提升100%编译性能

头文件搜索路径优化

嵌入式工程基于dep文件分析的构建优化

- 鱼丸ECU

头文件搜索拖慢编译的背景

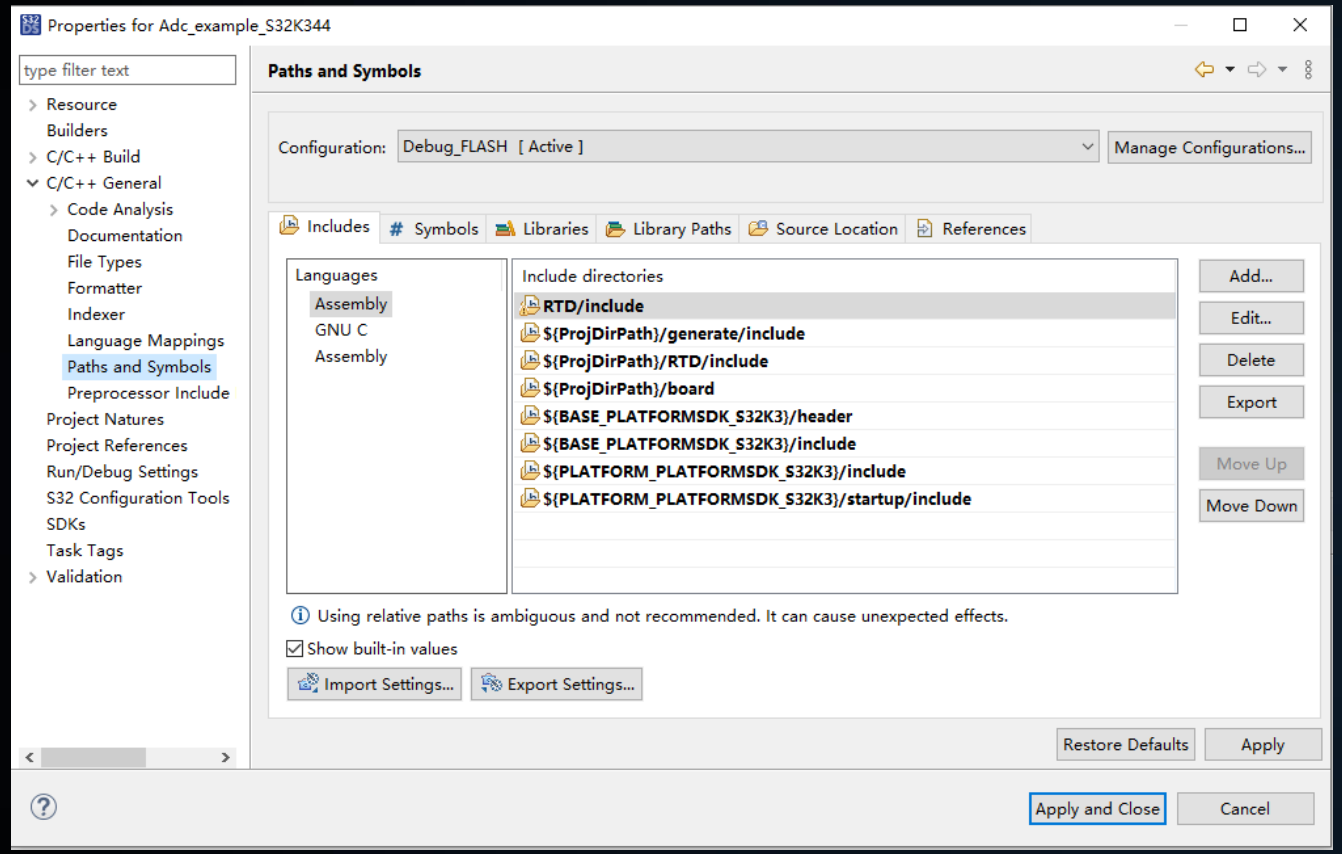
嵌入式项目

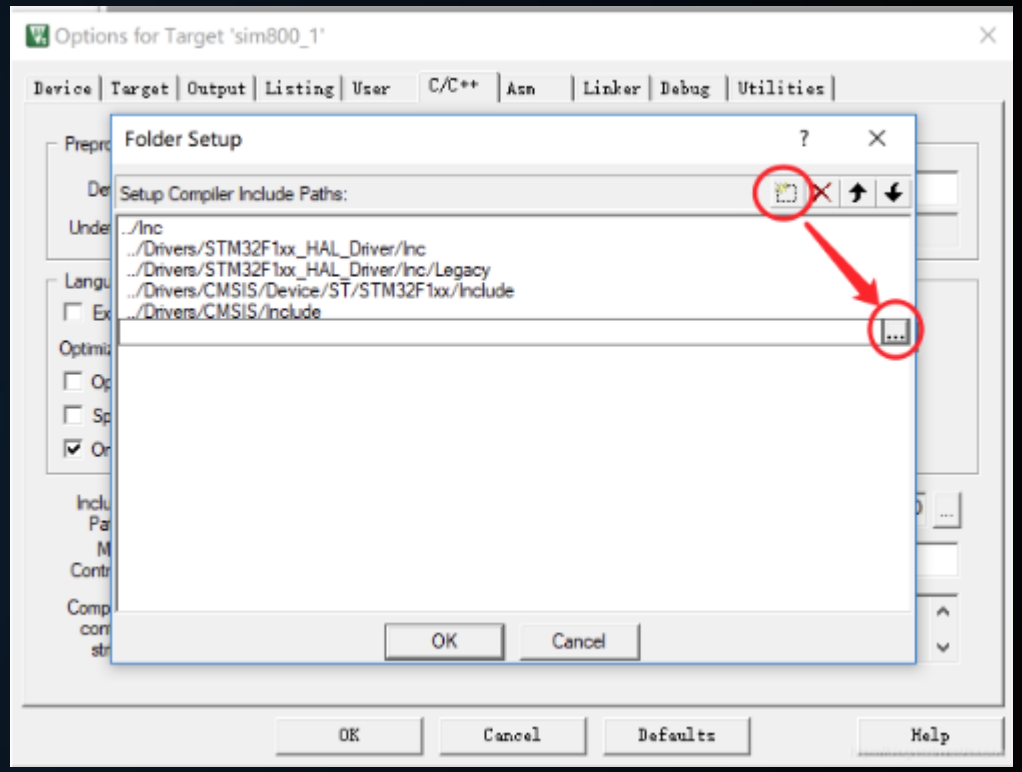
基于dep文件分析的解决方案

DEPENDENCY MAKEFILE

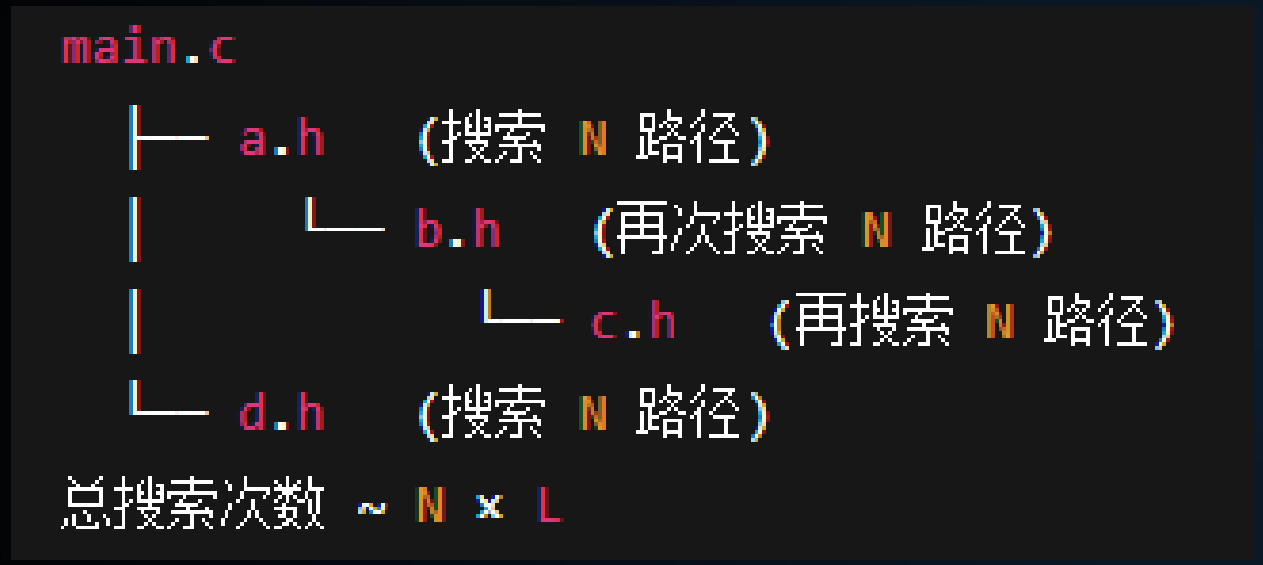
自动化解决方案演示

DEMO INC-STATS


Properties for Adc_example_S32K344
type filter text
Resource
Builders
C/C++ Build
C/C++ General
Code Analysis
Documentation
File Types
Formatter
Indexer
Language Mappings
Paths and Symbols
Preprocessor Include
Project Natures
Project References
Run/Debug Settings
S32 Configuration Tools
SDKs
Task Tags
Validation
Paths and Symbols
Configuration: Debug_FLASH [Active]
Manage Configurations...
Includes
Symbols
Libraries
Library Paths
Source Location
References
Languages
Assembly
GNU C
Assembly
Include directories
RTD/include
${ProjDirPath}/generate/include
${ProjDirPath}/RTD/include
${ProjDirPath}/board
${BASE_PLATFORMSDK_S32K3}/header
${BASE_PLATFORMSDK_S32K3}/include
${PLATFORM_PLATFORMSDK_S32K3}/include
${PLATFORM_PLATFORMSDK_S32K3}/startup/include
Add...
Edit...
Delete
Export
Move Up
Move Down
Using relative paths is ambiguous and not recommended. It can cause unexpected effects.
Show built-in values
Import Settings...
Export Settings...
Restore Defaults
Apply
Apply and Close
Cancel


Options for Target 'sim800_1'
Device
Target
Output
Listing
User
C/C++
Asm
Linker
Debug
Utilities
Folder Setup
Setup Compiler Include Paths:
../Inc
../Drivers/STM32F1xx_HAL_Driver/Inc
../Drivers/STM32F1xx_HAL_Driver/Inc/Legacy
../Drivers/CMSIS/Device/ST/STM32F1xx/Include
../Drivers/CMSIS/Include
OK
Cancel
OK
Cancel
Defaults
Help

头文件路径拖慢编译的背景



1. 现象（Phenomenon）

- 在 `.c/.cpp` 文件中使用 `#include` 引入头文件
- 编译器按 `-I path` 列表顺序查找头文件
- 若未找到，需要依次检查每个路径
- 头文件可能再次 `#include` 其他头文件 → 重复上述过程

2. 性能问题（Performance Problem）

- 假设搜索路径数 **N**，包含层级 **L**，则复杂度接近 **O(N × L)**
- 多文件编译时，这种开销成倍累积
- 每次查找都涉及**文件系统 I/O**，延迟显著
- 在大型项目中：
  - `-I` 路径可达几十甚至上百个
  - 包含层级常常 10+

3. 影响（Effect）

- 编译时间显著增长
- 磁盘 I/O 压力上升
- 增量编译效率下降
- 构建反馈周期变长，影响开发节奏

基于dep文件分析的解决方案 – 原理

.build > Adc_example_S32K344 > src > D main.d

```
.build/Adc_example_S32K344/src/main.o: \
 C\:/Users/yyd/Desktop/Make-Boost/demo1/Adc_example_S32K344/src/main.c \
 C\:/Users/yyd/Desktop/Make-Boost/demo1/Adc_example_S32K344/RTD/include/Adc.h \
 C\:/Users/yyd/Desktop/Make-Boost/demo1/Mcal/BaseNxp/include/Mcal.h \
 C\:/Users/yyd/Desktop/Make-Boost/demo1/Mcal/BaseNxp/include/Compiler.h \
 C\:/Users/yyd/Desktop/Make-Boost/demo1/Mcal/BaseNxp/include/Compiler_Cfg.h \
 C\:/Users/yyd/Desktop/Make-Boost/demo1/Mcal/BaseNxp/include/CompilerDefinition.h \
 C\:/Users/yyd/Desktop/Make-Boost/demo1/Mcal/BaseNxp/include/StandardTypes.h \
 C\:/Users/yyd/Desktop/Make-Boost/demo1/Mcal/BaseNxp/include/Platform_Types.h \
 C\:/Users/yyd/Desktop/Make-Boost/demo1/Mcal/BaseNxp/include/PlatformTypes.h \
 c\:\nxp\s32ds.3.5\s32ds\build_tools\gcc_v10.2\gcc-10.2-arm32-eabi\lib\gcc\arm-none-eabi\10.2.0\include\stdint.h \
 c\:\nxp\s32ds.3.5\s32ds\build_tools\gcc_v10.2\gcc-10.2-arm32-eabi\arm-none-eabi\include\stdint.h \
 c\:\nxp\s32ds.3.5\s32ds\build_tools\gcc_v10.2\gcc-10.2-arm32-eabi\arm-none-eabi\include\machine\_default_types.h \
 c\:\nxp\s32ds.3.5\s32ds\build_tools\gcc_v10.2\gcc-10.2-arm32-eabi\arm-none-eabi\include\sys\features.h \
 c\:\nxp\s32ds.3.5\s32ds\build_tools\gcc_v10.2\gcc-10.2-arm32-eabi\arm-none-eabi\include\_newlib_version.h \
 c\:\nxp\s32ds.3.5\s32ds\build_tools\gcc_v10.2\gcc-10.2-arm32-eabi\arm-none-eabi\include\sys\_intsup.h \
 c\:\nxp\s32ds.3.5\s32ds\build_tools\gcc_v10.2\gcc-10.2-arm32-eabi\arm-none-eabi\include\sys\_stdint.h \
 c\:\nxp\s32ds.3.5\s32ds\build_tools\gcc_v10.2\gcc-10.2-arm32-eabi\lib\gcc\arm-none-eabi\10.2.0\include\stdbool.h \
 c\:\nxp\s32ds.3.5\s32ds\build_tools\gcc_v10.2\gcc-10.2-arm32-eabi\lib\gcc\arm-none-eabi\10.2.0\include\stddef.h \
 C\:/Users/yyd/Desktop/Make-Boost/demo1/Adc_example_S32K344/generate/include/Soc_Ips.h \
 C\:/Users/yyd/Desktop/Make-Boost/demo1/Mcal/BaseNxp/include/Platform_Types.h \
 C\:/Users/yyd/Desktop/Make-Boost/demo1/Mcal/BaseNxp/include/IpVersionMacros.h \
 C\:/Users/yyd/Desktop/Make-Boost/demo1/Mcal/BaseNxp/include/Reg_eSys.h \
 C\:/Users/yyd/Desktop/Make-Boost/demo1/Mcal/BaseNxp/include/OsIf_Internal.h \
 C\:/Users/yyd/Desktop/Make-Boost/demo1/Adc_example_S32K344/generate/include/OsIf_Cfg.h \
 C\:/Users/yyd/Desktop/Make-Boost/demo1/Adc_example_S32K344/generate/include/OsIf_ArchCfg.h \
 C\:/Users/yyd/Desktop/Make-Boost/demo1/Mcal/BaseNxp/include/StandardTypes.h \
 C\:/Users/yyd/Desktop/Make-Boost/demo1/Mcal/BaseNxp/header/S32K344_SYSTICK.h \
 C\:/Users/yyd/Desktop/Make-Boost/demo1/Mcal/BaseNxp/header/S32K344_COMMON.h \
 C\:/Users/yyd/Desktop/Make-Boost/demo1/Mcal/BaseNxp/include/BasicTypes.h \
 C\:/Users/yyd/Desktop/Make-Boost/demo1/Adc_example_S32K344/generate/include/Adc_Cfg.h \
 C\:/Users/yyd/Desktop/Make-Boost/demo1/Adc_example_S32K344/generate/include/Adc_CfgDefines.h \
 C\:/Users/yyd/Desktop/Make-Boost/demo1/Adc_example_S32K344/RTD/include/Adc_Types.h \
 C\:/Users/yyd/Desktop/Make-Boost/demo1/Adc_example_S32K344/RTD/include/Adc_Ipw_Types.h \
 C\:/Users/yyd/Desktop/Make-Boost/demo1/Adc_example_S32K344/generate/include/Adc_Ipw_CfgDefines.h \
 C\:/Users/yyd/Desktop/Make-Boost/demo1/Adc_example_S32K344/RTD/include/Adc_Sar_Ip_Types.h \
 C\:/Users/yyd/Desktop/Make-Boost/demo1/Adc_example_S32K344/generate/include/Adc_Sar_Ip_CfgDefines.h \
```

将高频引用的头文件路径优先添加
让编译器能够高效查找高频头文件

1. Dependency (-MD) 文件
记录了obj依赖的所有头文件

2. 基于对每个Dep文件的分析
得到整个工程每个头文件路径被依赖次数

3. 按依赖次数降序重新导入编译头文件列表
使得高频头文件路径优先搜索

导入过程安全检查（头文件路径合法校验）
插入的路径必须本就存在于编译选项中
优化后的结果仅是调整了头文件列表顺序

基于dep文件分析的解决方案 – 实现

.build > Adc_example_S32K344 > src > D main.d

```
.build/Adc_example_S32K344/src/main.o: \
  C\:/Users/yyd/Desktop/Make-Boost/demo1/Adc_example_S32K344/src/main.c \
  C\:/Users/yyd/Desktop/Make-Boost/demo1/Adc_example_S32K344/RTD/include/Adc.h \
  C\:/Users/yyd/Desktop/Make-Boost/demo1/Mcal/BaseNxp/include/Mcal.h \
  C\:/Users/yyd/Desktop/Make-Boost/demo1/Mcal/BaseNxp/include/Compiler.h \
  C\:/Users/yyd/Desktop/Make-Boost/demo1/Mcal/BaseNxp/include/Compiler_Cfg.h \
  C\:/Users/yyd/Desktop/Make-Boost/demo1/Mcal/BaseNxp/include/CompilerDefinition.h \
  C\:/Users/yyd/Desktop/Make-Boost/demo1/Mcal/BaseNxp/include/StandardTypes.h \
  C\:/Users/yyd/Desktop/Make-Boost/demo1/Mcal/BaseNxp/include/Platform_Types.h \
  C\:/Users/yyd/Desktop/Make-Boost/demo1/Mcal/BaseNxp/include/PlatformTypes.h \
```

原始dep文件:
记录了obj依赖的所有头文件

.build > Adc_example_S32K344 > src > main.d.txt

```
C\:/Users/yyd/Desktop/Make-Boost/demo1/Adc_example_S32K344/RTD/include/
C\:/Users/yyd/Desktop/Make-Boost/demo1/Mcal/BaseNxp/include/
C\:/Users/yyd/Desktop/Make-Boost/demo1/Mcal/BaseNxp/include/
C\:/Users/yyd/Desktop/Make-Boost/demo1/Mcal/BaseNxp/include/
C\:/Users/yyd/Desktop/Make-Boost/demo1/Mcal/BaseNxp/include/
C\:/Users/yyd/Desktop/Make-Boost/demo1/Mcal/BaseNxp/include/
C\:/Users/yyd/Desktop/Make-Boost/demo1/Mcal/BaseNxp/include/
```

dep分析中间文件:
对每个依赖头文件提取路径

bin > _inc_stats_adc_demo.elf.txt

```
1900 Mcal/BaseNxp/include/
1126 Adc_example_S32K344/generate/include/
 909 Adc_example_S32K344/RTD/include/
 873 Mcal/BaseNxp/header/
   3 Mcal/PlatformNxp/startup/include/
   1 Adc_example_S32K344/include/
```

汇总所有中间文件:
提取相对路径 -> 去重(计数) -> 降序排列 -> 输出文件 -> 导入后续编译选项 🚀

基于dep文件分析的解决方案 – 实现

```
#
# $(MAKE) inc-stats
# |
# ├──> $(MAKE) inc-stats.phony
# |
# └──> $(MAKE) inc-stats.sort
#
```

BuildTools > make_file > scripts > $ collect_deps.sh

```
#!/bin/sh


# Usage: collect_deps.sh <dep_file.d> <output_file.d.txt>
# Extracts included header paths from a .d dependency file and writes them to a .txt file


# 使用 Perl 正则模式 (-P)，从输入文件 $1 中提取符合以下格式的路径片段：
# 正则说明：
#   \S*            :匹配连续的非空白字符（代表路径前缀）
#   [\\/]          :匹配一个正斜杠 / 或反斜杠 \，兼容 Windows/Linux 路径分隔符
#   (?=...)        :正向前瞻，确保后面是指定格式但不包括在输出中
#     [^\\/\s]+    :文件名（不能包含斜杠，反斜杠或空白）
#     \.[hH]       :以 .h 或 .H 结尾的头文件
#     (?:\s+(?!:)|$) :其后要么是空白 + 非冒号字符（避免冒号左侧表示 target），要么是行尾($)
#
# 总体作用：匹配以路径形式出现的 .h 或 .H 头文件路径（不包括文件名本身）
#           例如从 "src/util/mydir/file.h" 中提取 "src/util/mydir/"
# 如果 grep 失败（无匹配或出错），则仍创建空输出文件 $2，避免下游报错
#
grep -oP '\S*[\\/](?=[^\\/\s]+\.[hH](?:\s+(?!:)|$))' $1 > $2 || > $2
```

```
#=============================================================================================
#.....................................................include statistics
#=============================================================================================
inc-stats: | $(BUILD_DIR)
    @echo ''
    @echo '[Inc-stats]: removing intermediate dependency files (*.d.txt $(INC_STATS_FILE)) ...'
    @echo -n '............'
    find $(BUILD_DIR)/ -iname '*.d.txt' -delete
    @echo -n '............'
    rm -f '$(DST_DIR)/$(INC_STATS_FILE)'
    @echo -n '............'
    rm -f '$(BUILD_DIR)/$(INC_STATS_FILE)'
    @echo ''
    $(MAKE) -f $(THIS_MAKEFILE) --no-print-directory inc-stats.phony
    @echo ''
    @echo '[Inc-stats]: merging intermediate dependency files (*.d.txt) ...'
    @mkdir -p $(INC_STATS_PATH)/;.....................\
        find $(BUILD_DIR)/ -iname '*.d.txt' -exec cat {} + > \
        $(INC_STATS_PATH)/$(INC_STATS_FILE).tmp
    @echo ''
    $(MAKE) -f $(THIS_MAKEFILE) --no-print-directory inc-stats.sort

inc-stats.phony: $(addsuffix .txt, $(DEPS))

inc-stats.sort:
    @echo ''
    @echo '[Inc-stats]: ordering include path ...'
    @sort $(INC_STATS_PATH)/$(INC_STATS_FILE).tmp | uniq -c | sort -nr > \
        $(INC_STATS_PATH)/$(INC_STATS_FILE);.....................\
        rm $(INC_STATS_PATH)/$(INC_STATS_FILE).tmp;.....................\
        echo '[Inc-stats]: finished building $(INC_STATS_PATH)/$(INC_STATS_FILE)'
    @echo ''

%.d.txt: %.d
    @echo '[Inc-stats]: collecting $<'
    @$(BUILD_TOOLS_DIR)/make_file/scripts/collect_deps.sh $< $@

ifeq ($(MAKECMDGOALS),inc-stats.sort)
    INC_STATS_SORT := $(abspath $(subst \:,:,$(file < $(INC_STATS_PATH)/$(INC_STATS_FILE).tmp)))
    INC_STATS_SORT := $(patsubst $(CURDIR_WIN)/%,%/,$(filter $(CURDIR_WIN)/%,$(INC_STATS_SORT)))
    $(file > $(INC_STATS_PATH)/$(INC_STATS_FILE).tmp,$(subst $(SPACE),$(NEWLINE),$(INC_STATS_SORT)))
endif
```

自动化解决方案演示 – 生成

```
yyd@Desktop-Home-24 MINGW64 ~/Desktop/Make-Boost/demo1
$ build.bat inc-stats

execute C:\Users\yyd\Desktop\Make-Boost\demo1\build.bat
Usage: build.bat [make target]


C:\Users\yyd\Desktop\Make-Boost\demo1>BuildTools\binutils\shell-env\usr\bin\make -R -j16 -f Buil
dTools\make_file\makefile BUILD_PATH=".build/" SRC="Adc_example_S32K344/ Mcal/" SRC_OUT="" SRC_A
DD="" LD_FILE="Adc_example_S32K344/Project_Settings/Linker_Files/linker_flash_s32k344.ld" DST_PA
TH=bin/ DST_JSON= MCU=s32k344_gcc EXTRA_CC_ARGS="" EXTRA_AS_ARGS="" EXTRA_LD_ARGS="" TARGET=adc_
demo.elf inc-stats

[Inc-stats]: removing intermediate dependency files (*.d.txt _inc_stats_adc_demo.elf.txt) ...
             find .build/ -iname '*.d.txt' -delete
             rm -f 'bin/_inc_stats_adc_demo.elf.txt'
             rm -f '.build/_inc_stats_adc_demo.elf.txt'

C:/Users/yyd/Desktop/Make-Boost/demo1/BuildTools/binutils/shell-env/usr/bin/make -f BuildTools/m
ake_file/makefile --no-print-directory inc-stats.phony

[Scanning sources]: out source files ...
[Scanning sources]:  .h source files ...
[Scanning sources]:  .s source files ...
[Scanning sources]:  .c source files ...
[Scanning sources]:  .a source files ...
[Ordering incstats]: 0 / 0 / 7 (stats_add / stats_raw / path_raw)
[Finished scanning]

[Inc-stats]: collecting .build/Adc_example_S32K344/Project_Settings/Startup_Code/Vector_Table.d
[Inc-stats]: collecting .build/Adc_example_S32K344/Project_Settings/Startup_Code/nvic.d
[Inc-stats]: collecting .build/Adc_example_S32K344/RTD/src/Adc_Ipw.d
[Inc-stats]: collecting .build/Adc_example_S32K344/RTD/src/Adc_Ipw_Irq.d
[Inc-stats]: collecting .build/Adc_example_S32K344/RTD/src/Adc_Sar_Ip.d
```

集成到 AutoMake 构建工具中自动构建出统计信息文件
build.bat inc-stats

```
[Inc-stats]: collecting .build/Adc_example_S32K344/generate/src/Platform_Cfg.d
[Inc-stats]: collecting .build/Adc_example_S32K344/generate/src/Pflash_Ip_VS_0_PBcfg.d
[Inc-stats]: collecting .build/Adc_example_S32K344/generate/src/Platform_Ipw_Cfg.d
[Inc-stats]: collecting .build/Adc_example_S32K344/generate/src/Power_Ip_Cfg.d
[Inc-stats]: collecting .build/Adc_example_S32K344/generate/src/Power_Ip_VS_0_PBcfg.d
[Inc-stats]: collecting .build/Adc_example_S32K344/generate/src/Ram_Ip_Cfg.d
[Inc-stats]: collecting .build/Adc_example_S32K344/generate/src/Ram_Ip_VS_0_PBcfg.d
[Inc-stats]: collecting .build/Adc_example_S32K344/generate/src/Sdadc_Ip_VS_0_PBcfg.d
[Inc-stats]: collecting .build/Adc_example_S32K344/generate/src/Trgmux_Ip_Cfg.d
[Inc-stats]: collecting .build/Adc_example_S32K344/generate/src/Virt_Wrapper_Ip_VS_0_PBcfg.d
[Inc-stats]: collecting .build/Adc_example_S32K344/generate/src/Xrdc_Ip_VS_0_PBcfg.d
[Inc-stats]: collecting .build/Adc_example_S32K344/generate/src/Xbic_Ip_VS_0_PBcfg.d
[Inc-stats]: collecting .build/Adc_example_S32K344/generate/src/Trgmux_Ip_VS_0_PBcfg.d
[Inc-stats]: collecting .build/Adc_example_S32K344/generate/src/Xrdc_Ip_Cfg.d
[Inc-stats]: collecting .build/Adc_example_S32K344/src/main.d

[Inc-stats]: merging intermediate dependency files (*.d.txt) ...

C:/Users/yyd/Desktop/Make-Boost/demo1/BuildTools/binutils/shell-env/usr/bin/make -f BuildToo
ls/make_file/makefile --no-print-directory inc-stats.sort

[Inc-stats]: ordering include path ...
[Inc-stats]: finished building bin/_inc_stats_adc_demo.elf.txt

Make Successed!
```

自动化解决方案演示 – 导入

bin > _inc_stats_adc_demo.elf.txt

```
1900 Mcal/BaseNxp/include/
1126 Adc_example_S32K344/generate/include/
 909 Adc_example_S32K344/RTD/include/
 873 Mcal/BaseNxp/header/
   3 Mcal/PlatformNxp/startup/include/
   1 Adc_example_S32K344/include/
```

.build > _var_I_PATH.txt

```
-IC:/Users/yyd/Desktop/Make-Boost/demo1/Adc_example_S32K344/RTD/include/
-IC:/Users/yyd/Desktop/Make-Boost/demo1/Adc_example_S32K344/generate/include/
-IC:/Users/yyd/Desktop/Make-Boost/demo1/Adc_example_S32K344/include/
-IC:/Users/yyd/Desktop/Make-Boost/demo1/Mcal/BaseNxp/header/
-IC:/Users/yyd/Desktop/Make-Boost/demo1/Mcal/BaseNxp/include/
-IC:/Users/yyd/Desktop/Make-Boost/demo1/Mcal/PlatformNxp/include/
-IC:/Users/yyd/Desktop/Make-Boost/demo1/Mcal/PlatformNxp/startup/include/
```

.build > _var_I_PATH.txt

```
-IC:/Users/yyd/Desktop/Make-Boost/demo1/Mcal/BaseNxp/include/
-IC:/Users/yyd/Desktop/Make-Boost/demo1/Adc_example_S32K344/generate/include/
-IC:/Users/yyd/Desktop/Make-Boost/demo1/Adc_example_S32K344/RTD/include/
-IC:/Users/yyd/Desktop/Make-Boost/demo1/Mcal/BaseNxp/header/
-IC:/Users/yyd/Desktop/Make-Boost/demo1/Mcal/PlatformNxp/startup/include/
-IC:/Users/yyd/Desktop/Make-Boost/demo1/Adc_example_S32K344/include/
-IC:/Users/yyd/Desktop/Make-Boost/demo1/Mcal/PlatformNxp/include/
```

头文件路径统计信息 导入后续编译选项

打印导入日志:

[Ordering incstats]: 6 / 6 / 7 (stats_add / stats_raw / path_raw)

6 / 6 : 条存在于统计信息中的路径全部都插队成功

1(7-6): 条路径miss，因为该条路径下的h文件实际没有被任何c文件include
但因为该路径下存在h文件，构建系统仍然会添加

性能提升说明

1. 基于 112核心服务器 对包含了1700+obj 工程构建 3m40s -> 1m12s
2. 基于 PC(9700x) 对包含了700+头文件的 单源文件编译 2.0s -> 1.1s

特性 / 维度	现代编译器（Clang、GCC 新版本、MSVC 最新版）	传统编译器（Keil、IAR、老版本 GCC）
头文件缓存机制	✅ 增量编译缓存（Clang PCH、GCC PCH、Modules）	❌ 无缓存或仅有限 PCH 支持
缓存存储位置	🟡 本地磁盘缓存（可跨编译进程共享）+ 🧠 内存缓存（单进程）	🧠 仅内存缓存（单进程，退出即丢失）
跨进程缓存共享	✅ 支持（磁盘缓存文件复用）	❌ 不支持
首次构建速度	⏳ 与传统相近（需建立缓存）	⏳ 较慢（全量解析）
重复构建速度	⚡ 显著提升（命中缓存跳过解析）	🐌 无提升（重复解析）
并行编译性能	🚀 高效（多进程共享缓存）	🐌 重复消耗 CPU 与 I/O
模块化支持	🧩 完善（C++20 Modules、Clang Modules）	❌ 无模块化支持
典型表现	📈 大项目仍可快速增量编译	📉 项目大 → 编译速度明显下降