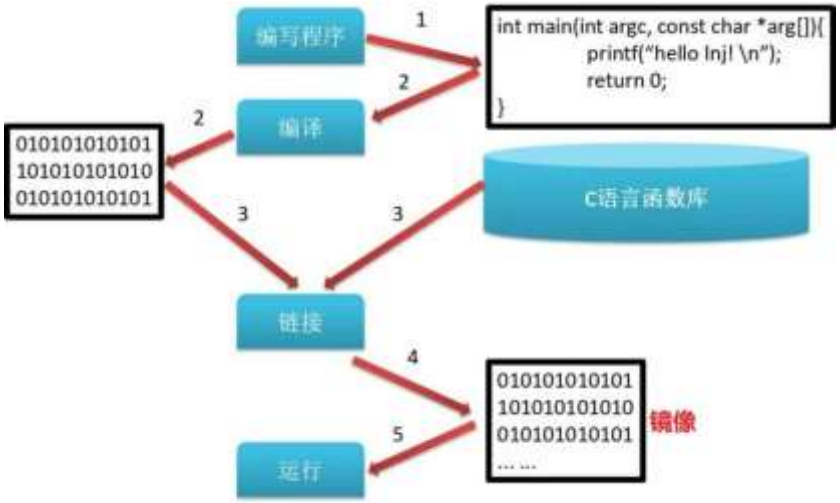
MDK 中分散文件使用

前言

因为嵌入式设备中存储器类型很多，我们需要在不同存储器中放置不同的代码或者数据。所以需要使用一种文件告诉链接器来完成这一操作。

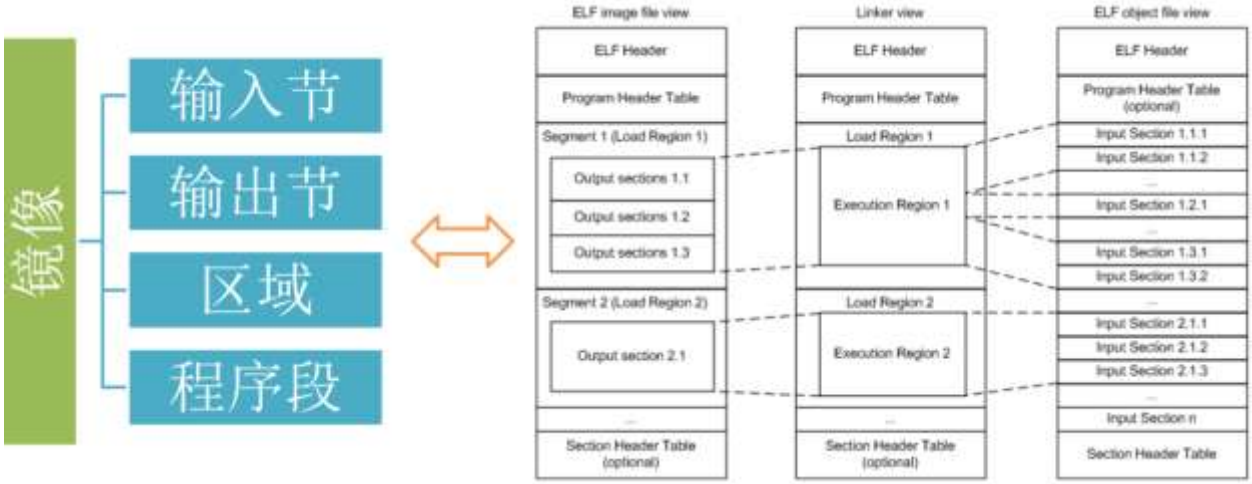
链接器与编译器区别

编译器只是完成代码的翻译工作，即把高级语言翻译成机器码。而链接器则是完成了组装工作，即把对象文件(相当于零件)按照要求打包成一个镜像文件(相当于产品)。



镜像文件组成

ARM ELF 镜像包含节，区域和段，并且每个链接阶段都有一个不同的镜像视图。



分散加载——加载域

//方括号中的为选填内容

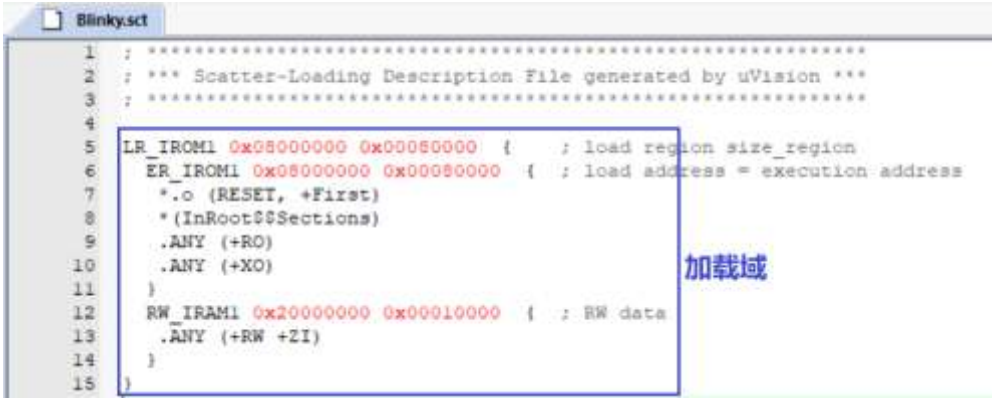
加载域名 (基地址 |("+" 地址偏移)) [属性列表] [最大容量]

"{"

执行区域描述+

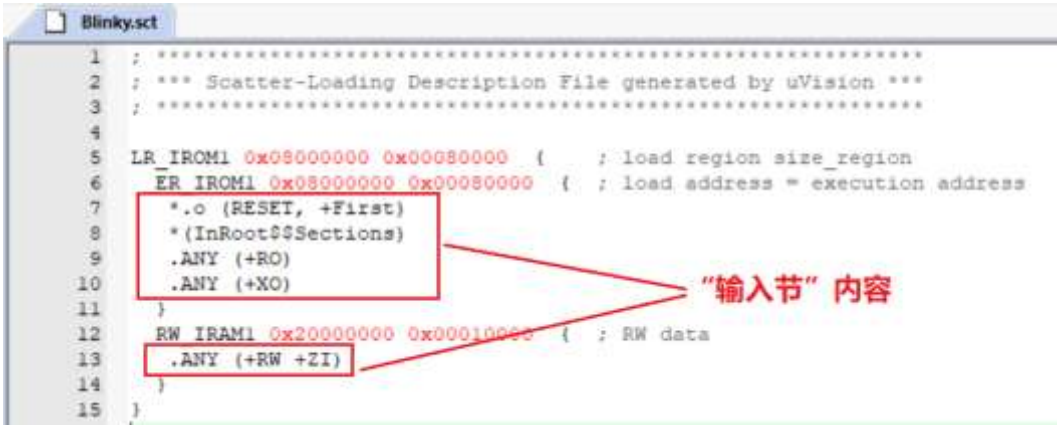
"}"

说明：当芯片的存在不连接的 flash 区域时，一个 scatter 文件中就有可能存在多个“加载域”。



分散加载——输入节

- 模块选择样式″(″输入节区样式″,″″+″输入节区属性″)″
- 模块选择样式″(″输入节区样式″,″″+″节区特性″)″
- 模块选择样式″(″输入符号样式″,″″+″节区特性″)″
- 模块选择样式″(″输入符号样式″,″″+″输入节区属性″)″



在 MDK 中添加编写好的 scatter 文件


Options for Target 'STM32F103 Flash'
Device | Target | Output | Listing | User | C/C++ | Asm | Linker | Debug | Utilities
Use Memory Layout from Target Dialog
Make RW Sections Position Independent
Make RO Sections Position Independent
Don't Search Standard Libraries
Report 'might fail' Conditions as Errors
X/O Base:
R/O Base:
R/W Base:
disable Warnings:
Scatter File
\Output\File_Demo.sct
Edit...
Misc controls
Linker control string
--cpu Cortex-M3 "o
--strict --scatter "\Output\File_Demo.sct"
OK
Cancel
Defaults
Help