1. webpack编译流程

- 初始化参数：从配置文件和 Shell 语句中读取与合并参数，得出最终的参数；
- 开始编译：用上一步得到的参数初始化 Compiler 对象，加载所有配置的插件，执行对象的 run 方法开始执行编译； 确定入口：根据配置中的 entry 找出所有的入口文件
- 编译模块：从入口文件出发，调用所有配置的 Loader 对模块进行编译，再找出该模块依赖的模块，再递归本步骤直到所有入口依赖的文件都经过了本步骤的处理；
- 完成模块编译：在经过第4步使用 Loader 翻译完所有模块后，得到了每个模块被翻译后的最终内容以及它们之间的依赖关系
- 输出资源：根据入口和模块之间的依赖关系，组装成一个个包含多个模块的 Chunk，再把每个 Chunk 转换成一个单独的文件加入到输出列表，这步是可以修改输出内容的最后机会
- 输出完成：在确定好输出内容后，根据配置确定输出的路径和文件名，把文件内容写入到文件系统

> 在以上过程中，Webpack 会在特定的时间点广播出特定的事件，插件在监听到感兴趣的事件后会执行特定的逻辑，并且插件可以调用 Webpack 提供的 API 改变 Webpack 的运行结果

2.调试webpack

- cli (https://webpack.js.org/api/cli)

2.1 如何生成调试文件

- 打开工程目录，点击调试按钮，再点击小齿轮的配置按钮系统就会生成launch.json配置文件
- 修改好了以后直接点击F5就可以启动调试

.vscode\launch.json

```
{
    "version": "0.2.0",
    "configurations": [
        {
            "type": "node",
            "request": "launch",
            "name": "debug webpack",
            "cwd":"${workspaceFolder}",
            "program":"${workspaceFolder}/node_modules/webpack-cli/bin/cli.js"
        }
    ]
}
```

2.2 webpack.cmd

- webpack-source\node_modules\.bin\webpack.cmd
- %~dp0是批处理文件所在的盘符:+路径(%~dp0 C:\vipdata\vipproject\webpack-source\node_modules.bin)
- SETLOCAL主要针对临时环境变量，不会影响到系统的变量环境设置，应与endlocal联用
- PATHEXT当在一个相同的目录结构下，有相同的多个主文件名，不同的文件后缀名时，系统会根据PATHEXT中的后缀名，选择其中顺序最靠前的那一个

```
@IF EXIST "%~dp0\node.exe" (
  "%~dp0\node.exe"  "%~dp0\..\_webpack@4.39.3@webpack\bin\webpack.js" %*
) ELSE (
  @SETLOCAL
  @SET PATHEXT=%PATHEXT:;.JS;=;%
  node  "%~dp0\..\_webpack@4.39.3@webpack\bin\webpack.js" %*
)
```

2.3 webpack.js

- node_modules\webpack\bin\webpack.js

```
const path = require("path");
const pkgPath = require.resolve(`${installedClis[0].package}/package.json`);
const pkg = require(pkgPath);
require(path.resolve(
    path.dirname(pkgPath),
    pkg.bin[installedClis[0].binName]
));
```

```
const path = require("path");
const pkgPath = require.resolve(`webpack-cli/package.json`);
const pkg = require(pkgPath);
require(path.resolve(path.dirname(pkgPath),pkg.bin['webpack-cli']));
```

```
npx webpack = node ./node_modules/webpack-cli/bin/cli.js
```

2.4 cli.js

```
const webpack = require("webpack");
const webpackOptions = require("./webpack.config");
const compiler = webpack(webpackOptions);
compiler.run((err, stats) => {
  console.log(err);
  console.log(stats.toJson({
    entries:true,
    chunks:true,
    modules:true,
    _modules:true,
    assets:true
  }));
});
```

3.Stats 对象

- 在 Webpack 的回调函数中会得到stats对象
- 这个对象实际来自于 Compilation.getStats()，返回的是主要含有 modules、 chunks和 assets三个属性值的对象。
- Stats对象本质上来自于lib/Stats.js (https://github.com/webpack/webpack/blob/v4.39.3/lib/Stats.js)的类实例

字段 含义 modules 记录了所有解析后的模块 chunks 记录了所有chunk assets 记录了所有要生成的文件

```
npx webpack --profile --json > stats.json
```

```json
{
  "errors": [],
  "warnings": [],
  "version": "4.39.3",
  "hash": "3e945ec6b2c56d0b010e",
  "time": 66,
  "builtAt": 1567225465347,
  "publicPath": "",
  "outputPath": "C:\\vipdata\\vipproject\\webpack-source\\dist",
  "assetsByChunkName": {
    "lazy": "lazy.bundle.js",
    "main": "bundle.js"
  },
  "assets": [
    {
      "name": "bundle.js",
      "size": 9043,
      "chunks": [
        "main"
      ],
      "chunkNames": [
        "main"
      ],
      "emitted": true
    },
    {
      "name": "lazy.bundle.js",
      "size": 336,
      "chunks": [
        "lazy"
      ],
      "chunkNames": [
        "lazy"
      ],
      "emitted": true
    }
  ],
  "filteredAssets": 0,
  "entrypoints": {
    "main": {
      "chunks": [
        "main"
      ],
      "assets": [
        "bundle.js"
      ],
      "children": {},
      "childAssets": {}
    }
  },
  "namedChunkGroups": {
    "main": {
      "chunks": [
        "main"
      ],
      "assets": [
        "bundle.js"
      ],
      "children": {},
      "childAssets": {}
    },
    "lazy": {
      "chunks": [
        "lazy"
      ],
      "assets": [
        "lazy.bundle.js"
      ],
      "children": {},
      "childAssets": {}
    }
  },
  "chunks": [
    {
      "id": "lazy",
      "rendered": true,
      "initial": false,
      "entry": false,
      "size": 24,
      "names": [
        "lazy"
      ],
      "files": [
        "lazy.bundle.js"
      ],
      "hash": "d08a8b502d30324f81e1",
      "siblings": [],
      "parents": [
        "main"
      ],
      "children": [],
      "childrenByOrder": {},
      "modules": [
        {
          "id": "./src/lazy.js",
          "identifier": "C:\\vipdata\\vipproject\\webpack-source\\src\\lazy.js",
          "name": "./src/lazy.js",
          "index": 2,
          "index2": 2,
          "size": 24,
```

```json
        "cacheable": true,
        "built": true,
        "optional": false,
        "prefetched": false,
        "chunks": [
          "lazy"
        ],
        "issuer": "C:\\vipdata\\vipproject\\webpack-source\\src\\index.js",
        "issuerId": "./src/index.js",
        "issuerName": "./src/index.js",
        "issuerPath": [
          {
            "id": "./src/index.js",
            "identifier": "C:\\vipdata\\vipproject\\webpack-source\\src\\index.js",
            "name": "./src/index.js",
            "profile": {
              "factory": 18,
              "building": 14
            }
          }
        ],
        "profile": {
          "factory": 4,
          "building": 2
        },
        "failed": false,
        "errors": 0,
        "warnings": 0,
        "assets": [],
        "reasons": [
          {
            "moduleId": "./src/index.js",
            "moduleIdentifier": "C:\\vipdata\\vipproject\\webpack-source\\src\\index.js",
            "module": "./src/index.js",
            "moduleName": "./src/index.js",
            "type": "import()",
            "userRequest": "./lazy",
            "loc": "2:0-46"
          }
        ],
        "providedExports": null,
        "optimizationBailout": [],
        "depth": 1,
        "source": "module.exports = 'lazy';"
      }
    ],
    "filteredModules": 0,
    "origins": [
      {
        "moduleId": "./src/index.js",
        "module": "C:\\vipdata\\vipproject\\webpack-source\\src\\index.js",
        "moduleIdentifier": "C:\\vipdata\\vipproject\\webpack-source\\src\\index.js",
        "moduleName": "./src/index.js",
        "loc": "2:0-46",
        "request": "./lazy",
        "reasons": []
      }
    ]
  },
  {
    "id": "main",
    "rendered": true,
    "initial": true,
    "entry": true,
    "size": 162,
    "names": [
      "main"
    ],
    "files": [
      "bundle.js"
    ],
    "hash": "263cadc0459e8470151b",
    "siblings": [],
    "parents": [],
    "children": [
      "lazy"
    ],
    "childrenByOrder": {},
    "modules": [
      {
        "id": "./src/hello.js",
        "identifier": "C:\\vipdata\\vipproject\\webpack-source\\src\\hello.js",
        "name": "./src/hello.js",
        "index": 1,
        "index2": 0,
        "size": 25,
        "cacheable": true,
        "built": true,
        "optional": false,
        "prefetched": false,
        "chunks": [
          "main"
        ],
        "issuer": "C:\\vipdata\\vipproject\\webpack-source\\src\\index.js",
        "issuerId": "./src/index.js",
        "issuerName": "./src/index.js",
        "issuerPath": [
          {
            "id": "./src/index.js",
            "identifier": "C:\\vipdata\\vipproject\\webpack-source\\src\\index.js",
            "name": "./src/index.js",
            "profile": {
              "factory": 18,
              "building": 14
```

```json
                }
              }
            ],
            "profile": {
              "factory": 4,
              "building": 2
            },
            "failed": false,
            "errors": 0,
            "warnings": 0,
            "assets": [],
            "reasons": [
              {
                "moduleId": "./src/index.js",
                "moduleIdentifier": "C:\\vipdata\\vipproject\\webpack-source\\src\\index.js",
                "module": "./src/index.js",
                "moduleName": "./src/index.js",
                "type": "cjs require",
                "userRequest": "./hello",
                "loc": "1:12-30"
              }
            ],
            "providedExports": null,
            "optimizationBailout": [],
            "depth": 1,
            "source": "module.exports = 'hello';"
          },
          {
            "id": "./src/index.js",
            "identifier": "C:\\vipdata\\vipproject\\webpack-source\\src\\index.js",
            "name": "./src/index.js",
            "index": 0,
            "index2": 1,
            "size": 137,
            "cacheable": true,
            "built": true,
            "optional": false,
            "prefetched": false,
            "chunks": [
              "main"
            ],
            "issuer": null,
            "issuerId": null,
            "issuerName": null,
            "issuerPath": null,
            "profile": {
              "factory": 18,
              "building": 14
            },
            "failed": false,
            "errors": 0,
            "warnings": 0,
            "assets": [],
            "reasons": [
              {
                "moduleId": null,
                "moduleIdentifier": null,
                "module": null,
                "moduleName": null,
                "type": "single entry",
                "userRequest": "./src/index.js",
                "loc": "main"
              }
            ],
            "providedExports": null,
            "optimizationBailout": [],
            "depth": 0,
            "source": "let hello = require('./hello');\r\nimport(/* webpackChunkName: \"lazy\" */'./lazy').then(result=>{\r\n console.log(hello,resut.default)\r\n});"
          }
        ],
        "filteredModules": 0,
        "origins": [
          {
            "module": "",
            "moduleIdentifier": "",
            "moduleName": "",
            "loc": "main",
            "request": "./src/index.js",
            "reasons": []
          }
        ]
      }
    ],
    "modules": [
      {
        "id": "./src/hello.js",
        "identifier": "C:\\vipdata\\vipproject\\webpack-source\\src\\hello.js",
        "name": "./src/hello.js",
        "index": 1,
        "index2": 0,
        "size": 25,
        "cacheable": true,
        "built": true,
        "optional": false,
        "prefetched": false,
        "chunks": [
          "main"
        ],
        "issuer": "C:\\vipdata\\vipproject\\webpack-source\\src\\index.js",
        "issuerId": "./src/index.js",
        "issuerName": "./src/index.js",
        "issuerPath": [
          {
```

```json
          "id": "./src/index.js",
          "identifier": "C:\\vipdata\\vipproject\\webpack-source\\src\\index.js",
          "name": "./src/index.js",
          "profile": {
            "factory": 18,
            "building": 14
          }
        }
      ],
      "profile": {
        "factory": 4,
        "building": 2
      },
      "failed": false,
      "errors": 0,
      "warnings": 0,
      "assets": [],
      "reasons": [
        {
          "moduleId": "./src/index.js",
          "moduleIdentifier": "C:\\vipdata\\vipproject\\webpack-source\\src\\index.js",
          "module": "./src/index.js",
          "moduleName": "./src/index.js",
          "type": "cjs require",
          "userRequest": "./hello",
          "loc": "1:12-30"
        }
      ],
      "providedExports": null,
      "optimizationBailout": [],
      "depth": 1,
      "source": "module.exports = 'hello';"
    },
    {
      "id": "./src/index.js",
      "identifier": "C:\\vipdata\\vipproject\\webpack-source\\src\\index.js",
      "name": "./src/index.js",
      "index": 0,
      "index2": 1,
      "size": 137,
      "cacheable": true,
      "built": true,
      "optional": false,
      "prefetched": false,
      "chunks": [
        "main"
      ],
      "issuer": null,
      "issuerId": null,
      "issuerName": null,
      "issuerPath": null,
      "profile": {
        "factory": 18,
        "building": 14
      },
      "failed": false,
      "errors": 0,
      "warnings": 0,
      "assets": [],
      "reasons": [
        {
          "moduleId": null,
          "moduleIdentifier": null,
          "module": null,
          "moduleName": null,
          "type": "single entry",
          "userRequest": "./src/index.js",
          "loc": "main"
        }
      ],
      "providedExports": null,
      "optimizationBailout": [],
      "depth": 0,
      "source": "let hello = require('./hello');\r\nimport(/* webpackChunkName: \"lazy\" */'./lazy').then(result=>{\r\n console.log(hello,resut.default)\r\n});"
    },
    {
      "id": "./src/lazy.js",
      "identifier": "C:\\vipdata\\vipproject\\webpack-source\\src\\lazy.js",
      "name": "./src/lazy.js",
      "index": 2,
      "index2": 2,
      "size": 24,
      "cacheable": true,
      "built": true,
      "optional": false,
      "prefetched": false,
      "chunks": [
        "lazy"
      ],
      "issuer": "C:\\vipdata\\vipproject\\webpack-source\\src\\index.js",
      "issuerId": "./src/index.js",
      "issuerName": "./src/index.js",
      "issuerPath": [
        {
          "id": "./src/index.js",
          "identifier": "C:\\vipdata\\vipproject\\webpack-source\\src\\index.js",
          "name": "./src/index.js",
          "profile": {
            "factory": 18,
            "building": 14
          }
        }
      ],
      "errors": 0,
```

```
    "profile": {
      "factory": 4,
      "building": 2
    },
    "failed": false,
    "errors": 0,
    "warnings": 0,
    "assets": [],
    "reasons": [
      {
        "moduleId": "./src/index.js",
        "moduleIdentifier": "C:\\vipdata\\vipproject\\webpack-source\\src\\index.js",
        "module": "./src/index.js",
        "moduleName": "./src/index.js",
        "type": "import()",
        "userRequest": "./lazy",
        "loc": "2:0-46"
      }
    ],
    "providedExports": null,
    "optimizationBailout": [],
    "depth": 1,
    "source": "module.exports = 'lazy';"
  }
],
"filteredModules": 0,
"logging": {
  "webpack.buildChunkGraph.visitModules": {
    "entries": [],
    "filteredEntries": 5,
    "debug": false
  }
},
"children": []
}
```

4. 主要工作流程

- Webpack 的运行流程是一个串行的过程，从启动到结束会依次执行以下流程：
- 初始化参数：从配置文件和 Shell 语句中读取与合并参数，得出最终的参数；
- 开始编译：用上一步得到的参数初始化 Compiler 对象，加载所有配置的插件，执行对象的run方法开始执行编译； 确定入口：根据配置中的 entry 找出所有的入口文件
- 编译模块：从入口文件出发，调用所有配置的 Loader 对模块进行编译，再找出该模块依赖的模块，再递归本步骤直到所有入口依赖的文件都经过了本步骤的处理；
- 完成模块编译：在经过第4步使用 Loader 翻译完所有模块后，得到了每个模块被翻译后的最终内容以及它们之间的依赖关系；
- 输出资源：根据入口和模块之间的依赖关系，组装成一个个包含多个模块的 Chunk，再把每个 Chunk 转换成一个单独的文件加入到输出列表，这步是可以修改输出内容的最后机会；
- 输出完成：在确定好输出内容后，根据配置确定输出的路径和文件名，把文件内容写入到文件系统。
- 在以上过程中，Webpack 会在特定的时间点广播出特定的事件，插件在监听到感兴趣的事件后会执行特定的逻辑，并且插件可以调用 Webpack 提供的 API 改变 Webpack 的运行结果。