse Version: v1
  RunAsNode is Enabled
  EnableCookieEncryption is Disabled
  EnableNodeOptionsEnvironmentVariable is Enabled
  EnableNodeCliInspectArguments is Enabled
  EnableEmbeddedAsarIntegrityValidation is Disabled
  OnlyLoadAppFromAsar is Disabled
~ ❯❯❯
```

0x04 特性可能带来的危害

现在的情况是官方比较幽默，fuse 的默认值设置的像是安全在为功能让步，但打包工具又反转过来，当然我们作为安全研究人员更希望向默认即安全的建设方向去走

我们接下来就看一下这些特性可能带来的危害

1. runAsNode

`runAsNode` 特性的含义是程序当作普通的 `Node.js` 进程启动，如果是普通的 `Node.js` ，那么可以给该程序传递很多启动参数，官方的文档说是否考虑 `ELECTRON_RUN_AS_NODE` 环境变量

> ELECTRON_RUN_AS_NODE 参考如下文档
>
> https://www.electronjs.org/zh/docs/latest/api/environment-variables#electron_run_as_node

文档中说默认情况下，除了以下标志，标准的 `cli` 选项传递给程序都会生效

- `--openssl-config`
- `--use-bundled-ca`
- `--use-openssl-ca`
- `--force-fips`
- `--enable-fips`

这些标志无效是因为 Electron 在构建 Node.js 的 `crypto` 模块时使用 BoringSSL 而不是 OpenSSL

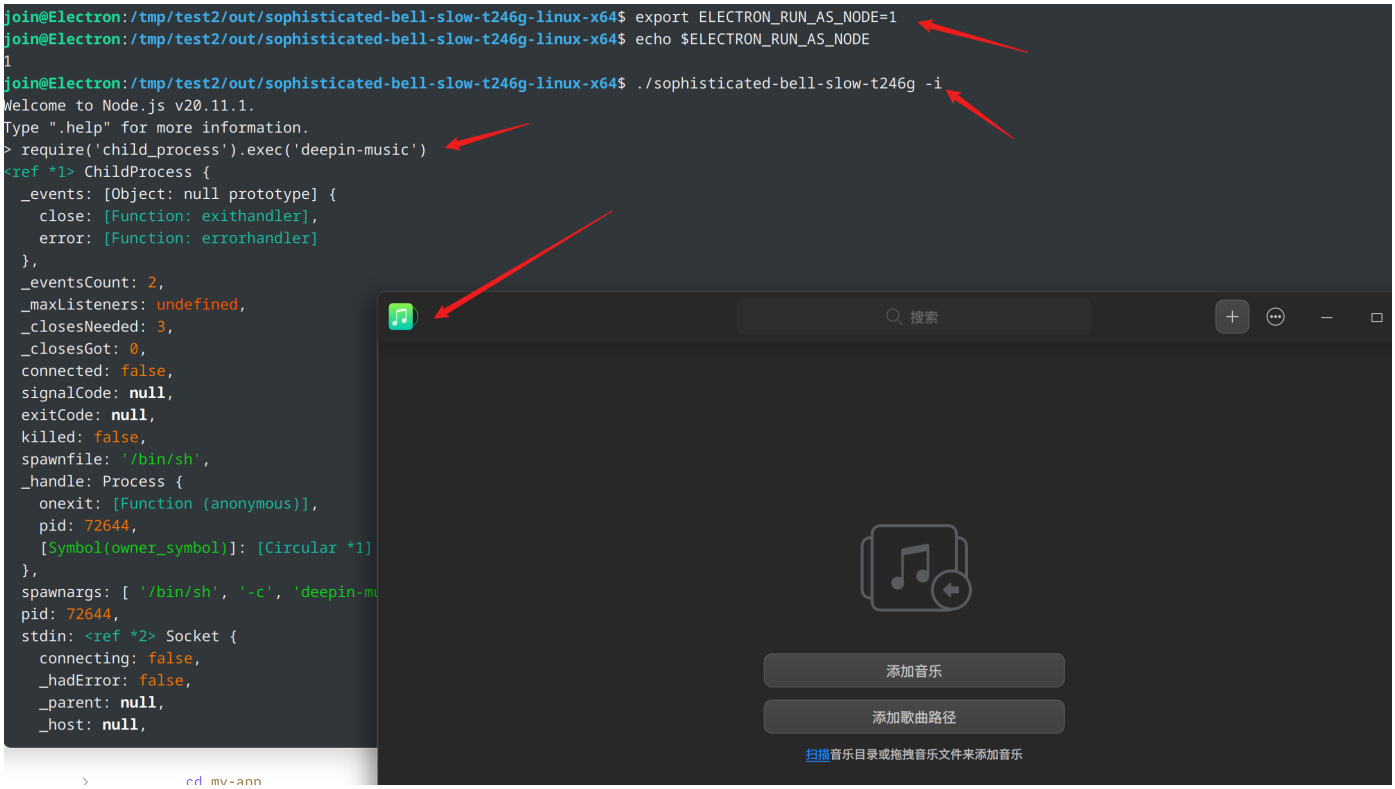
> `cli` 选项可以参考
>
> https://nodejs.org/api/cli.html

现在我编译一个 `runAsNode` 为 `Enabled` 的程序

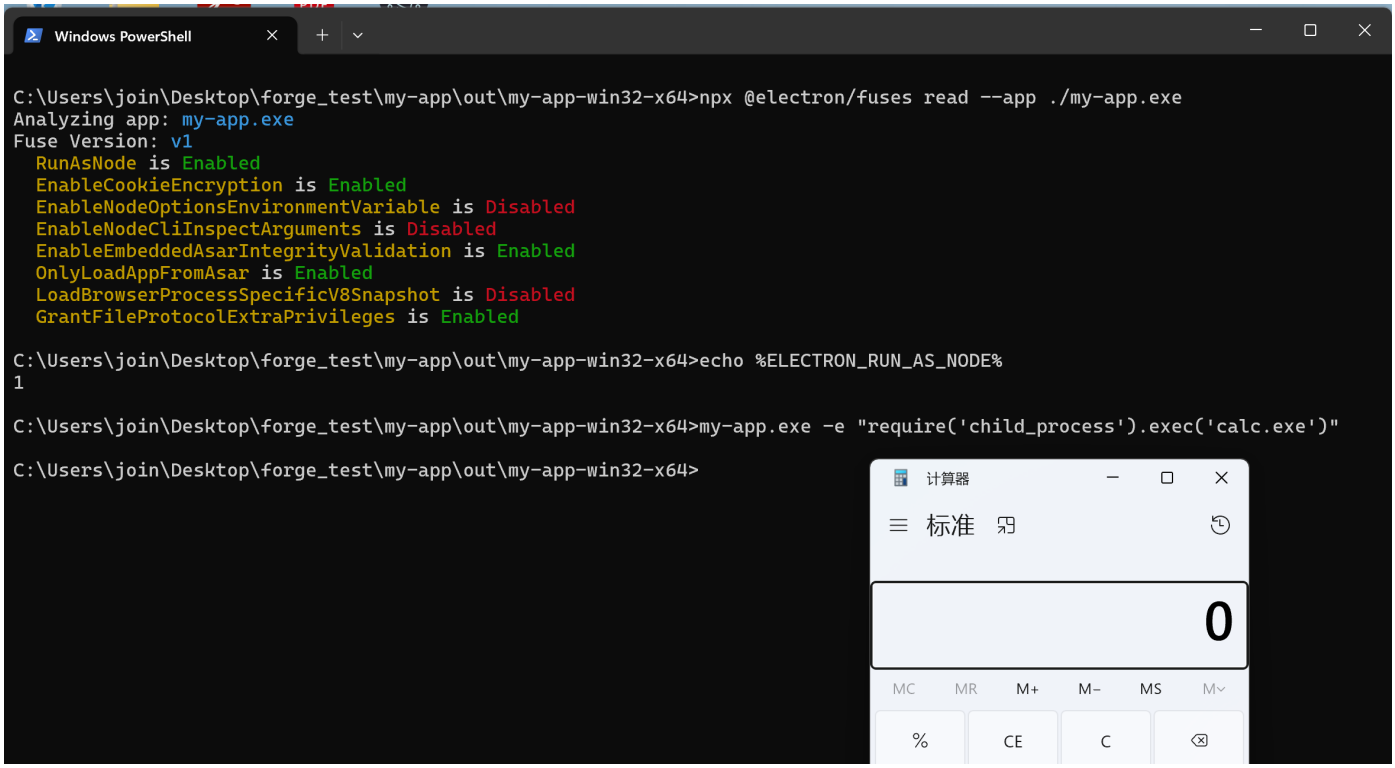
```
➜  test2 git:(master) ✗ proxychains npm run make
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.15
[proxychains] DLL init: proxychains-ng 4.15

> sophisticated-bell-slow-t246g@1.0.0 make
> electron-forge make

[proxychains] DLL init: proxychains-ng 4.15
[proxychains] DLL init: proxychains-ng 4.15
[proxychains] DLL init: proxychains-ng 4.15
✔ Checking your system
[proxychains] DLL init: proxychains-ng 4.15
✔ Loading configuration
✔ Resolving make targets
  › Making for the following targets:
✔ Running package command
  ✔ Preparing to package application
  ✔ Running packaging hooks
    ✔ Running generateAssets hook
    ✔ Running prePackage hook
  ✔ Packaging application
    ✔ Packaging for x64 on linux [3s]
  ✔ Running postPackage hook
✔ Running preMake hook
✔ Making distributables
  ✔ Making a deb distributable for linux/x64 [34s]
✔ Running postMake hook
  › Artifacts available at: /tmp/test2/out/make

› [proxychains] DLL init: proxychains-ng 4.15
➜  test2 git:(master) ✗ cd out/
```

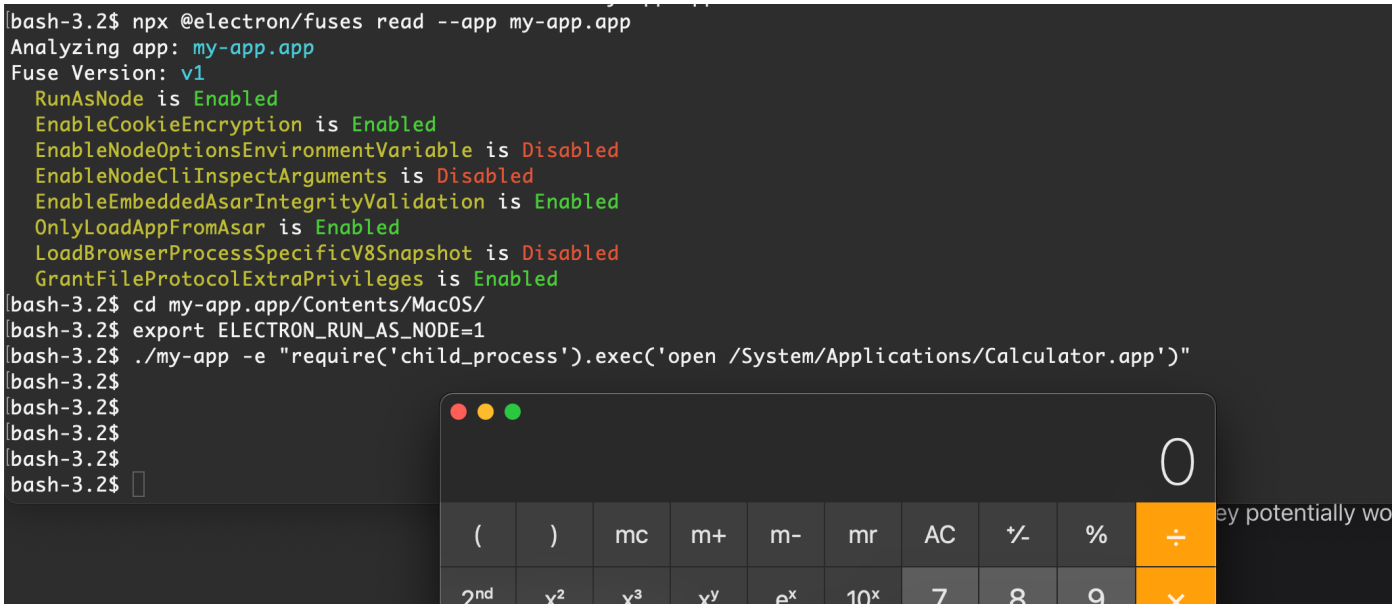
```
➜  sophisticated-bell-slow-t246g-linux-x64 git:(master) ✗ npx @electron/fuses read --app ./sophisticated-bell-slow-t246g
Analyzing app: sophisticated-bell-slow-t246g
Fuse Version: v1
  RunAsNode is Enabled
  EnableCookieEncryption is Enabled
  EnableNodeOptionsEnvironmentVariable is Disabled
  EnableNodeCliInspectArguments is Disabled
  EnableEmbeddedAsarIntegrityValidation is Enabled
  OnlyLoadAppFromAsar is Enabled
  LoadBrowserProcessSpecificV8Snapshot is Disabled
  GrantFileProtocolExtraPrivileges is Enabled
```

尝试通过设置环境变量 `ELECTRON_RUN_AS_NODE=1` 并传递 `-i` 参数


join@Electron:/tmp/test2/out/sophisticated-bell-slow-t246g-linux-x64$ export ELECTRON_RUN_AS_NODE=1
join@Electron:/tmp/test2/out/sophisticated-bell-slow-t246g-linux-x64$ echo $ELECTRON_RUN_AS_NODE
1
join@Electron:/tmp/test2/out/sophisticated-bell-slow-t246g-linux-x64$ ./sophisticated-bell-slow-t246g -i
Welcome to Node.js v20.11.1.
Type ".help" for more information.
> require('child_process').exec('deepin-music')
搜索
添加音乐
添加歌曲路径
扫描音乐目录或拖拽音乐文件来添加音乐

成功实现本地命令执行


Windows PowerShell
C:\Users\join\Desktop\forge_test\my-app\out\my-app-win32-x64>npx @electron/fuses read --app ./my-app.exe
Analyzing app: my-app.exe
Fuse Version: v1
RunAsNode is Enabled
EnableCookieEncryption is Enabled
EnableNodeOptionsEnvironmentVariable is Disabled
EnableNodeCliInspectArguments is Disabled
EnableEmbeddedAsarIntegrityValidation is Enabled
OnlyLoadAppFromAsar is Enabled
LoadBrowserProcessSpecificV8Snapshot is Disabled
GrantFileProtocolExtraPrivileges is Enabled
C:\Users\join\Desktop\forge_test\my-app\out\my-app-win32-x64>echo %ELECTRON_RUN_AS_NODE%
1
C:\Users\join\Desktop\forge_test\my-app\out\my-app-win32-x64>my-app.exe -e "require('child_process').exec('calc.exe')"
C:\Users\join\Desktop\forge_test\my-app\out\my-app-win32-x64>
计算器
标准
0
MC MR M+ M- MS M
% CE C

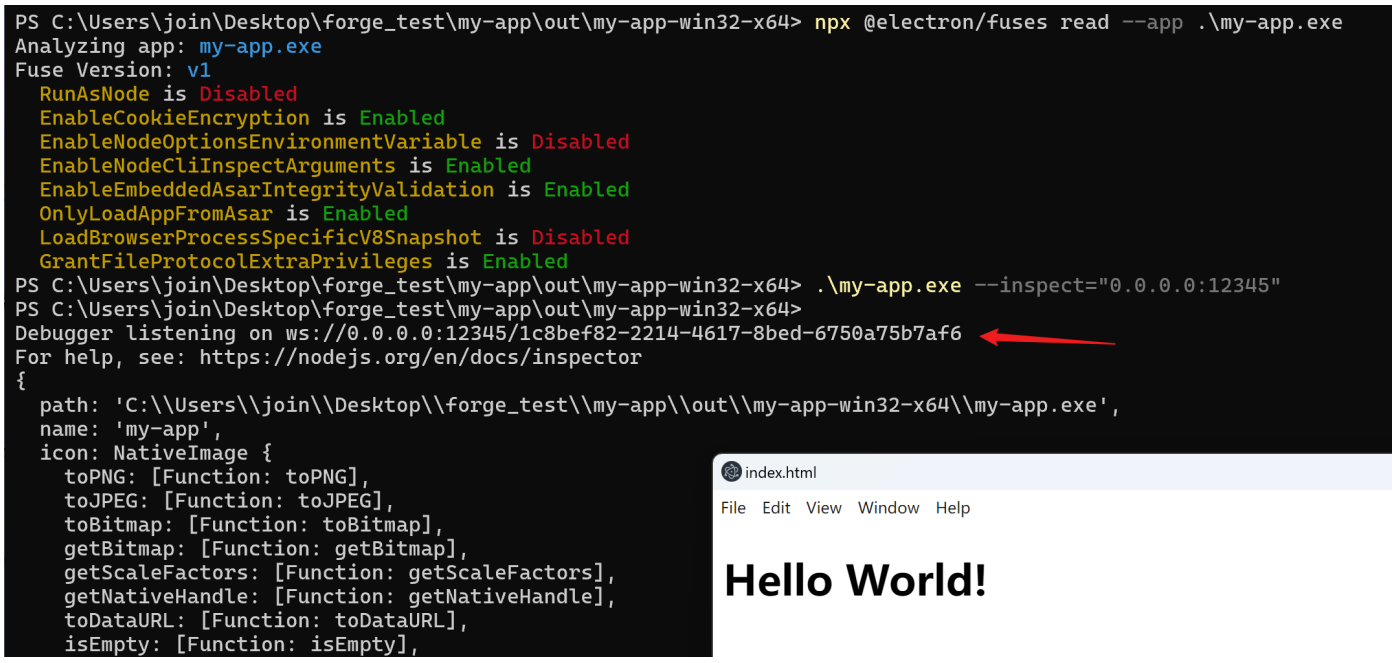


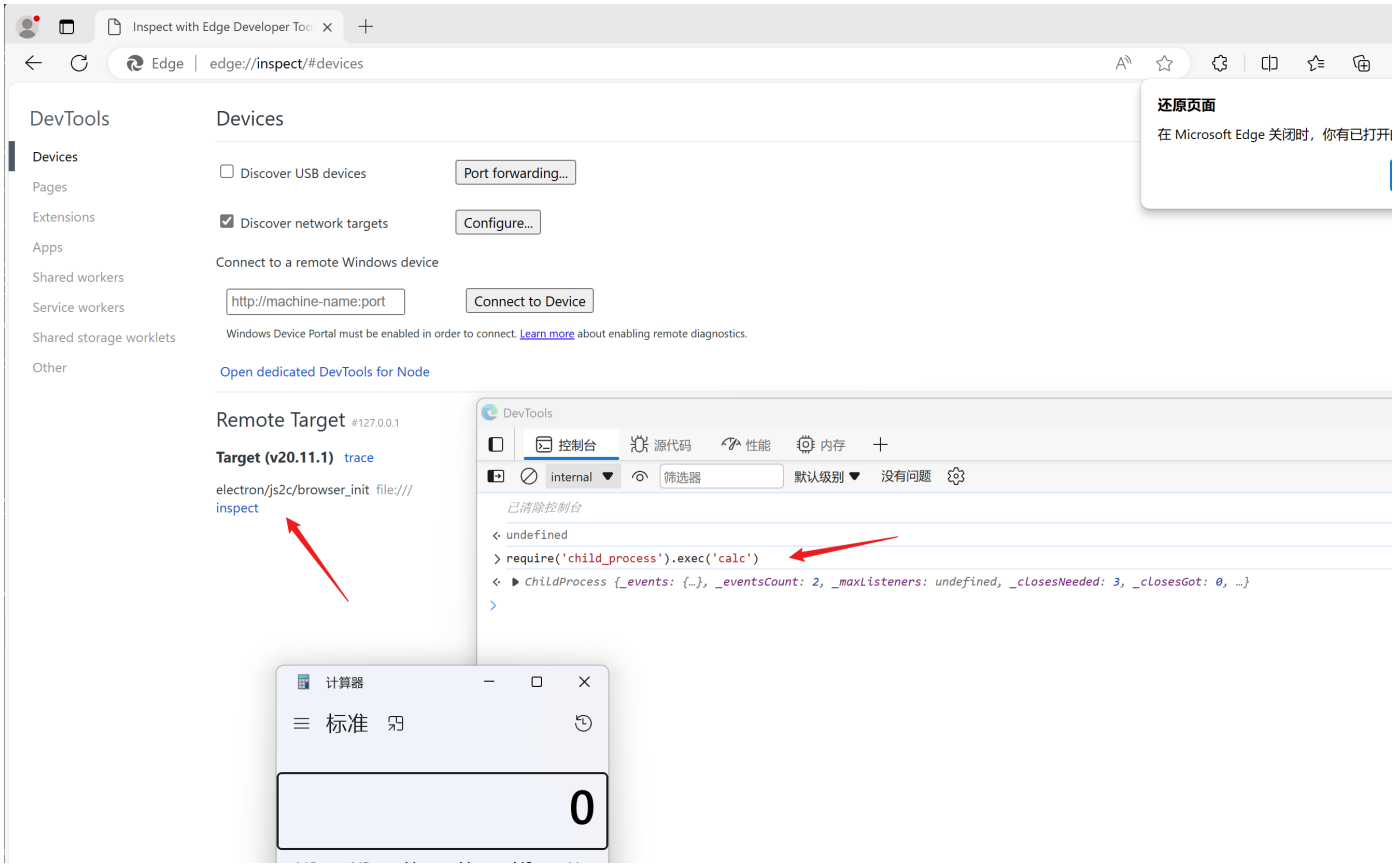
Windows 和 MacOS 也成功执行系统命令

2. nodeCliInspect

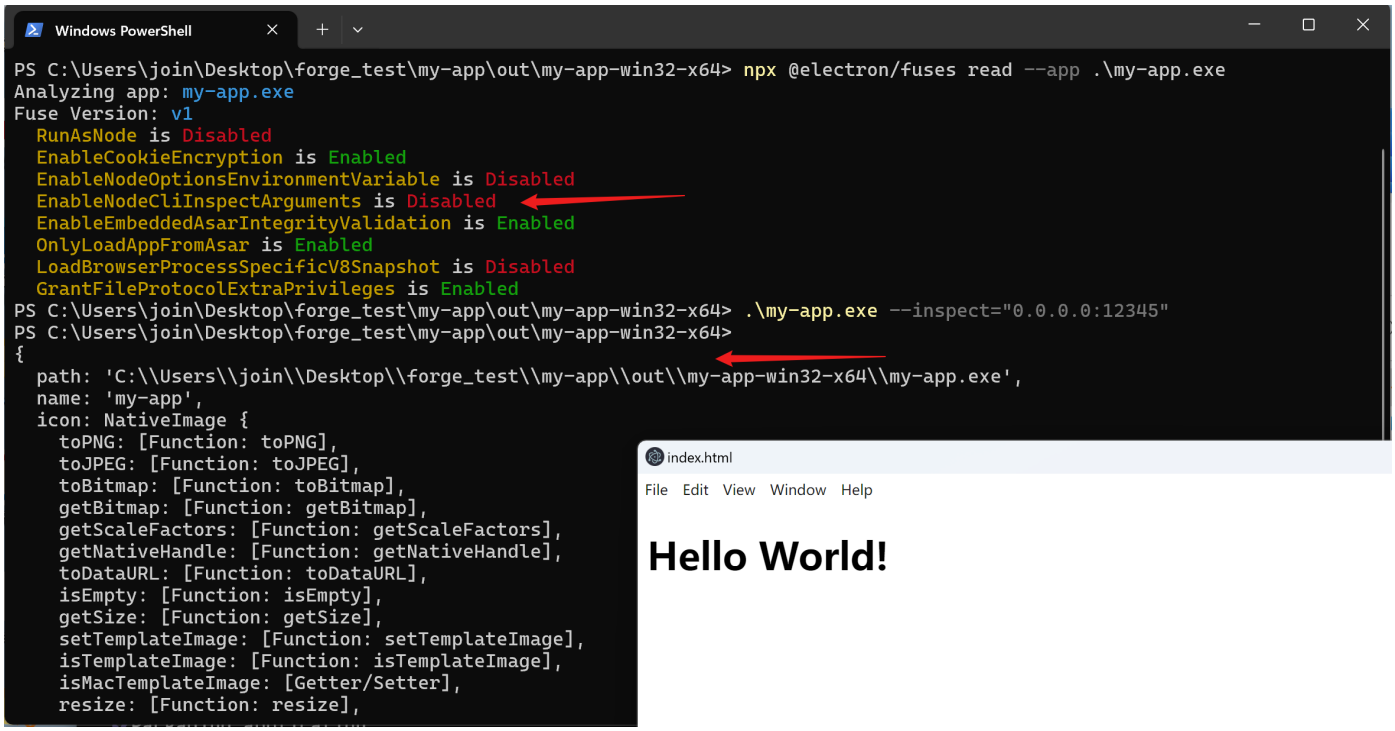
这个 fuse 就是之前的 远程调试的利用 文章了，这个fuse 决定是否可以进行远程调试

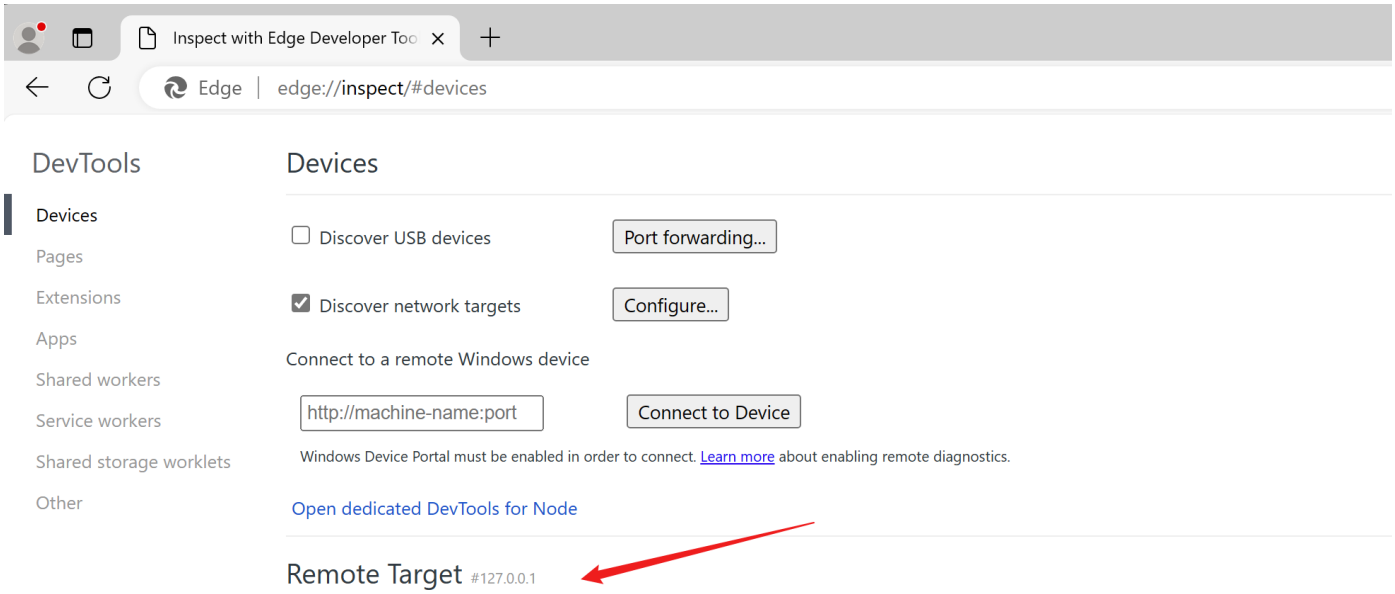
如果设置允许远程调试，情况如下





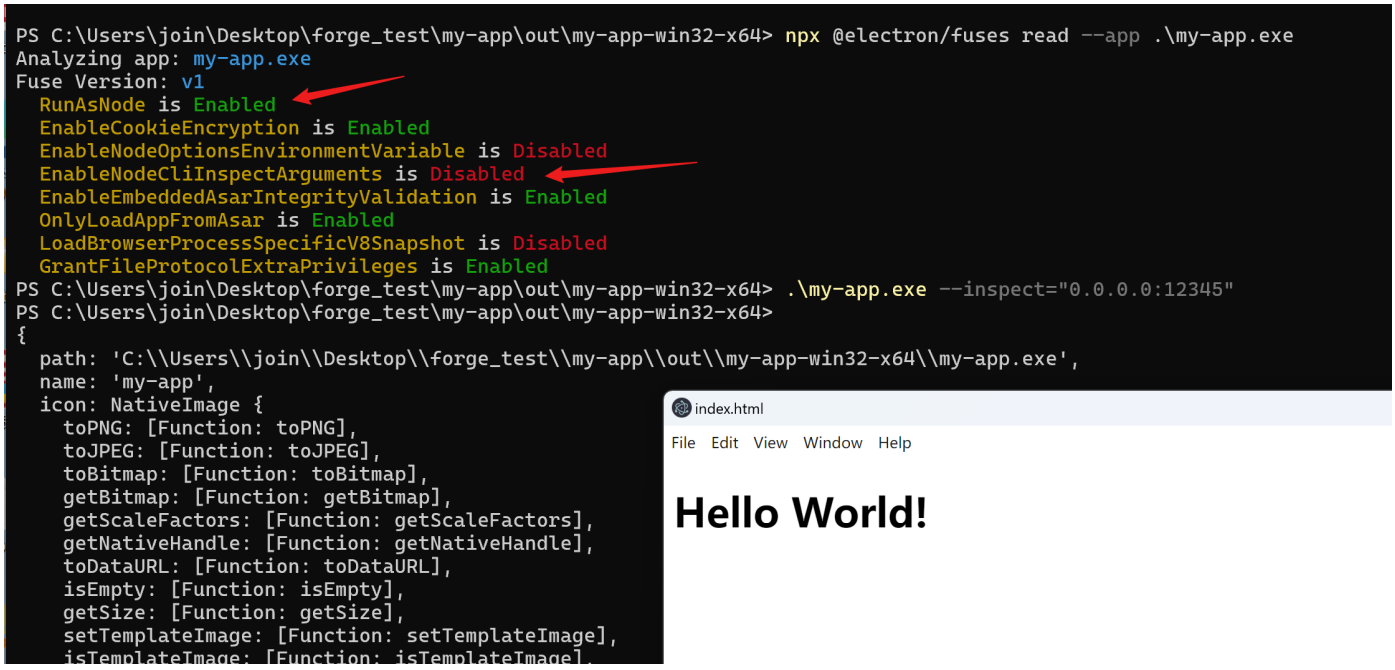
如果设置不允许远程调试，则情况如下

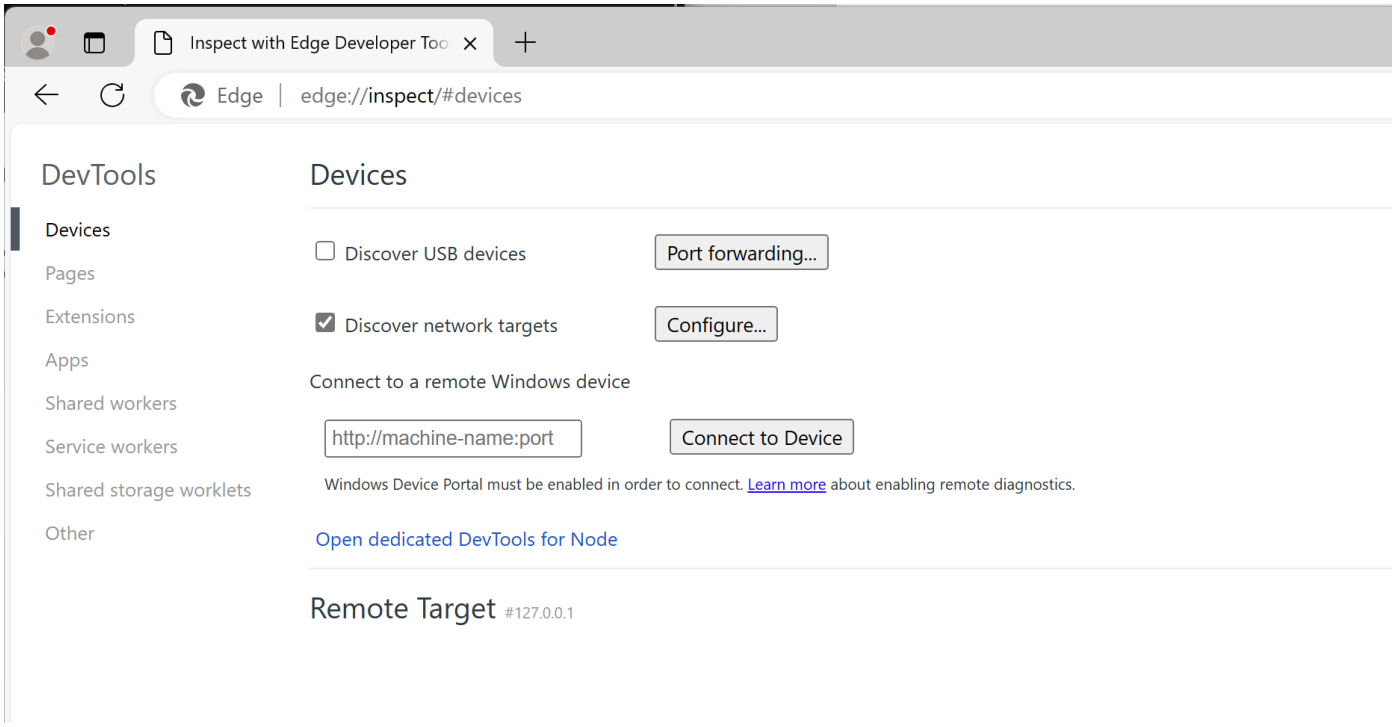




远程调试无法启动

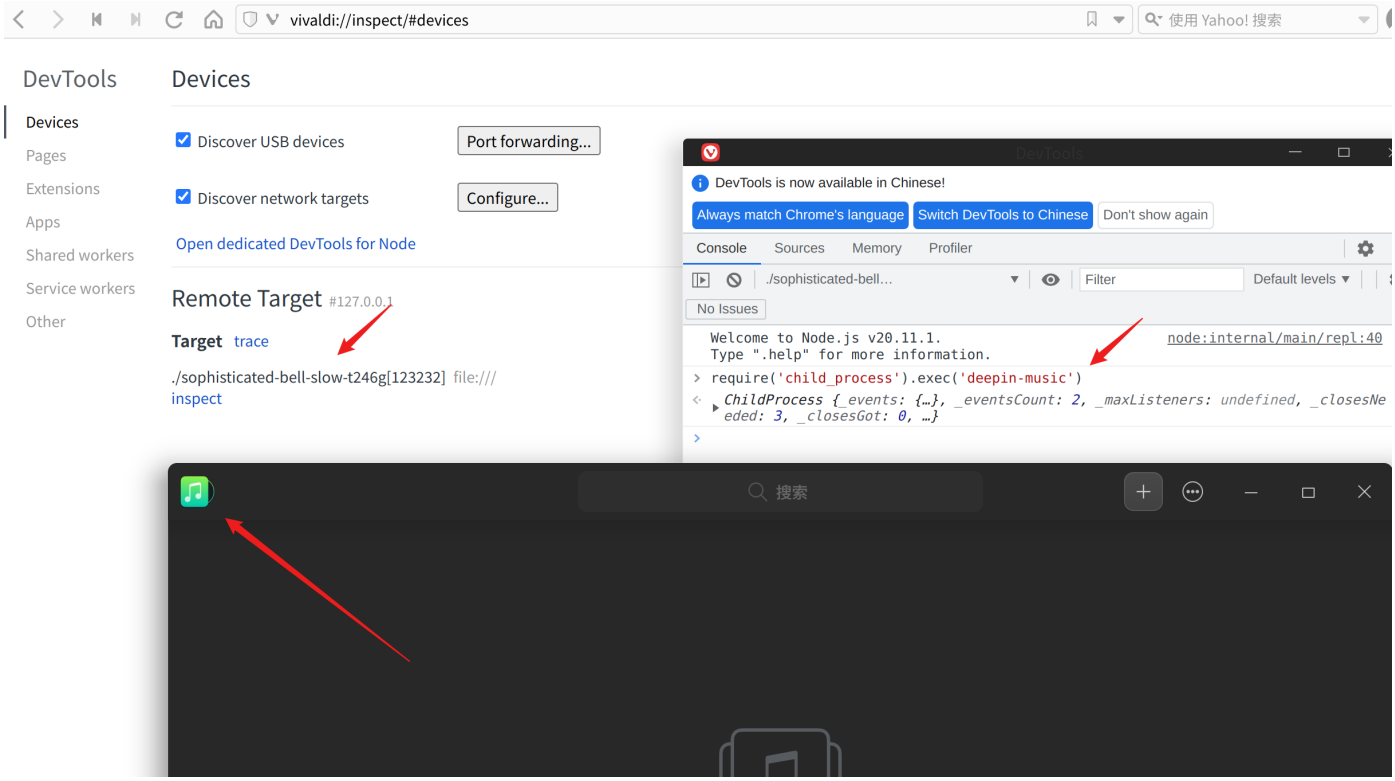
当 `runAsNode` 被设置为 `Enable` ，但是远程调试被设置为 `Disabled` 时会怎么样呢?

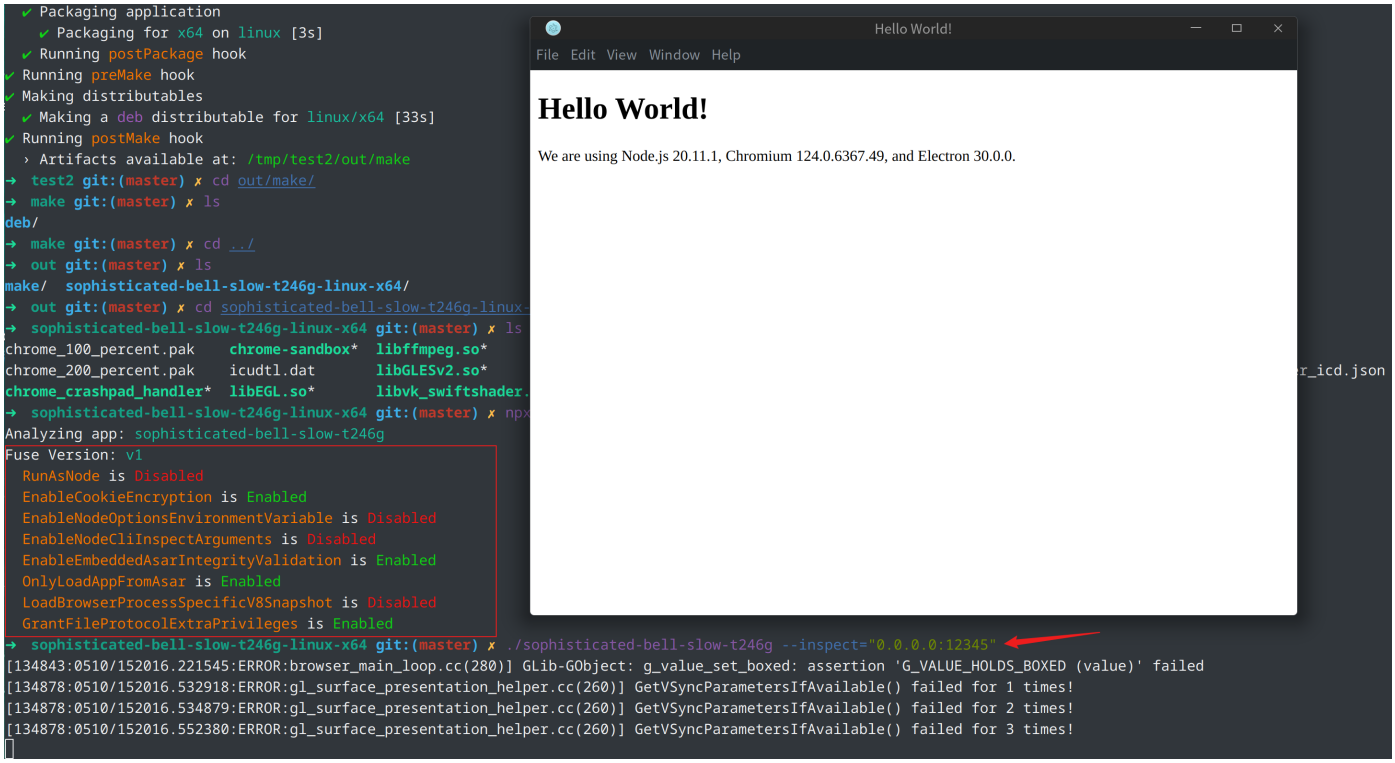


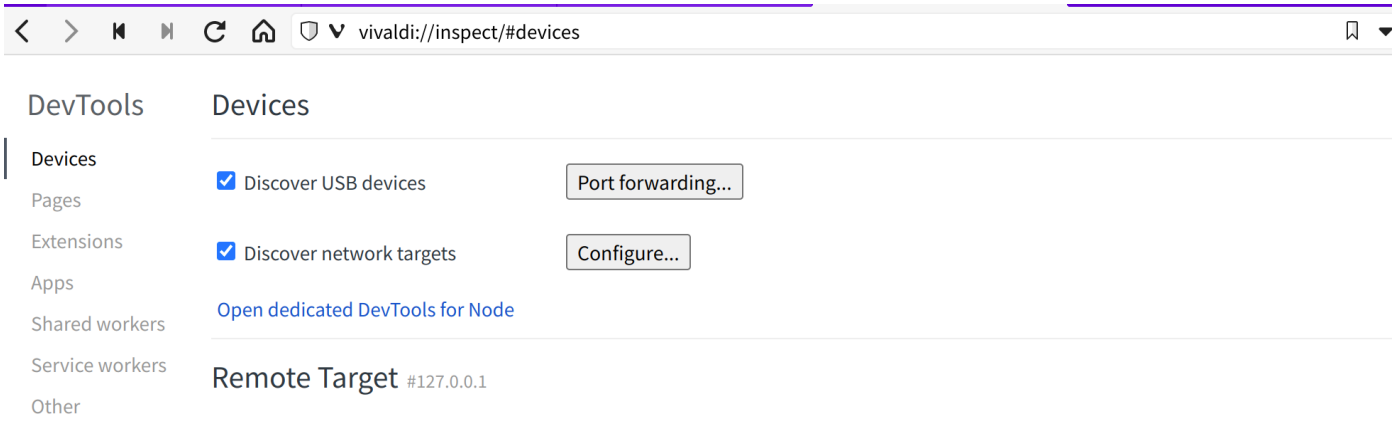


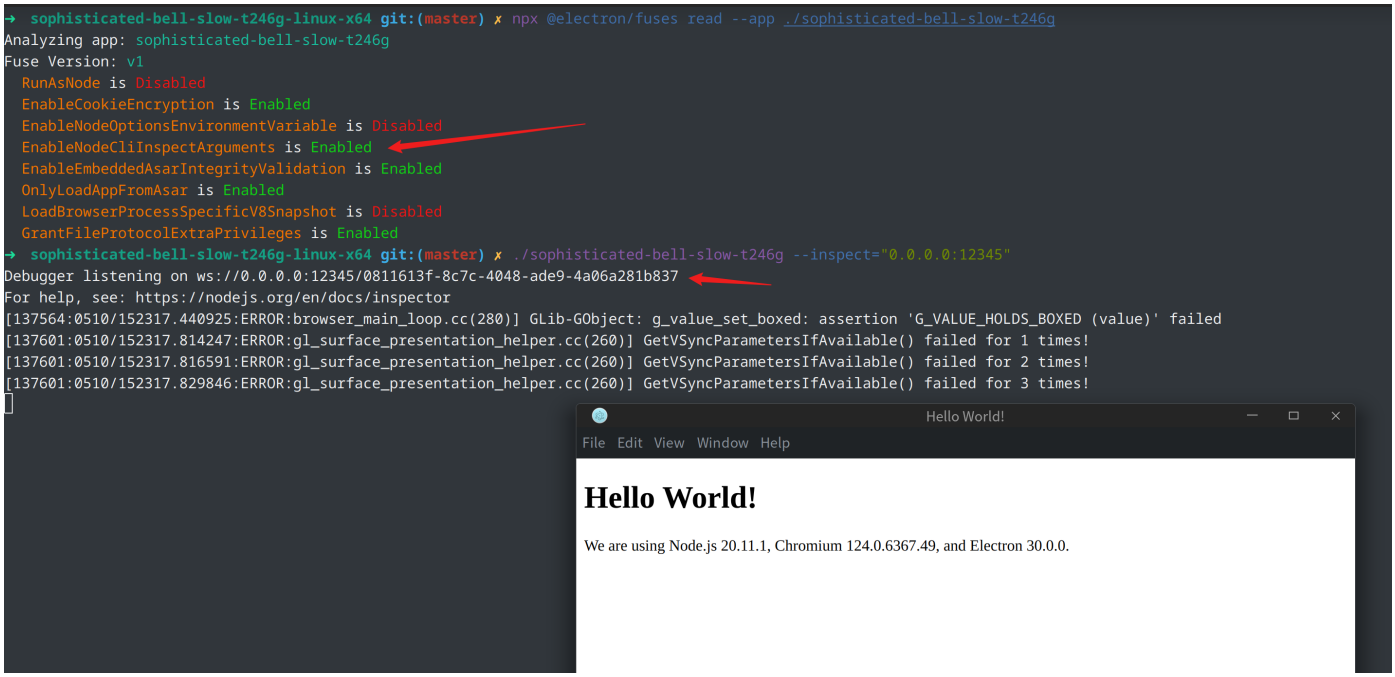
在 Windows 平台上并不会开启远程调试，但在 `Deepin Linux` 上则不同

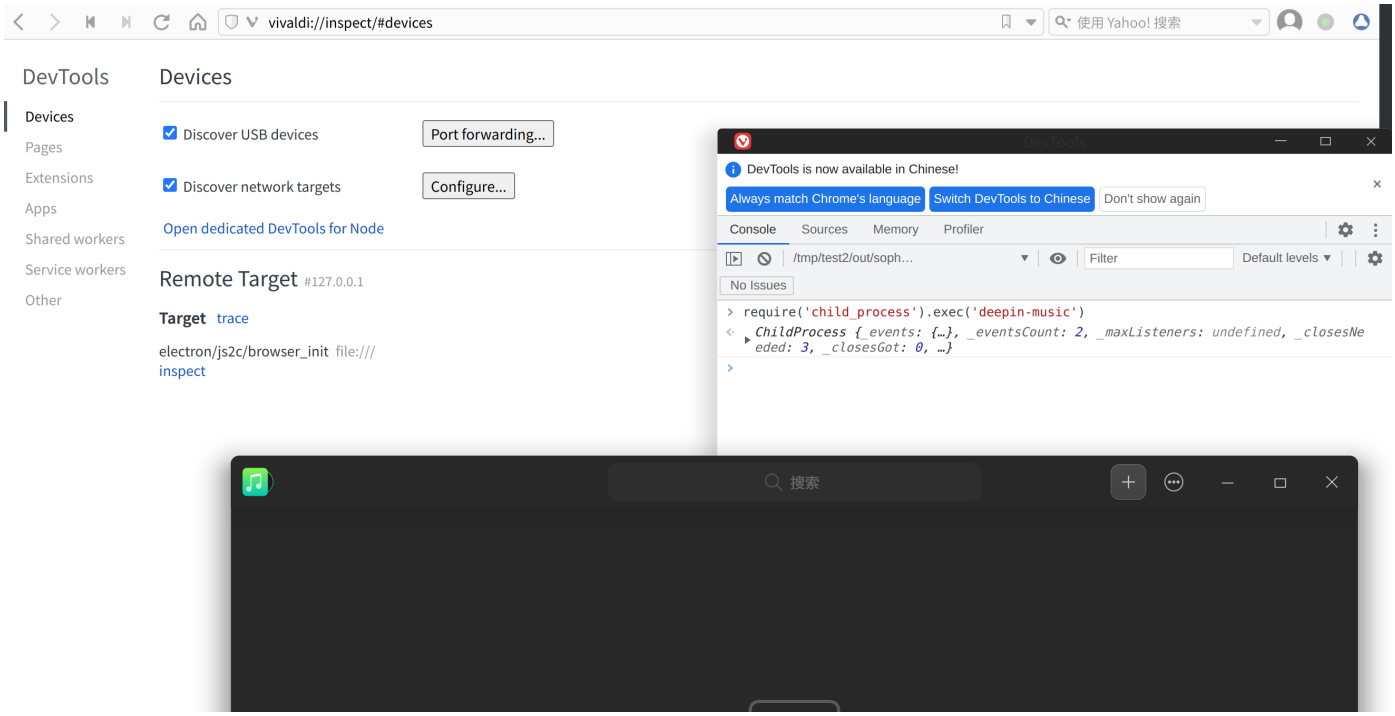
```
join@Electron:/tmp/test2/out/sophisticated-bell-slow-t246g-linux-x64$ npx @electron/fuses read --app sophisticated-bell-slow-t246g
npx: 13 安装成功，用时 8.106 秒
Analyzing app: sophisticated-bell-slow-t246g
Fuse Version: v1
  RunAsNode is Enabled
  EnableCookieEncryption is Enabled
  EnableNodeOptionsEnvironmentVariable is Disabled
  EnableNodeCliInspectArguments is Disabled
  EnableEmbeddedAsarIntegrityValidation is Enabled
  OnlyLoadAppFromAsar is Enabled
  LoadBrowserProcessSpecificV8Snapshot is Disabled
  GrantFileProtocolExtraPrivileges is Enabled
join@Electron:/tmp/test2/out/sophisticated-bell-slow-t246g-linux-x64$ ./sophisticated-bell-slow-t246g --inspect="0.0.0.0:12345"
Debugger listening on ws://0.0.0.0:12345/e34660da-15c5-4565-bd12-c7766f5361e7
For help, see: https://nodejs.org/en/docs/inspector
Welcome to Node.js v20.11.1.
Type ".help" for more information.
>
```


vivaldi://inspect/#devices
DevTools
Devices
Pages
Extensions
Apps
Shared workers
Service workers
Other
Devices
Discover USB devices
Port forwarding...
Discover network targets
Configure...
Open dedicated DevTools for Node
Remote Target #127.0.0.1
Target trace
./sophisticated-bell-slow-t246g[123232] file:///
inspect
DevTools is now available in Chinese!
Always match Chrome's language
Switch DevTools to Chinese
Don't show again
Console
Sources
Memory
Profiler
Filter
Default levels
No Issues
Welcome to Node.js v20.11.1.
Type ".help" for more information.
node:internal/main/repl:40
require('child_process').exec('deepin-music')
ChildProcess {_events: {…}, _eventsCount: 2, _maxListeners: undefined, _closesNeeded: 3, _closesGot: 0, …}
搜索


✔ Packaging application
✔ Packaging for x64 on linux [3s]
✔ Running postPackage hook
✔ Running preMake hook
✔ Making distributables
✔ Making a deb distributable for linux/x64 [33s]
✔ Running postMake hook
› Artifacts available at: /tmp/test2/out/make
→ test2 git:(master) ✗ cd out/make/
→ make git:(master) ✗ ls
deb/
→ make git:(master) ✗ cd ../
→ out git:(master) ✗ ls
make/ sophisticated-bell-slow-t246g-linux-x64/
→ out git:(master) ✗ cd sophisticated-bell-slow-t246g-linux-x64/
→ sophisticated-bell-slow-t246g-linux-x64 git:(master) ✗ ls
chrome_100_percent.pak chrome-sandbox* libffmpeg.so*
chrome_200_percent.pak icudtl.dat libGLESv2.so*
chrome_crashpad_handler* libEGL.so* libvk_swiftshader.
→ sophisticated-bell-slow-t246g-linux-x64 git:(master) ✗ npx
Analyzing app: sophisticated-bell-slow-t246g
Fuse Version: v1
RunAsNode is Disabled
EnableCookieEncryption is Enabled
EnableNodeOptionsEnvironmentVariable is Disabled
EnableNodeCliInspectArguments is Disabled
EnableEmbeddedAsarIntegrityValidation is Enabled
OnlyLoadAppFromAsar is Enabled
LoadBrowserProcessSpecificV8Snapshot is Disabled
GrantFileProtocolExtraPrivileges is Enabled
→ sophisticated-bell-slow-t246g-linux-x64 git:(master) ✗ ./sophisticated-bell-slow-t246g --inspect="0.0.0.0:12345"
[134843:0510/152016.221545:ERROR:browser_main_loop.cc(280)] GLib-GObject: g_value_set_boxed: assertion 'G_VALUE_HOLDS_BOXED (value)' failed
[134878:0510/152016.532918:ERROR:gl_surface_presentation_helper.cc(260)] GetVSyncParametersIfAvailable() failed for 1 times!
[134878:0510/152016.534879:ERROR:gl_surface_presentation_helper.cc(260)] GetVSyncParametersIfAvailable() failed for 2 times!
[134878:0510/152016.552380:ERROR:gl_surface_presentation_helper.cc(260)] GetVSyncParametersIfAvailable() failed for 3 times!
Hello World!
File Edit View Window Help
Hello World!
We are using Node.js 20.11.1, Chromium 124.0.6367.49, and Electron 30.0.0.


vivaldi://inspect/#devices
DevTools
Devices
Pages
Extensions
Apps
Shared workers
Service workers
Other
Devices
Discover USB devices
Port forwarding...
Discover network targets
Configure...
Open dedicated DevTools for Node
Remote Target #127.0.0.1


sophisticated-bell-slow-t246g git:(master) ✗ npx @electron/fuses read --app ./sophisticated-bell-slow-t246g
Analyzing app: sophisticated-bell-slow-t246g
Fuse Version: v1
RunAsNode is Disabled
EnableCookieEncryption is Enabled
EnableNodeOptionsEnvironmentVariable is Disabled
EnableNodeCliInspectArguments is Enabled
EnableEmbeddedAsarIntegrityValidation is Enabled
OnlyLoadAppFromAsar is Enabled
LoadBrowserProcessSpecificV8Snapshot is Disabled
GrantFileProtocolExtraPrivileges is Enabled
sophisticated-bell-slow-t246g git:(master) ✗ ./sophisticated-bell-slow-t246g --inspect="0.0.0.0:12345"
Debugger listening on ws://0.0.0.0:12345/0811613f-8c7c-4048-ade9-4a06a281b837
For help, see: https://nodejs.org/en/docs/inspector
[137564:0510/152317.440925:ERROR:browser_main_loop.cc(280)] GLib-GObject: g_value_set_boxed: assertion 'G_VALUE_HOLDS_BOXED (value)' failed
[137601:0510/152317.814247:ERROR:gl_surface_presentation_helper.cc(260)] GetVSyncParametersIfAvailable() failed for 1 times!
[137601:0510/152317.816591:ERROR:gl_surface_presentation_helper.cc(260)] GetVSyncParametersIfAvailable() failed for 2 times!
[137601:0510/152317.829846:ERROR:gl_surface_presentation_helper.cc(260)] GetVSyncParametersIfAvailable() failed for 3 times!
Hello World!
File Edit View Window Help
Hello World!
We are using Node.js 20.11.1, Chromium 124.0.6367.49, and Electron 30.0.0.

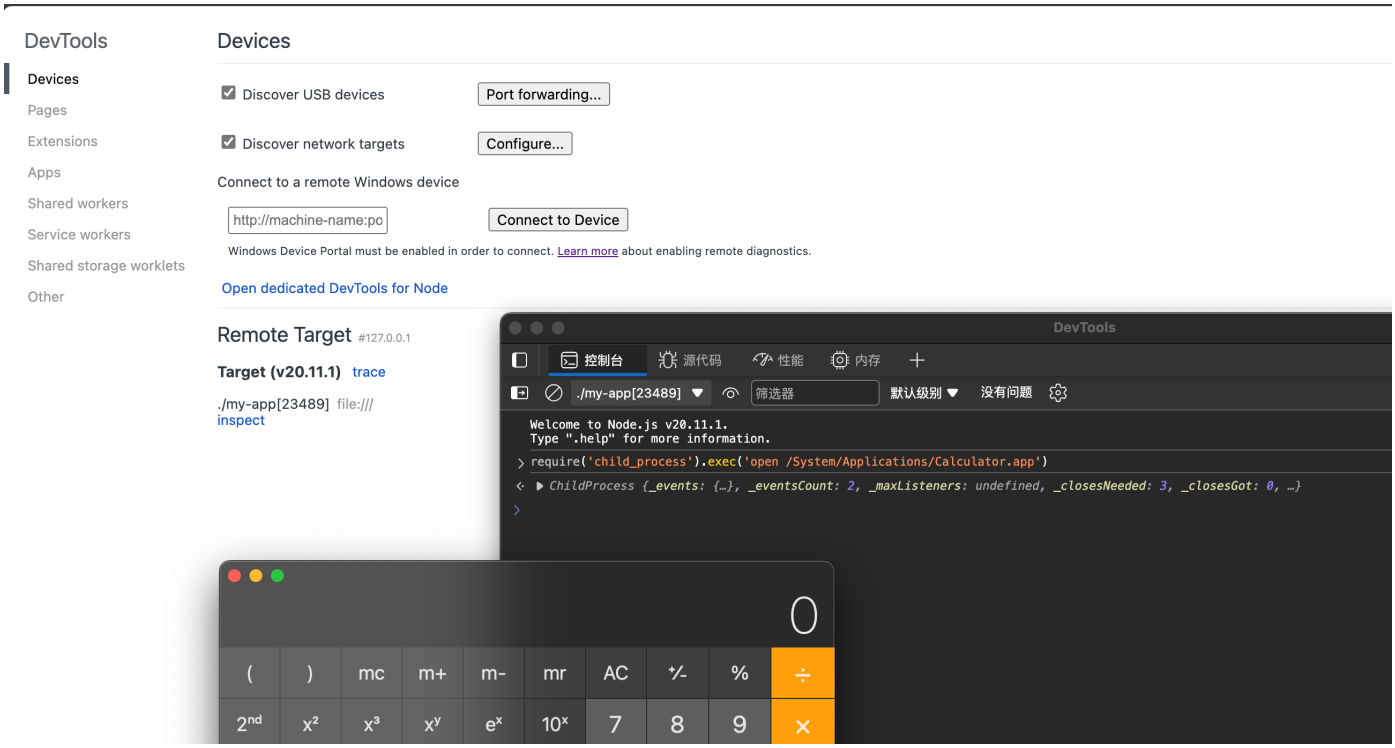

vivaldi://inspect/#devices
DevTools
Devices
Pages
Extensions
Apps
Shared workers
Service workers
Other
Devices
Discover USB devices
Port forwarding...
Discover network targets
Configure...
Open dedicated DevTools for Node
Remote Target #127.0.0.1
Target trace
electron/js2c/browser_init file:///
inspect
DevTools is now available in Chinese!
Always match Chrome's language
Switch DevTools to Chinese
Don't show again
Console
Sources
Memory
Profiler
No Issues
require('child_process').exec('deepin-music')
搜索

在 `Deepin Linux` 上，当 `runAsNode` 或 `nodeCliInspect` 其中一个被设置为 `Enabled` ，就可以进行远程调试

在 MacOS 上表现如何呢

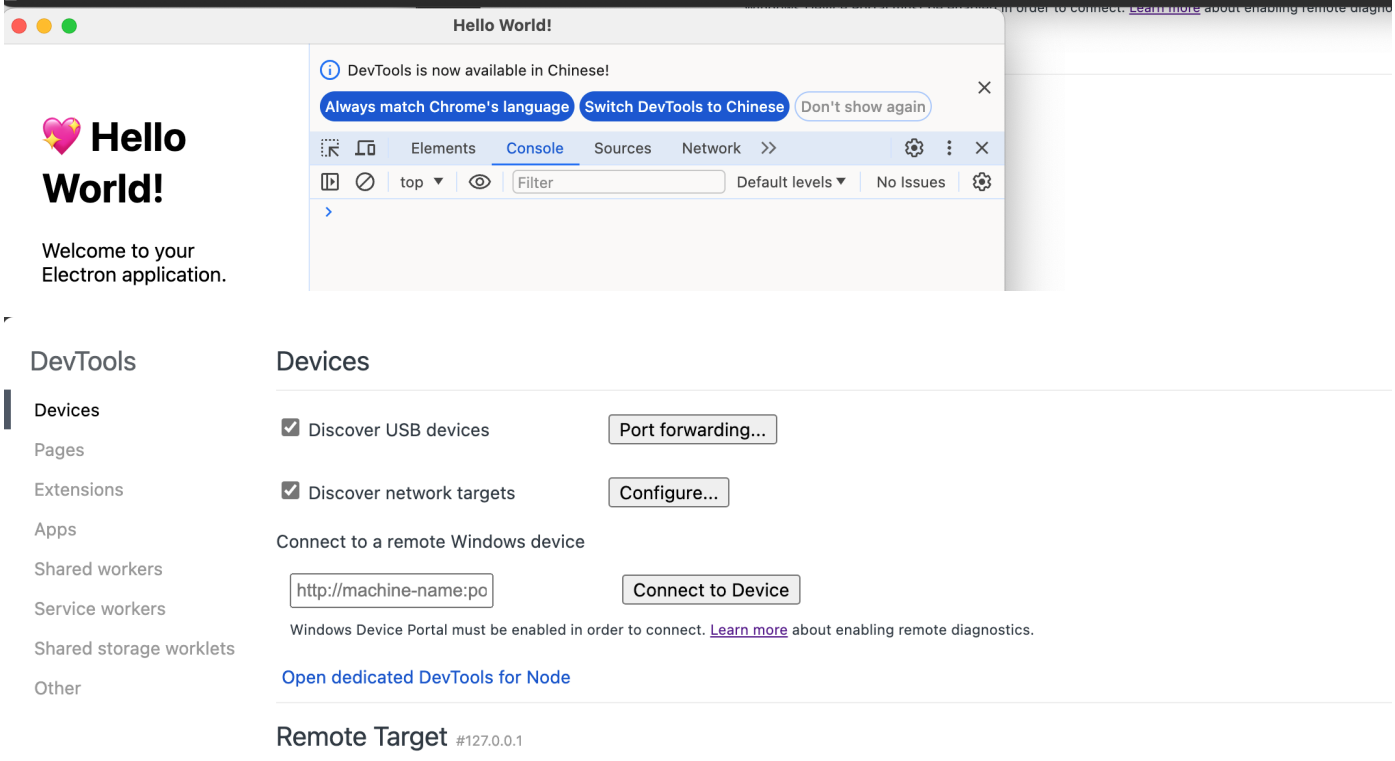
当 `runAsNode` 为 `Enable` ，远程调试设置为 `Disabled` 时

```
bash-3.2$ npx @electron/fuses read --app my-app.app
Analyzing app: my-app.app
Fuse Version: v1
  RunAsNode is Enabled
  EnableCookieEncryption is Enabled
  EnableNodeOptionsEnvironmentVariable is Disabled
  EnableNodeCliInspectArguments is Disabled
  EnableEmbeddedAsarIntegrityValidation is Enabled
  OnlyLoadAppFromAsar is Enabled
  LoadBrowserProcessSpecificV8Snapshot is Disabled
  GrantFileProtocolExtraPrivileges is Enabled
bash-3.2$ cd my-app.app/Contents/MacOS/
bash-3.2$ ./my-app --inspect="0.0.0.0:12345"
Debugger listening on ws://0.0.0.0:12345/acd741c5-074c-47a9-ab42-bcdeeb4e1bc5
For help, see: https://nodejs.org/en/docs/inspector
Welcome to Node.js v20.11.1.
Type ".help" for more information.
>
```



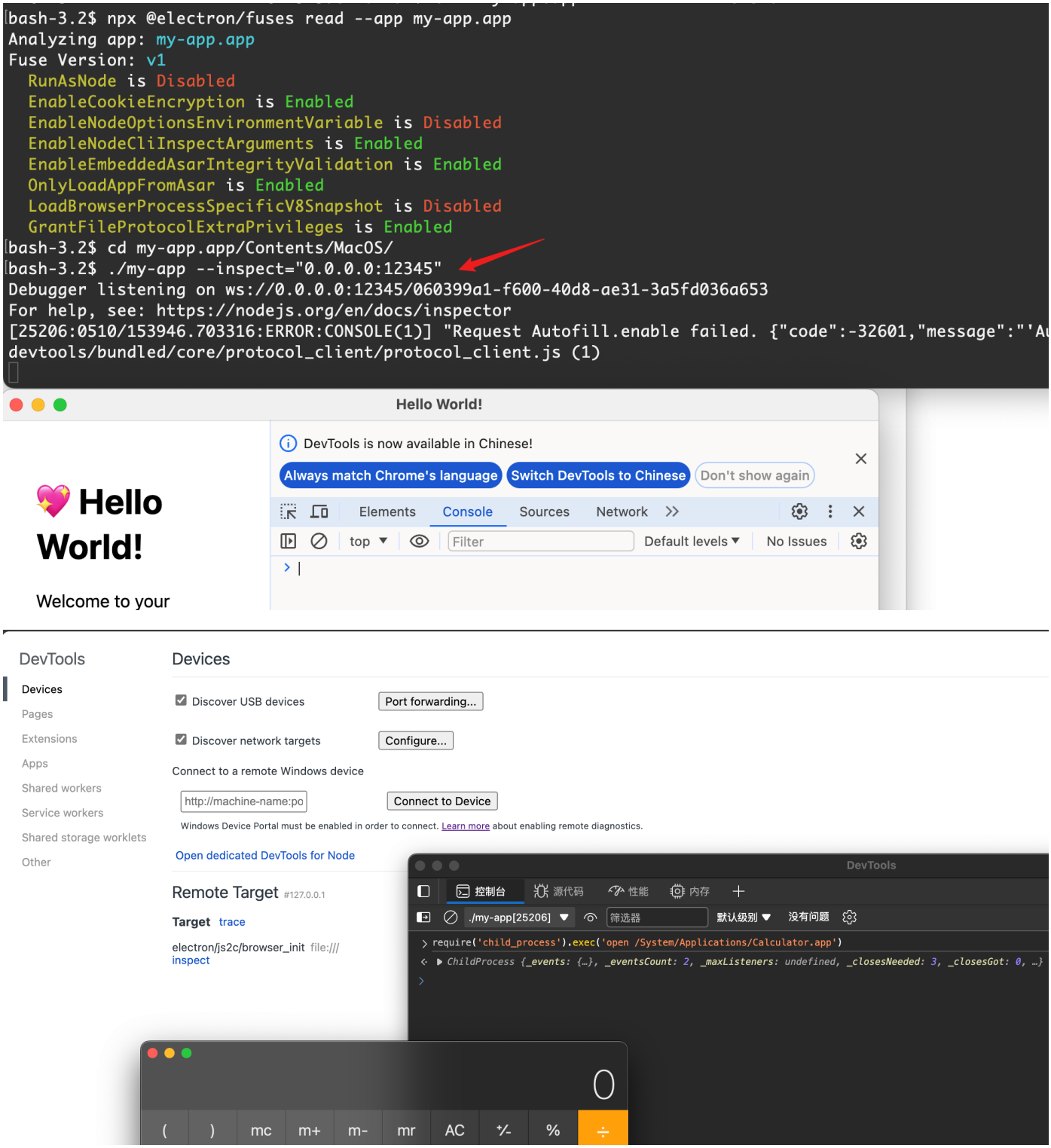
当 `runAsNode` 和远程调试都设置为 `Disabled` 时

```
bash-3.2$ npx @electron/fuses read --app my-app.app
Analyzing app: my-app.app
Fuse Version: v1
  RunAsNode is Disabled
  EnableCookieEncryption is Enabled
  EnableNodeOptionsEnvironmentVariable is Disabled
  EnableNodeCliInspectArguments is Disabled
  EnableEmbeddedAsarIntegrityValidation is Enabled
  OnlyLoadAppFromAsar is Enabled
  LoadBrowserProcessSpecificV8Snapshot is Disabled
  GrantFileProtocolExtraPrivileges is Enabled
bash-3.2$ cd my-app.app/Contents/MacOS/
bash-3.2$ ./my-app --inspect="0.0.0.0:12345"
[24614:0510/153619.446425:ERROR:CONSOLE(1)] "Request Autofill.enable failed. {"code":-32601,"message":"'Autofill.enable' wa
devtools/bundled/core/protocol_client/protocol_client.js (1)
```



无法执行远程调试

当 `runAsNode` 为 `Disabled` ，远程调试设置为 `Enabled` 时



可以远程调试

所以 `nodeCliInspect` 这个 fuse 的效果设置在 MacOS 和 Deepin Linux 上表现一致，即当 `runAsNode` 或 `nodeCliInspect` 其中一个被设置为 `Enabled` ，就可以进行远程调试

在 Windows 11 上则只有当 `nodeCliInspect` 被设置为 `Enabled` 时才可以进行远程调试，与 `runAsNode` 无关

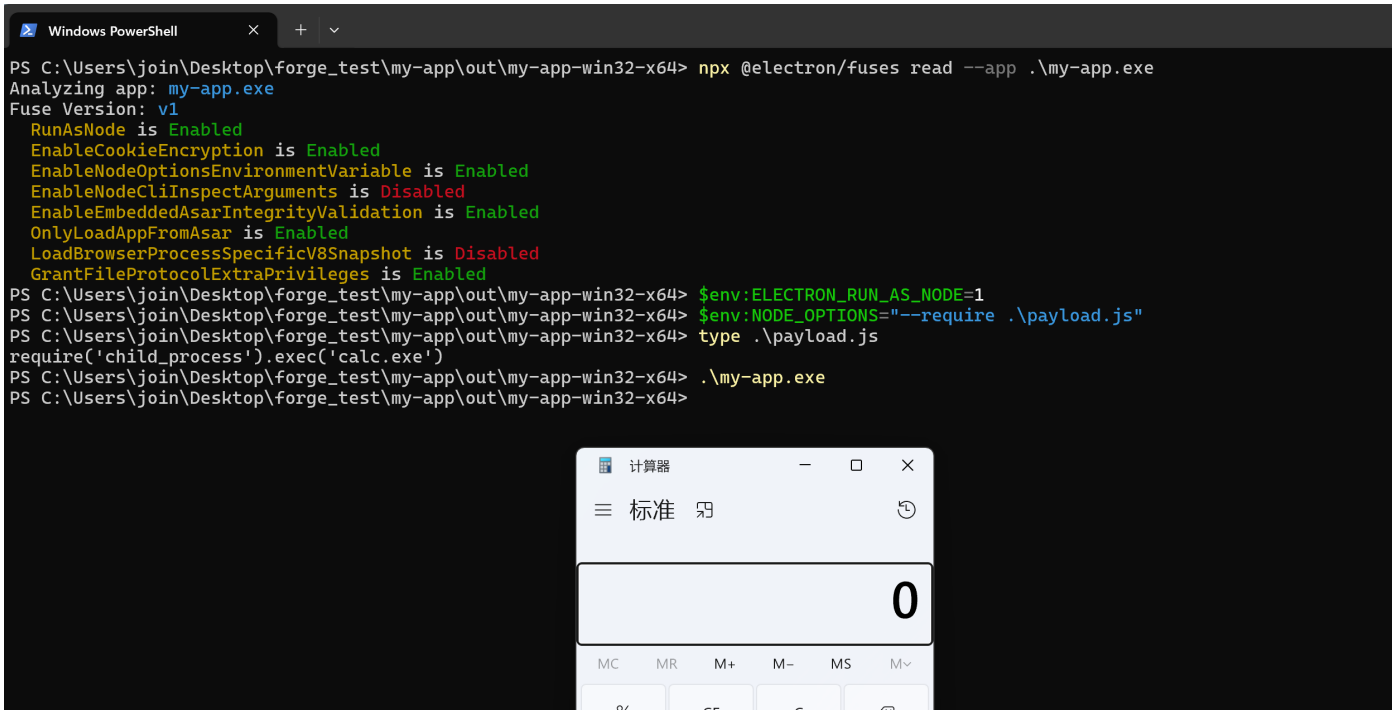
不过 `Electron` 还在发展，在未来可能还会有变化

3. nodeOptions

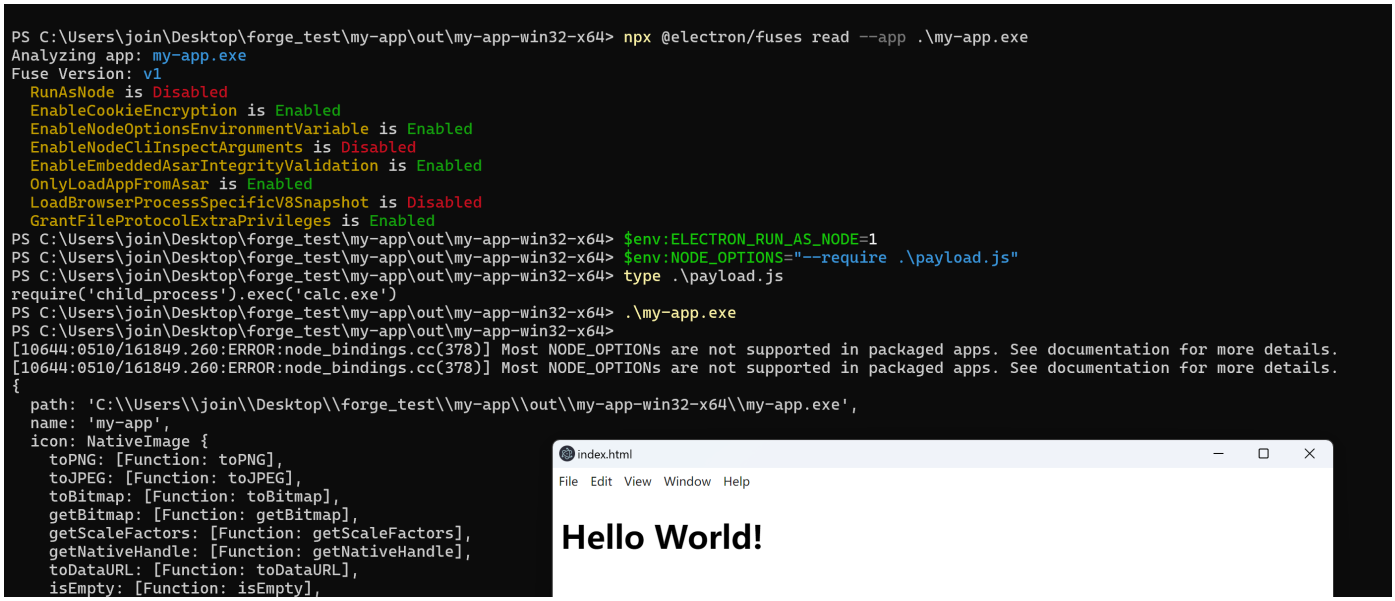
这个 fuse 是决定程序是否要使用两个环境变量 `NODE_OPTIONS` 和 `NODE_EXTRA_CA_CERTS`

> https://nodejs.org/api/cli.html#node_optionsoptions
>
> https://github.com/nodejs/node/blob/main/doc/api/cli.md#node_extra_ca_certsfile

这个 fuse 只在 `runAsNode` 被设置为 `Enabled` 时有效，其实就是给 `Node.js` 传递的那些参数被写进了这个环境变量里



关闭 `runAsNode` 后



就无法运行 `Node.js` 代码并执行系统命令了

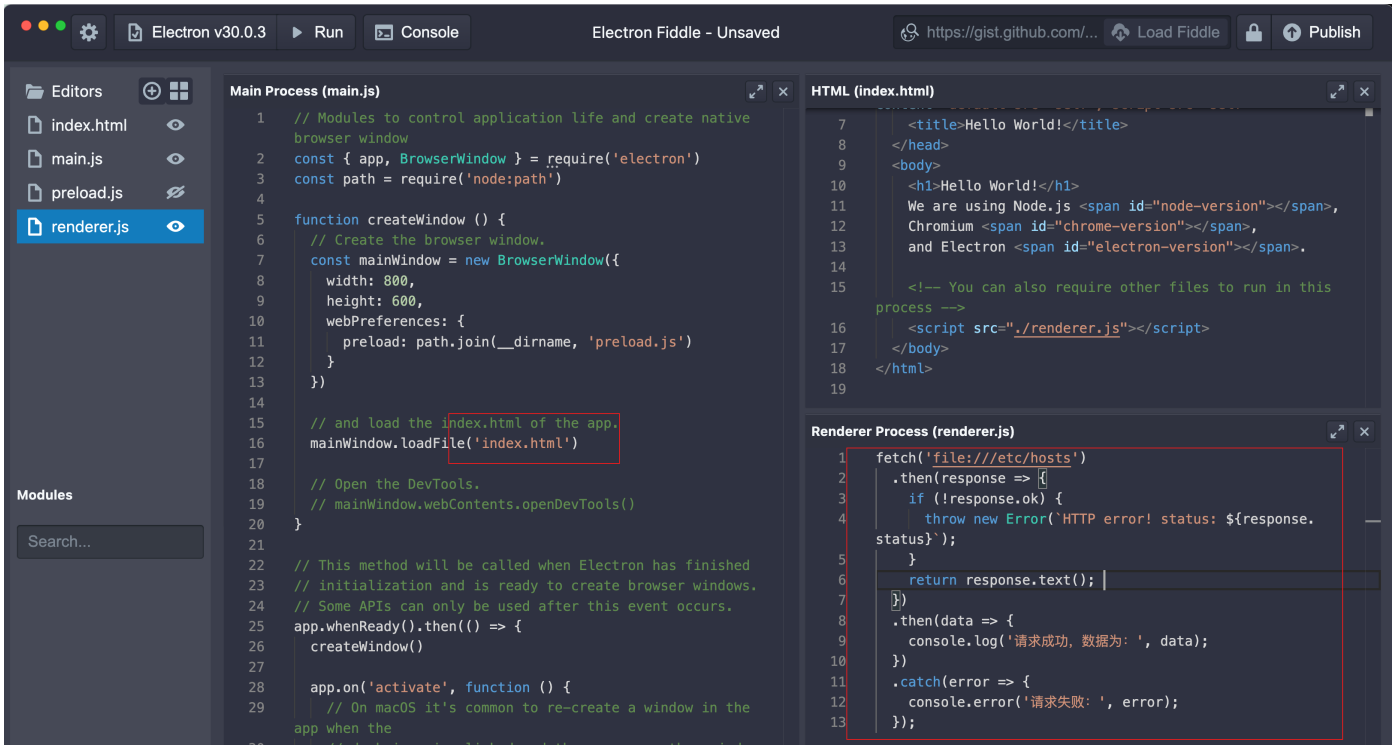
4. grantFileProtocolExtraPrivileges

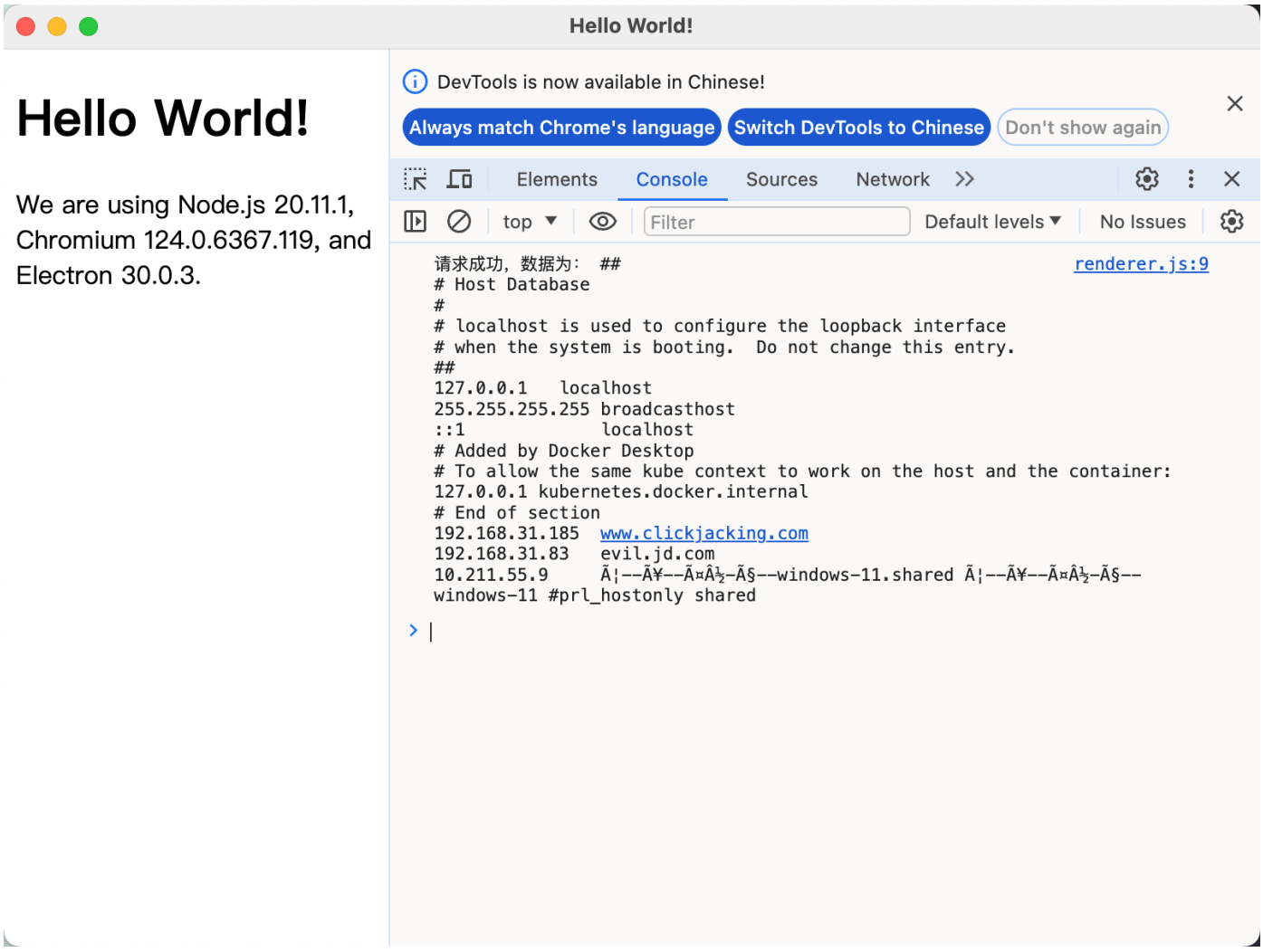
这个 fuse 是关于 `file://` 协议的，在 `Electron` 中 `file://` 协议比 web 浏览器中的 `file://` 协议具备更强大的功能，包括但不限于

- `file://` 协议加载的页面可以通过 `fetch` 加载其他 `file://` 协议的资源
- `file://` 协议加载的页面能够使用 `service workers`
- `file://` 协议加载的页面能够访问子 `frames`
- `file://` 无视沙盒限制

官方推荐，加载本地文件尽可能使用自定义协议，而不是开启这个 fuse ，对于旧版本 `Electron` ，这是核心功能，所以默认开启；在 `Electron Forge` 中也没有对其进行额外设置，这是合理的，毕竟不是所有开发者都会去自定义协议

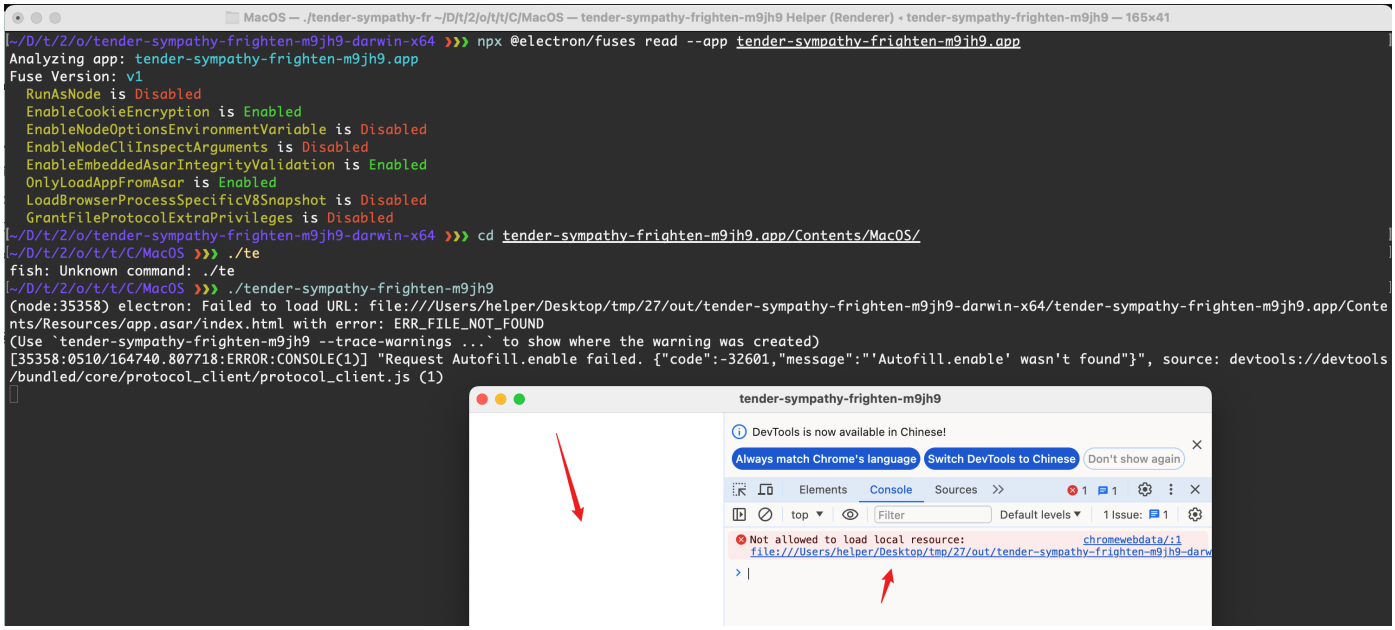
我们尝试直接使用 `fiddle` 进行测试第一项





确实可以获取到数据，而且之前就测试过，`file` 协议之间没有同源策略限制

现在我们将程序用 `Electron Forge` 进行打包



不仅使用 `fetch` 请求 `file://` 协议资源不可用了，通过 `file://` 创建主窗口都不行了，可能需要对路径进行配置，但明确的是使用 `fetch` 请求 `file://` 协议资源这种特权没有了

5. CVE-2024-23743

这个 `CVE` 就是上面提到的 `runAsNode` 和 `nodeCliInspect` 开启的效果，有一些安全研究员直接给提交 CVE 了，我看了一下，好像还提交了不少，为此官方在博客中专门谈了一下这个问题

官方的态度首先是认为描述不准确，提交者认为这是一个远程代码执行，并且认定是严重 `critical`，其实是强调和 `Chrome` 的漏洞模型保持一致，不考虑本地物理攻击

其实这些 fuse 的问题是因为在一场安全大会上，有位安全研究员提出来的，并且还制作了一个检测工具，具体官方声明以及检测工具查看下方链接

> https://www.electronjs.org/zh/blog/statement-run-as-node-cves
>
> https://github.com/r3ggi/electroniz3r

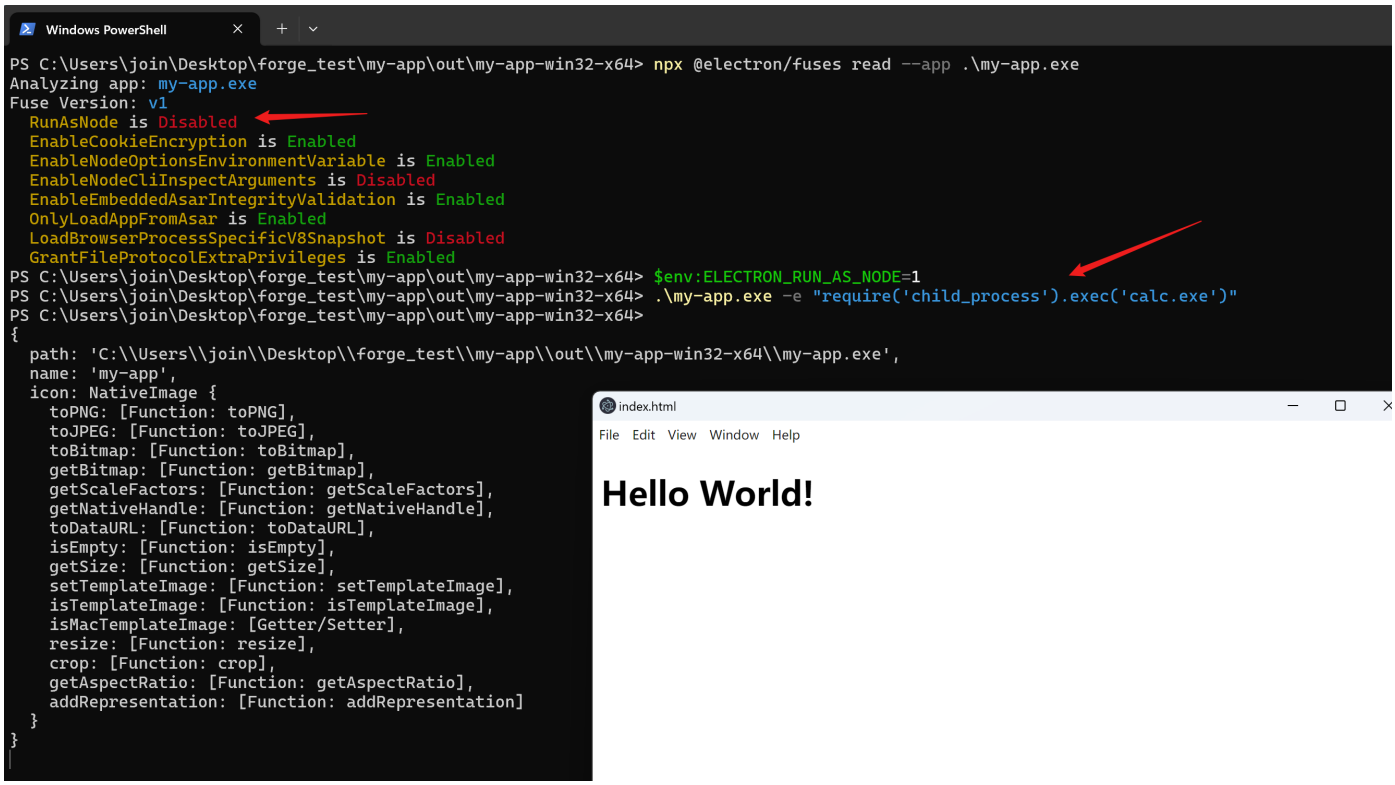
0x05 修改程序的 fuse

程序的 fuse 是可以手动修改的，由于 fuse 是在签名前打包时候设置的，所以在签名后修改 fuse 应该会导致签名失效

有两种方式，一种是使用官方的工具 `@electron/fuses` ，另一种方式是直接修改二进制文件，官方提供了一些格式信息，但显然，官方的工具是更简单的

```
PS C:\Users\join\Desktop\forge_test\my-app\out\my-app-win32-x64> npx @electron/fuses read --app .\my-app.exe
Analyzing app: my-app.exe
Fuse Version: v1
  RunAsNode is Disabled
  EnableCookieEncryption is Enabled
  EnableNodeOptionsEnvironmentVariable is Enabled
  EnableNodeCliInspectArguments is Disabled
  EnableEmbeddedAsarIntegrityValidation is Enabled
  OnlyLoadAppFromAsar is Enabled
  LoadBrowserProcessSpecificV8Snapshot is Disabled
  GrantFileProtocolExtraPrivileges is Enabled
PS C:\Users\join\Desktop\forge_test\my-app\out\my-app-win32-x64>
```

可以看到，当前程序的 `RunAsNode` 是 `Disabled` 的，也就是无法当作 `Node.js` 执行



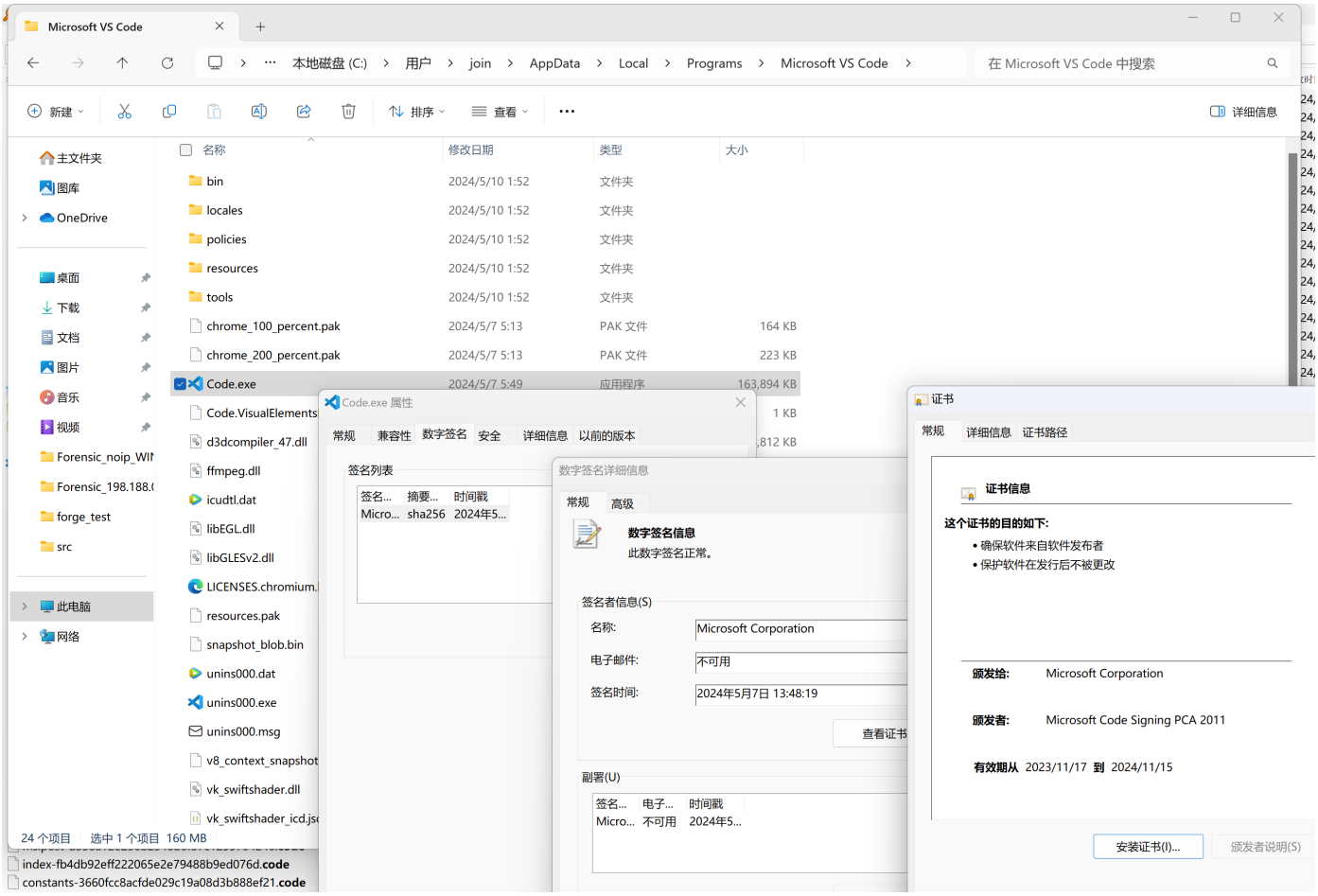
现在我们尝试翻转 `RunAsNode`

```
PS C:\Users\join\Desktop\forge_test\my-app\out\my-app-win32-x64> npx @electron/fuses write --app .\my-app.exe RunAsNode=on
Analyzing app: my-app.exe
Fuse Version: v1
  RunAsNode is Disabled and will become Enabled
Writing to app: my-app.exe
Fuses written to disk
PS C:\Users\join\Desktop\forge_test\my-app\out\my-app-win32-x64> npx @electron/fuses read --app .\my-app.exe
Analyzing app: my-app.exe
Fuse Version: v1
  RunAsNode is Enabled
  EnableCookieEncryption is Enabled
  EnableNodeOptionsEnvironmentVariable is Enabled
  EnableNodeCliInspectArguments is Disabled
  EnableEmbeddedAsarIntegrityValidation is Enabled
  OnlyLoadAppFromAsar is Enabled
  LoadBrowserProcessSpecificV8Snapshot is Disabled
  GrantFileProtocolExtraPrivileges is Enabled
PS C:\Users\join\Desktop\forge_test\my-app\out\my-app-win32-x64>
```

现在 `RunAsNode` 变成 `Enabled` 了，尝试执行 `Node.js`

```
PS C:\Users\join\Desktop\forge_test\my-app\out\my-app-win32-x64> npx @electron/fuses write --app .\my-app.exe RunAsNode=on
Analyzing app: my-app.exe
Fuse Version: v1
  RunAsNode is Disabled and will become Enabled
Writing to app: my-app.exe
Fuses written to disk
PS C:\Users\join\Desktop\forge_test\my-app\out\my-app-win32-x64> npx @electron/fuses read --app .\my-app.exe
Analyzing app: my-app.exe
Fuse Version: v1
  RunAsNode is Enabled
  EnableCookieEncryption is Enabled
  EnableNodeOptionsEnvironmentVariable is Enabled
  EnableNodeCliInspectArguments is Disabled
  EnableEmbeddedAsarIntegrityValidation is Enabled
  OnlyLoadAppFromAsar is Enabled
  LoadBrowserProcessSpecificV8Snapshot is Disabled
  GrantFileProtocolExtraPrivileges is Enabled
PS C:\Users\join\Desktop\forge_test\my-app\out\my-app-win32-x64> .\my-app.exe -i
PS C:\Users\join\Desktop\forge_test\my-app\out\my-app-win32-x64> Welcome to Node.js v20.11.1.
Type ".help" for more information.
>                                                 > require('child_process').exec('calc.exe')
方法调用失败，因为 [System.String] 不包含名为“exec”的方法。
所在位置 行:1 字符: 1
+ require('child_process').exec('calc.exe')
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : InvalidOperation: (:) [], RuntimeException
    + FullyQualifiedErrorId : MethodNotFound

PS C:\Users\join\Desktop\forge_test\my-app\out\my-app-win32-x64> .\my-app.exe -e "require('child_process').exec('calc.exe')"
PS C:\Users\join\Desktop\forge_test\my-app\out\my-app-win32-x64>
```

接下来我们测试一下，将 `vscode` 的 fuse 翻转后，它的签名是否依旧有效



`"C:\Users\join\AppData\Local\Programs\Microsoft VS Code\Code.exe"`

读取 `vscode` 的 fuse

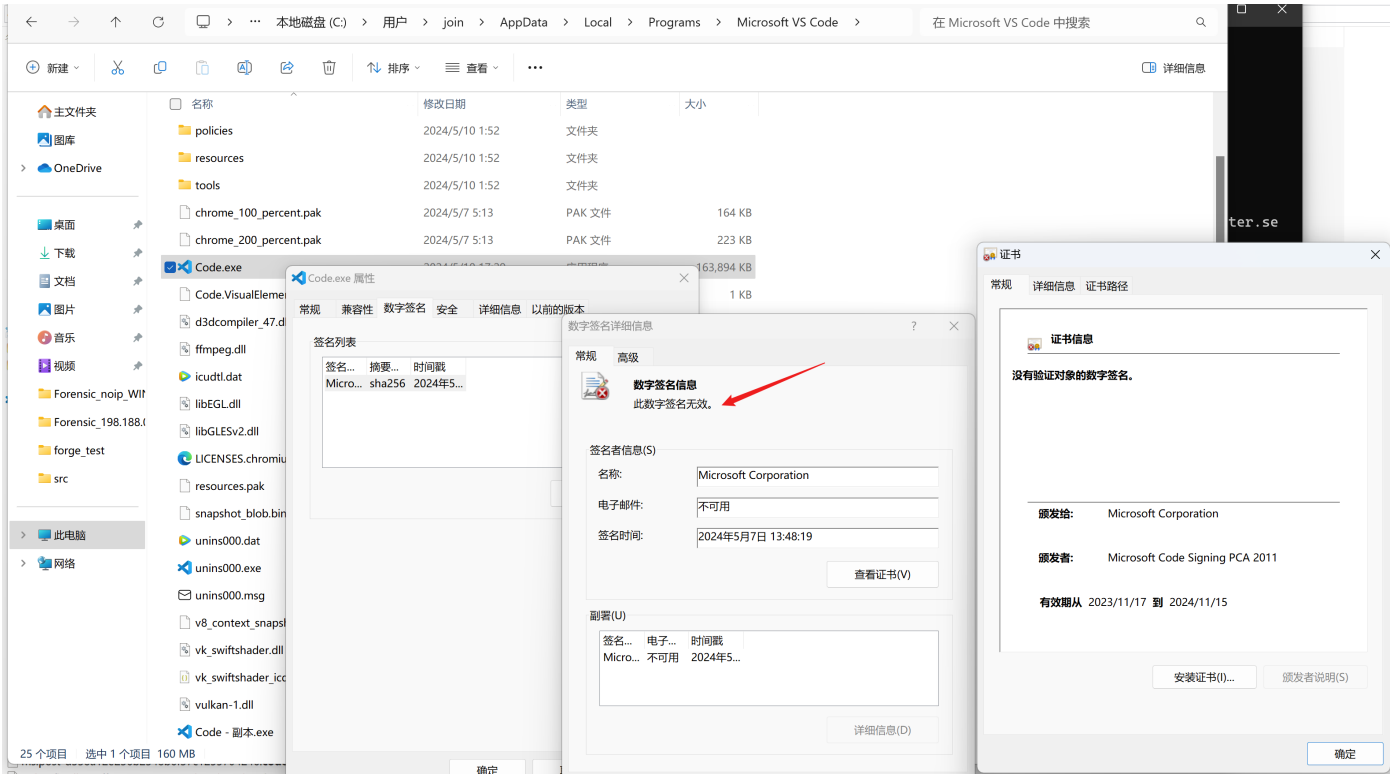
```
npx @electron/fuses read --app
"C:\Users\join\AppData\Local\Programs\Microsoft VS Code\Code.exe"
```

```
PS C:\Users\join> npx @electron/fuses read --app "C:\Users\join\AppData\Local\Programs\Microsoft VS Code\Code.exe"
Analyzing app: Code.exe
Fuse Version: v1
  RunAsNode is Enabled
  EnableCookieEncryption is Disabled
  EnableNodeOptionsEnvironmentVariable is Enabled
  EnableNodeCliInspectArguments is Enabled
  EnableEmbeddedAsarIntegrityValidation is Disabled
  OnlyLoadAppFromAsar is Disabled
  LoadBrowserProcessSpecificV8Snapshot is Disabled
PS C:\Users\join>
```

备份好原文件后，尝试将 `RunAsNode` 设置为 `Disabled`

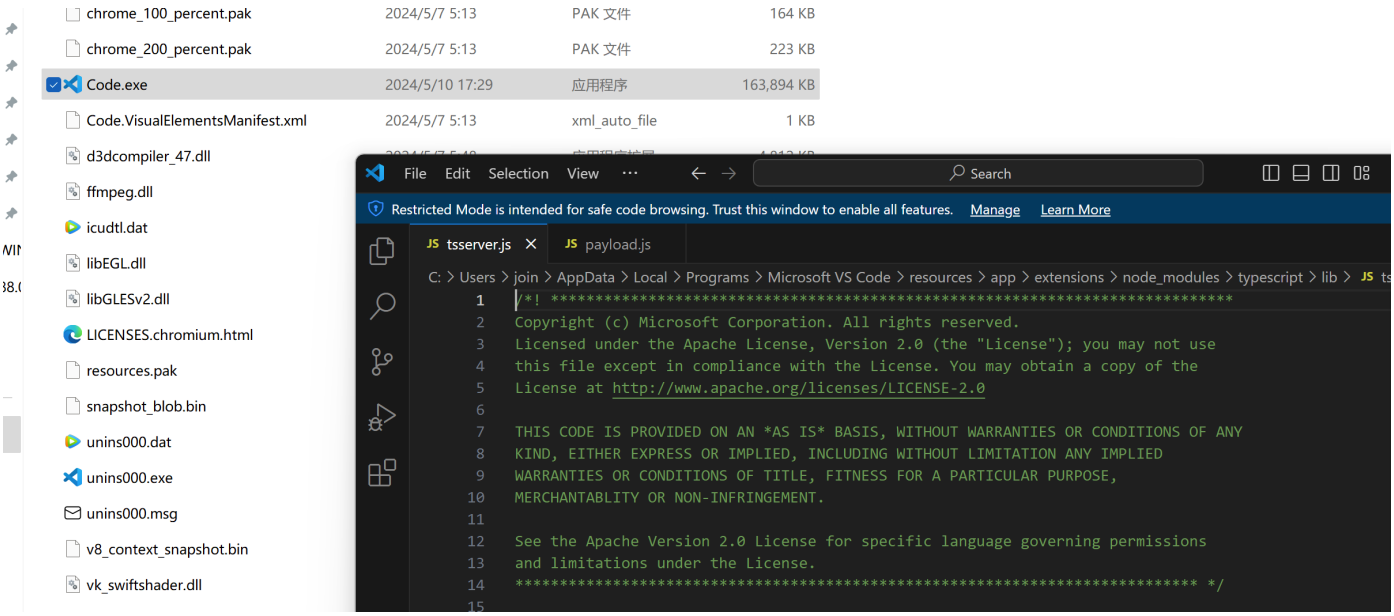
```
PS C:\Users\join> npx @electron/fuses read --app "C:\Users\join\AppData\Local\Programs\Microsoft VS Code\Code.exe"
Analyzing app: Code.exe
Fuse Version: v1
  RunAsNode is Enabled
  EnableCookieEncryption is Disabled
  EnableNodeOptionsEnvironmentVariable is Enabled
  EnableNodeCliInspectArguments is Enabled
  EnableEmbeddedAsarIntegrityValidation is Disabled
  OnlyLoadAppFromAsar is Disabled
  LoadBrowserProcessSpecificV8Snapshot is Disabled
PS C:\Users\join> npx @electron/fuses write --app "C:\Users\join\AppData\Local\Programs\Microsoft VS Code\Code.exe" RunAsNode=off
(node:5100) MaxListenersExceededWarning: Possible EventEmitter memory leak detected. 11 close listeners added to [TLSSocket]. Use emitter.se
tMaxListeners() to increase limit
(Use `node --trace-warnings ...` to show where the warning was created)
Analyzing app: Code.exe
Fuse Version: v1
  RunAsNode is Enabled and will become Disabled
Writing to app: Code.exe
Fuses written to disk
PS C:\Users\join> npx @electron/fuses read --app "C:\Users\join\AppData\Local\Programs\Microsoft VS Code\Code.exe"
Analyzing app: Code.exe
Fuse Version: v1
  RunAsNode is Disabled
  EnableCookieEncryption is Disabled
  EnableNodeOptionsEnvironmentVariable is Enabled
  EnableNodeCliInspectArguments is Enabled
  EnableEmbeddedAsarIntegrityValidation is Disabled
  OnlyLoadAppFromAsar is Disabled
  LoadBrowserProcessSpecificV8Snapshot is Disabled
PS C:\Users\join>
```

成功翻转 `RunAsNode` 的设置，我们看一下签名情况



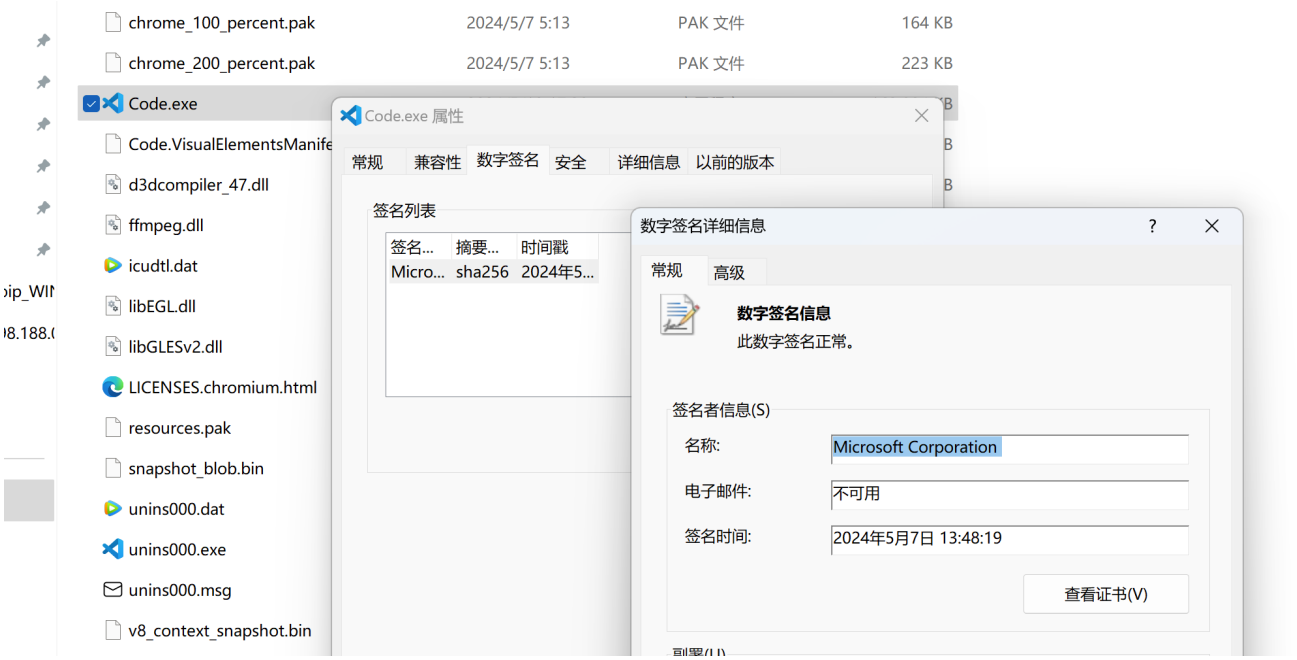
果然，签名失效

是否可以正常执行呢？



功能依旧正常，我们尝试再翻转一次，看看能不能让签名再次生效

```
PS C:\Users\join> npx @electron/fuses write --app "C:\Users\join\AppData\Local\Programs\Microsoft VS Code\Code.exe" RunAsNode=on
Analyzing app: Code.exe
Fuse Version: v1
  RunAsNode is Disabled and will become Enabled
Writing to app: Code.exe
Fuses written to disk
PS C:\Users\join> npx @electron/fuses read --app "C:\Users\join\AppData\Local\Programs\Microsoft VS Code\Code.exe"
Analyzing app: Code.exe
Fuse Version: v1
  RunAsNode is Enabled
  EnableCookieEncryption is Disabled
  EnableNodeOptionsEnvironmentVariable is Enabled
  EnableNodeCliInspectArguments is Enabled
  EnableEmbeddedAsarIntegrityValidation is Disabled
  OnlyLoadAppFromAsar is Disabled
  LoadBrowserProcessSpecificV8Snapshot is Disabled
PS C:\Users\join>
```



签名又恢复了正常，这有点意思

0x06 总结

`Electron` 开发的程序默认会带有一些特性，然而这些特性并不总是能用到，甚至很多特性大部分开发者都用不到，所以官方给了一个总开关，可以在打包等过程中，显式的关闭或启用这些特性

目前来看，这些特性能够引起的主要是本地命令执行、文件读取，主要涉及的特性如下

- runAsNode
- nodeCliInspect
- nodeOptions
- grantFileProtocolExtraPrivileges

应用程序的 fuse 是可以翻转的，官方也提供了工具，由于特性的启用与关闭是在打包过程中完成的，所以翻转已经签名的程序的fuse 会导致签名失效，但将已经翻转的fuse 再翻转一次，保持和原本的程序一致，签名就会重新生效