

资源输入输出

资源处理流程

我们需要指定一共或者多个入口 entry，告诉Webpack从源码目录下哪个文件开始打包

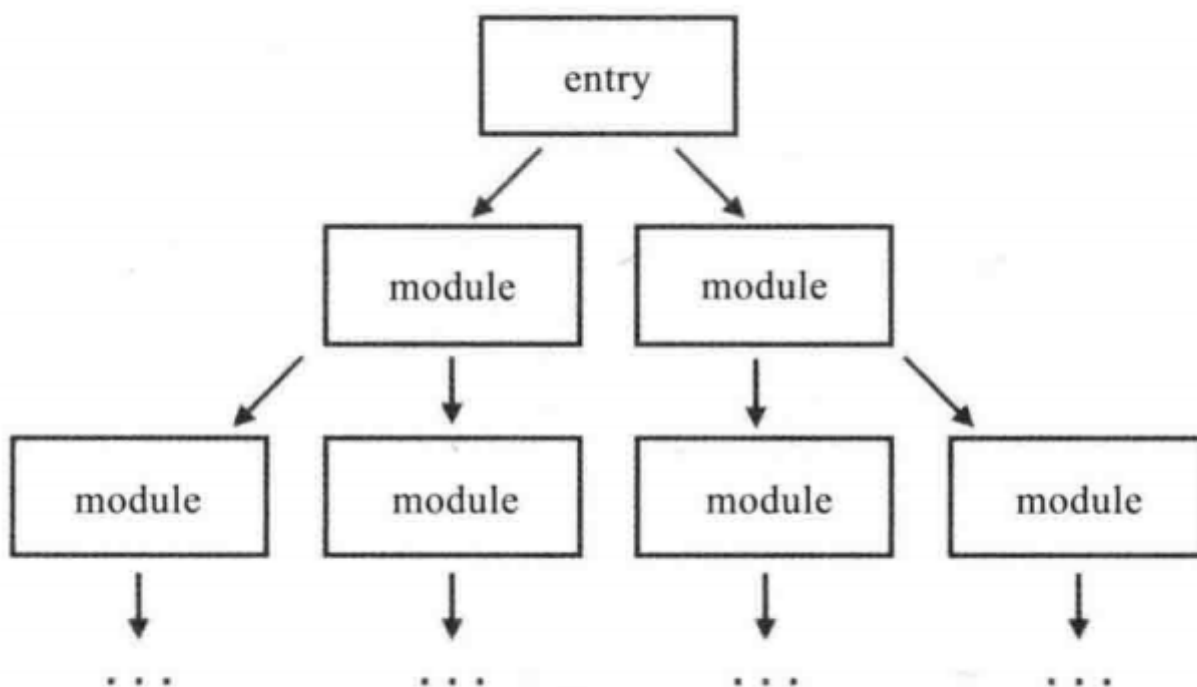
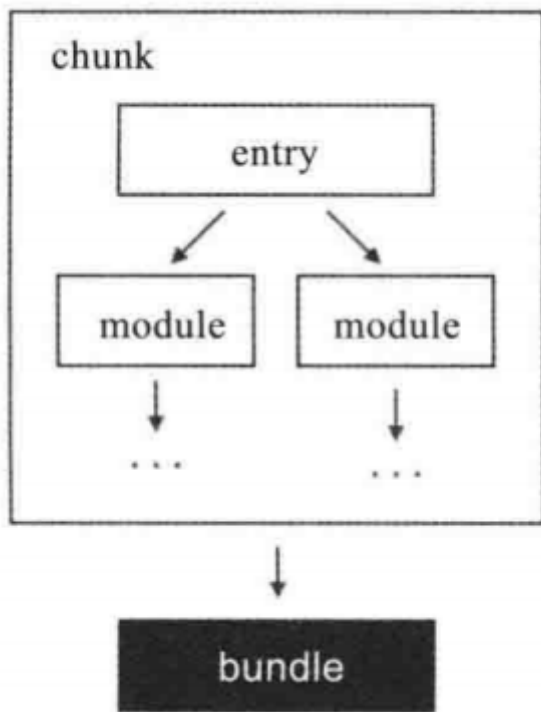


图 3-1 依赖关系树

这些依赖关系会在打包时被封装为一个chunk（代码块），即，被抽象和包装后的一些模块。根据配置，一个工程打包可能会产生一个或多个chunk



流程：

1. Webpack从入口文件开始检索
2. 把依赖关系的模型生成一颗依赖树，最终得到一个chunk
3. 由这个chunk得到的打包产物为bundle

每个入口会产生一个结果资源，entry与bundle存在对应关系

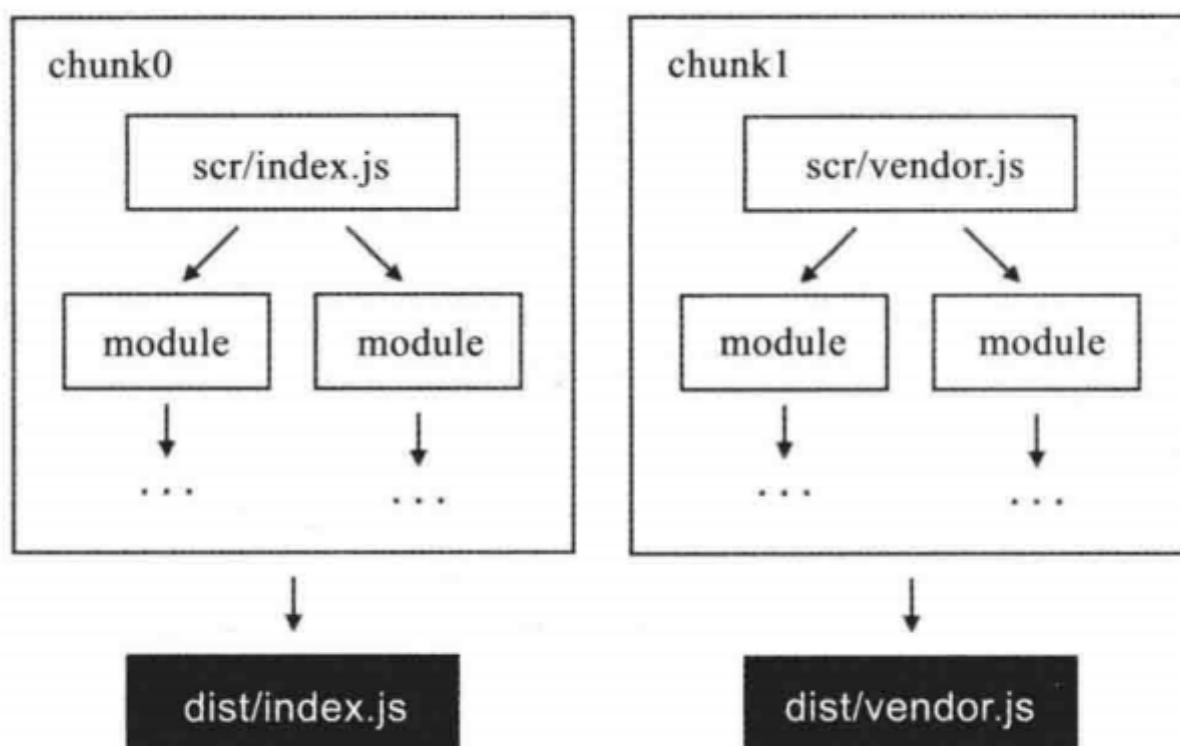


图 3-3 entry 与 bundle 的对应关系

配置资源入口

配置入口做了两件事

1. 确定入口模块位置，告诉Webpack从哪里开始打包
2. 定义chunk name
如果只有一个入口，默认chunk name 为'main';
如果有多个入口，需要为每个入口定义chunk name，作为chunk的唯一标识

content 和 entry两个配置来共同决定入口文件的路径

content

只能为字符串

资源入口的路径前缀，必须绝对路径，为了简化entry
默认为当前工程的根目录

entry

字符串，数组，对象，函数

```
// 字符串入口
entry: './src/index.js'
```

数组入口作用：多个资源预先合并，打包时最后一个元素作为实际入口路径

```
// 数组入口
entry:['babel-polyfill','./src/index.js']
// 等同于 在index中加入 import 'babel-polyfill'
```

对象类型入口：定义多入口，只能用对象形式，key是chunkname，value是入口路径
对象属性值也可以是字符串或者数组

```
// 对象类型入口
module.exports = {
  entry:{
    index:['babel-polyfill','./src/index.js']
    lib:'./src/lib.js'
  }
}
```

函数定义入口，只要返回值是上面的形式就即可

实例

单页应用(SAP)

SAP，一般定义单一入口即可

无论框架，库，还是每个页面的模块，都由app.js单一入口进行引用，只产生一个js文件。
所有模块打包到一起，会导致资源体积过大，降低用户页面渲染速度

webpack默认配置bundle大于250kb（压缩前）就会认为bundle过大，会警告

提取vendor

如果只生成一个JS文件，一旦代码更新，只有一点点改动，用户都要重新下载整个资源，对页面性能很不友好
使用vender，把工程所使用的库，框架等第三方模块集中打包而产生的bundle

```
module.exports = {
  content:path.join(__dirname,'./src'),
  entry:{
    app:"./src/app.js"
    vendor:['react','react-dom','react-router']
  }
}
```

我们在入口文件并没有为vendor设置入口路径，需要我们通过CommonsChunkPlugin（Webpack4已弃用，optimization.splitChunks）把app与vendor两个chunk公共模块提取出来。app只有业务模块。vendor则是第三方模块，一般不会变动，可以有效利用缓存

多页应用

每个页面加载自己必要的逻辑，多入口
最后再提取vendor将公共模块打包

配置资源出口

filename

输出资源文件名

路径不存在也可以，Webpack会自动创建

filename:'[name].js',[name]会被替换为chunkname

[hash],此次打包所所有资源生成的hash

[chunkhash],当前chunk内容的hash

[id],当前chunk的id

[query],filename配置项中的query

配置项作用：[hash]与[chunkhash]与内容相关，内容变了，值就变了，用这些做文件名有利于缓存命中

一般我们用[name]+[chunkhash]

path

资源输出位置

绝对路径

webpack4后默认为dist，webpack4之前需要专门配置

publicPath

指定资源的请求位置

js或者css所请求的间接资源路径

- Html
'./'或者"请求资源会以当前html路径加上相对路径
- Host
以'/'开始，请求资源会以当前host name为基础路径
- CDN
需要绝对路径形式指定

webpack-dev-server也有publicPath，指静态资源服务路径

小结

entry chunk bundle

context 路径前缀

单入口 chunk name 为 main

多入口 必需为每个chunk指定chunk name

vendor 提取
path
publicPath