



(12) 发明专利

(10) 授权公告号 CN 110471663 B

(45) 授权公告日 2024. 01. 02

(21) 申请号 201810450390.0

(22) 申请日 2018.05.11

(65) 同一申请的已公布的文献号

申请公布号 CN 110471663 A

(43) 申请公布日 2019.11.19

(73) 专利权人 杭州海康威视数字技术股份有限公司

地址 310051 浙江省杭州市滨江区阡陌路
555号

(72) 发明人 齐斌

(74) 专利代理机构 北京柏杉松知识产权代理事
务所(普通合伙) 11413

专利代理师 马敬 项京

(51) Int. Cl.

G06F 8/41 (2018.01)

(56) 对比文件

CN 104714788 A, 2015.06.17

CN 101963915 A, 2011.02.02

CN 101697135 A, 2010.04.21

CN 102955698 A, 2013.03.06

US 2010/0205591 A1, 2010.08.12

审查员 魏晨蕾

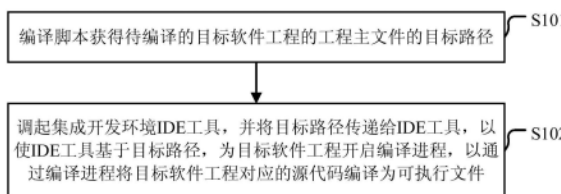
权利要求书2页 说明书10页 附图1页

(54) 发明名称

一种代码编译方法、装置及电子设备

(57) 摘要

本发明实施例提供了一种代码编译方法、装置及电子设备,该方法包括:编译脚本获得待编译的目标软件工程的工程主文件的目标路径;调起集成开发环境IDE工具,并将所述目标路径传递给所述IDE工具,以使所述IDE工具基于所述目标路径,为所述目标软件工程开启编译进程,以通过所述编译进程将所述目标软件工程对应的源代码编译为可执行文件。应用本发明实施例提供的方案可以提高编译效率。



1. 一种代码编译方法,其特征在于,应用于编译脚本,所述方法包括:

编译脚本获得待编译的目标软件工程的工程主文件的目标路径;所述编译脚本存储在所述目标软件工程所在的根目录下;其中,所述待编译的目标软件工程为多个;

调起集成开发环境IDE工具,并将所述目标路径传递给所述IDE工具,以使所述IDE工具基于所述目标路径,为所述目标软件工程开启编译进程,以通过所述编译进程将所述目标软件工程对应的源代码编译为可执行文件;

所述编译脚本获得待编译的目标软件工程的工程主文件的目标路径的步骤,包括:

编译脚本获得所述编译脚本所在的根目录;根据所述根目录,生成所述根目录下的所述目标软件工程的工程主文件的目标路径;

所述将所述目标路径传递给所述IDE工具,以使所述IDE工具基于所述目标路径,为所述目标软件工程开启编译进程,以通过所述编译进程将所述目标软件工程对应的源代码编译为可执行文件的步骤,包括:

将各个目标软件工程的目標路径传递给所述IDE工具,以使所述IDE工具基于各个目标软件工程的目標路径,分别为各个目标软件工程开启编译进程,以通过各个编译进程并行将各个目标软件工程对应的源代码编译为可执行文件。

2. 根据权利要求1所述的方法,其特征在于,所述方法还包括:

根据所述目标软件工程对应的编译日志文件中的第一特征字段,确定所述目标软件工程的编译状态,其中,所述第一特征字段为:用于描述软件工程的编译状态的字段。

3. 根据权利要求2所述的方法,其特征在于,所述方法还包括:

在确定所述目标软件工程的编译状态为编译完成时,根据所述目标软件工程对应的编译日志文件中的第二特征字段,确定所述目标软件工程的编译结果,其中,所述第二特征字段为:用于描述软件工程的编译结果的字段。

4. 根据权利要求3所述的方法,其特征在于,所述方法还包括:

基于所述目标软件工程的编译结果,输出针对所述目标软件工程的编译结果汇总信息。

5. 一种代码编译装置,其特征在于,应用于编译脚本,所述装置包括:

获得模块,用于编译脚本获得待编译的目标软件工程的工程主文件的目标路径;所述编译脚本存储在所述目标软件工程所在的根目录下;

编译模块,用于调起集成开发环境IDE工具,并将所述目标路径传递给所述IDE工具,以使所述IDE工具基于所述目标路径,为所述目标软件工程开启编译进程,以通过所述编译进程将所述目标软件工程对应的源代码编译为可执行文件;其中,所述待编译的目标软件工程为多个;

所述获得模块,包括:

获得子模块,用于编译脚本获得所述编译脚本所在的根目录;

生成子模块,用于根据所述根目录,生成所述根目录下的所述目标软件工程的工程主文件的目标路径;

所述编译模块将所述目标路径传递给所述IDE工具,以使所述IDE工具基于所述目标路径,为所述目标软件工程开启编译进程,以通过所述编译进程将所述目标软件工程对应的源代码编译为可执行文件的步骤,具体为:

将各个目标软件工程的目标路径传递给所述IDE工具,以使所述IDE工具基于各个目标软件工程的目标路径,分别为各个目标软件工程开启编译进程,以通过各个编译进程并行将各个目标软件工程对应的源代码编译为可执行文件。

6.一种电子设备,其特征在于,包括处理器、通信接口、存储器和通信总线,其中,处理器,通信接口,存储器通过通信总线完成相互间的通信;

存储器,用于存放计算机程序;

处理器,用于执行存储器上所存放的程序时,实现权利要求1-4任一所述的方法步骤。

一种代码编译方法、装置及电子设备

技术领域

[0001] 本发明涉及软件程序开发技术领域,特别是涉及一种代码编译方法、装置及电子设备。

背景技术

[0002] 在软件程序开发领域,编译是不可或缺的重要环节。编译就是将以高级语言编写的代码转换为计算机可识别二进制代码。目前,在嵌入式软件工程开发领域,采用嵌入式芯片厂商提供的集成开发环境(IDE)进行代码编译,即IDE工具通过调用内部编译器把源代码编译为可执行文件,其中,可执行文件中的代码为计算机可识别二进制代码。

[0003] 以单片机工程为例,对单片机工程的源代码进行编译时,采用的是手动在IDE工具中打开单片机工程的工程主文件从而实现对源代码进行编译的方式。这种方式需要人工参与,编译效率较低,尤其是当需要对源代码进行编译的单片机工程数量较多时,这种方式由于编译效率低,难以满足开发需求。

发明内容

[0004] 本发明实施例的目的在于提供一种代码编译方法、装置及电子设备,以提高编译效率。具体技术方案如下:

[0005] 第一方面,本发明实施例提供了一种代码编译方法,所述方法包括:

[0006] 编译脚本获得待编译的目标软件工程的工程主文件的目标路径;

[0007] 调起集成开发环境IDE工具,并将所述目标路径传递给所述IDE工具,以使所述IDE工具基于所述目标路径,为所述目标软件工程开启编译进程,以通过所述编译进程将所述目标软件工程对应的源代码编译为可执行文件。

[0008] 可选的,所述待编译的目标软件工程为多个;

[0009] 所述将所述目标路径传递给所述IDE工具,以使所述IDE工具基于所述目标路径,为所述目标软件工程开启编译进程,以通过所述编译进程将所述目标软件工程对应的源代码编译为可执行文件的步骤,包括:

[0010] 将各个目标软件工程的目标路径传递给所述IDE工具,以使所述IDE工具基于各个目标软件工程的目标路径,分别为各个目标软件工程开启编译进程,以通过各个编译进程并行将各个目标软件工程对应的源代码编译为可执行文件。

[0011] 可选的,所述编译脚本存储在所述目标软件工程所在的根目录下;

[0012] 所述编译脚本获得待编译的目标软件工程的工程主文件的目标路径的步骤,包括:

[0013] 编译脚本获得所述编译脚本所在的根目录;

[0014] 根据所述根目录,生成所述根目录下的所述目标软件工程的工程主文件的目标路径。

[0015] 可选的,所述方法还包括:

[0016] 根据所述目标软件工程对应的编译日志文件中的第一特征字段,确定所述目标软件工程的编译状态,其中,所述第一特征字段为:用于描述软件工程的编译状态的字段。

[0017] 可选的,所述方法还包括:

[0018] 在确定所述目标软件工程的编译状态为编译完成时,根据所述目标软件工程对应的编译日志文件中的第二特征字段,确定所述目标软件工程的编译结果,其中,所述第二特征字段为:用于描述软件工程的编译结果的字段。

[0019] 可选的,所述方法还包括:

[0020] 基于所述目标软件工程的编译结果,输出针对所述目标软件工程的编译结果汇总信息。

[0021] 第二方面,本发明实施例还提供了一种代码编译装置,所述装置包括:

[0022] 获得模块,用于编译脚本获得待编译的目标软件工程的工程主文件的目标路径;

[0023] 编译模块,用于调起集成开发环境IDE工具,并将所述目标路径传递给所述IDE工具,以使所述IDE工具基于所述目标路径,为所述目标软件工程开启编译进程,以通过所述编译进程将所述目标软件工程对应的源代码编译为可执行文件。

[0024] 可选的,所述待编译的目标软件工程为多个;

[0025] 所述编译模块将所述目标路径传递给所述IDE工具,以使所述IDE工具基于所述目标路径,为所述目标软件工程开启编译进程,以通过所述编译进程将所述目标软件工程对应的源代码编译为可执行文件的步骤,具体为:

[0026] 将各个目标软件工程的目标路径传递给所述IDE工具,以使所述IDE工具基于各个目标软件工程的目标路径,分别为各个目标软件工程开启编译进程,以通过各个编译进程并行将各个目标软件工程对应的源代码编译为可执行文件。

[0027] 可选的,所述编译脚本存储在所述目标软件工程所在的根目录下;

[0028] 所述获得模块,包括:

[0029] 获得子模块,用于编译脚本获得所述编译脚本所在的根目录;

[0030] 生成子模块,用于根据所述根目录,生成所述根目录下的所述目标软件工程的工程主文件的目标路径。

[0031] 可选的,所述装置还包括:

[0032] 第一确定模块,用于根据所述目标软件工程对应的编译日志文件中的第一特征字段,确定所述目标软件工程的编译状态,其中,所述第一特征字段为:用于描述软件工程的编译状态的字段。

[0033] 可选的,所述装置还包括:

[0034] 第二确定模块,用于在确定所述目标软件工程的编译状态为编译完成时,根据所述目标软件工程对应的编译日志文件中的第二特征字段,确定所述目标软件工程的编译结果,其中,所述第二特征字段为:用于描述软件工程的编译结果的字段。

[0035] 可选的,所述装置还包括:

[0036] 输出模块,用于基于所述目标软件工程的编译结果,输出针对所述目标软件工程的编译结果汇总信息。

[0037] 第三方面,本发明实施例还提供了一种电子设备,包括处理器、通信接口、存储器和通信总线,其中,处理器,通信接口,存储器通过通信总线完成相互间的通信;

[0038] 存储器,用于存放计算机程序;

[0039] 处理器,用于执行存储器上所存放的程序时,实现第一方面所述的代码编译方法。

[0040] 第四方面,本发明实施例提供了一种计算机可读存储介质,所述计算机可读存储介质内存储有计算机程序,所述计算机程序被处理器执行时实现第一方面所述的代码编译方法。

[0041] 本发明实施例提供的方案,编译脚本获得待编译的目标软件工程的工程主文件的目标路径,然后调起集成开发环境IDE工具,并将目标路径传递给IDE工具,以使IDE工具基于目标路径,为目标软件工程开启编译进程,以通过编译进程将目标软件工程对应的源代码编译为可执行文件。可见,本发明实施例提供的方案,利用编译脚本进行自动化编译,不需要人工参与,由编译脚本直接调起IDE工具并将待编译的目标软件工程的工程主文件的目标路径传递给IDE工具,IDE工具基于目标路径为目标软件工程开启编译进程,从而实现对目标软件工程的源代码的编译,提高了编译效率。当然,实施本发明的任一产品或方法必不一定需要同时达到以上所述的所有优点。

附图说明

[0042] 为了更清楚地说明本发明实施例或现有技术中的技术方案,下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图仅仅是本发明的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图获得其他的附图。

[0043] 图1为本发明实施例提供的一种代码编译方法的流程示意图;

[0044] 图2为本发明实施例提供的一种代码编译装置的结构示意图;

[0045] 图3为本发明实施例提供的一种电子设备的结构示意图。

具体实施方式

[0046] 下面将结合本发明实施例中的附图,对本发明实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例仅仅是本发明一部分实施例,而不是全部的实施例。基于本发明中的实施例,本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例,都属于本发明保护的范围。

[0047] 为了便于理解方案,首先对本发明实施例所提供方案所涉及的技术术语进行介绍。

[0048] 脚本:是指使用一种特定的描述性语言,依据一定的格式编写的可执行文件。在Windows平台下,脚本文件本质上就是一个批处理文件(.bat或者.cmd),点击脚本文件,可以直接运行,弹出的窗口类似于DOS(Disk Operating System,磁盘操作系统)界面。

[0049] 编译:是指利用编译程序把源代码程序转换为可执行文件的过程。

[0050] 多进程:是指操作系统中正在运行的程序的多个实例。

[0051] 单片机:是指一种集成电路芯片,这种芯片把CPU(Central Processing Unit,中央处理器)、RAM(Random Access Memory,随机存取存储器)、ROM(Read Only Memory,只读存储器)、多种I/O(Input/Output,输入/输出)接口和中断系统、定时器、A/D转换器(analog to digital converter,模数转换器)等功能集成到一块硅片上构成的一个小而完善的微

型计算机系统。

[0052] IDE工具：是指在集成开发环境中可以辅助开发程序的应用软件，如Keil uVision、Code Composed Studio、ST Visual Develop等。

[0053] 为提高编译效率，本发明实施例提供了一种代码编译方法、装置及电子设备。

[0054] 下面首先对本发明实施例提供的一种代码编译方法进行介绍。

[0055] 需要说明的是，本发明实施例提供的代码编译方法的执行主体，可以为编译脚本，其中，编译脚本运行于电子设备中。在具体应用中，该电子设备可以为终端设备或服务器，这都是合理的。

[0056] 本发明实施例中，编译脚本和IDE工具可以运行于windows操作系统中，也可以运行于其他类型的操作系统，本发明实施例对此不做限定。并且，本发明实施例中待编译的目标软件工程可以为单片机工程，也可以为其它类型的工程，本发明实施例对此亦不做限定。

[0057] 参见图1所示，本发明实施例提供的一种代码编译方法，包括如下步骤：

[0058] S101，编译脚本获得待编译的目标软件工程的工程主文件的目标路径。

[0059] 本领域技术人员可以理解的是，一个软件工程对应一个文件夹，该文件夹下有一个工程主文件，工程主文件中存储着软件工程的所有配置信息，如软件工程对应的源代码的存储位置信息。因此，当要对软件工程的源代码进行编译时，需要在IDE工具中打开工程主文件，从而对源代码进行编译。

[0060] 另外，软件工程的工程主文件是利用IDE工具创建的，对于某一IDE工具而言，该IDE工具所创建的工程主文件即为该IDE工具可识别的类型的文件，例如，若IDE工具为Keil uVision，则工程主文件可以是后缀为.uvproj的文件。

[0061] 为了实现对软件工程的编译，该编译脚本可以获得待编译的目标软件工程的工程主文件的目标路径，进而利用该目标路径执行后续的步骤。可以理解的是，用户在磁盘上寻找某一文件时所历经的文件夹线路叫该文件的路径，也就是说，磁盘上的任一个文件均有其对应的路径。因此，当该编译脚本确定待编译的目标软件工程之后，可以确定出目标软件工程的工程主文件的目标路径。

[0062] 在一种实现方式中，目标软件工程可以为预先指定的待编译的软件工程，则编译脚本可以获得目标软件工程的文件夹在磁盘中的存储位置，进而获得目标软件工程的工程主文件的目标路径。

[0063] 在另一种实现方式中，编译脚本可以存储在目标软件工程所在的根目录下；

[0064] 在这种情况下，上述编译脚本获得待编译的目标软件工程的工程主文件的目标路径的步骤(S101)，可以包括：

[0065] 编译脚本获得编译脚本所在的根目录；

[0066] 根据根目录，生成根目录下的目标软件工程的工程主文件的目标路径。

[0067] 可以理解的是，这种实现方式中，与编译脚本处于同一根目录下的软件工程可以被确定为目标软件工程。则，将目标软件工程对应的文件夹名称以及工程主文件对应的文件名称添加至根目录中，即可获得根目录下的目标软件工程的工程主文件的目标路径。

[0068] 例如，编译脚本所在的根目录的绝对路径为root_path，目标软件工程对应的文件夹名称为：D0_100VB，目标软件工程的工程主文件对应的名称为：D0_100VB.uvproj，则该目标软件工程的工程主文件的目标路径可以表示为：%root_path%D0_100VB/D0_

100VB.uvproj。

[0069] S102,调起集成开发环境IDE工具,并将目标路径传递给IDE工具,以使IDE工具基于目标路径,为目标软件工程开启编译进程,以通过编译进程将目标软件工程对应的源代码编译为可执行文件。

[0070] 编译脚本可以通过命令行的方式调起IDE工具,并且以命令行的方式将目标路径传递给IDE工具。例如,当IDE工具为Keil编译程序Uv4.exe时,可以使用命令行“START Uv4.exe”来调起IDE工具,另外还可以使用命令行“/B Uv4.exe”来隐藏IDE工具的界面,也就是实现无编译界面的后台编译。

[0071] IDE工具获得目标路径后,可以根据目标路径查找到目标软件工程的工程主文件,然后调用该工程主文件并打开该工程主文件在IDE工具中对应的工程界面,实现为目标软件工程开启编译进程的目的。

[0072] 另外,编译进程将目标软件工程对应的源代码编译为可执行文件的过程属于现有技术,在此不做赘述。

[0073] 在一种实现方式中,在通过编译进程将目标软件工程对应的源代码编译为可执行文件的过程中,还可以根据目标软件工程对应的编译日志文件中的第一特征字段,确定目标软件工程的编译状态,其中,第一特征字段为:用于描述软件工程的编译状态的字段。

[0074] 其中,编译状态可以为编译进行中、编译完成等状态。

[0075] 编译日志文件可以为在执行步骤S101之前已存在的日志文件,也可以为编译进程临时为目标软件工程创建的文件。若是编译进程临时为目标软件工程创建编译日志文件,那么,编译脚本可以指定编译日志文件的名称与目标软件工程的名称相对应,例如,若软件工程的名称为:D0_100VB,则指定该目标软件工程对应的编译日志文件的名称为:pjt_D0_100VB.log。在一种实现方式中,编译脚本可以通过命令行“-o+编译日志文件名称”将指定编译日志文件的名称传递给IDE工具,则IDE工具在为目标软件工程开启编译进程后,编译进程可以根据所指定的名称来创建编译日志文件。

[0076] 在一种实现方式中,在确定目标软件工程的编译状态为编译完成时,还可以根据目标软件工程对应的编译日志文件中的第二特征字段,确定目标软件工程的编译结果,其中,第二特征字段为:用于描述软件工程的编译结果的字段。

[0077] 其中,编译结果可以为编译成功、编译失败等结果。

[0078] 具体的,在编译执行过程中,编译脚本还可以创建一个空的脚本日志文件,脚本日志文件用于记录编译脚本在执行时的事件,这样,在确定目标软件工程的编译结果后,还可以实时将编译结果写入脚本日志文件。

[0079] 另外,还可以基于目标软件工程的编译结果,输出针对目标软件工程的编译结果汇总信息。例如,汇总信息可以为“软件工程名称1编译成功”、“软件工程名称2编译失败”,将汇总信息输出到电子设备的屏幕中,可以使技术人员直观的了解编译结果。

[0080] 可见,本发明实施例提供的方案,利用编译脚本进行自动化编译,不需要人工参与,由编译脚本直接调起IDE工具并将待编译的目标软件工程的工程主文件的目标路径传递给IDE工具,IDE工具基于目标路径为目标软件工程开启编译进程,从而实现对目标软件工程的源代码的编译,提高了编译效率。

[0081] 在本发明实施例的另一种实现方式中,待编译的目标软件工程可以为多个,在这

种情况下,在图1所示实施例的基础上,上述步骤S102中将目标路径传递给IDE工具,以使IDE工具基于目标路径,为目标软件工程开启编译进程,以通过编译进程将目标软件工程对应的源代码编译为可执行文件的步骤,可以包括:

[0082] 将各个目标软件工程的目標路径传递给IDE工具,以使IDE工具基于各个目标软件工程的目標路径,分别为各个目标软件工程开启编译进程,以通过各个编译进程并行将各个目标软件工程对应的源代码编译为可执行文件。

[0083] 这种方式中,编译脚本将各个目标软件工程的目標路径传递给IDE工具,这样,IDE工具可以为每一目标软件工程开启独立的编译进程,从而实现多工程并行编译。

[0084] 在一种实现方式中,在通过编译进程并行将各个目标软件工程对应的源代码编译为可执行文件的过程中,可以针对每一目标软件工程,根据该目标软件工程对应的编译日志文件中的第一特征字段,确定该目标软件工程的编译状态,其中,第一特征字段为:用于描述软件工程的编译状态的字段。

[0085] 具体的,针对所有目标软件工程,可以以轮询的方式检查目标软件工程对应的编译日志文件是否存在,如果编译日志文件不存在,则继续检查下一目标软件工程,如果编译日志文件存在,则查询编译日志文件的内容,根据第一特征字段确定当前目标软件工程的编译状态,然后继续检查下一目标软件工程直到确定出所有目标软件工程的编译状态。

[0086] 在一种实现方式中,针对每一目标软件工程,在确定该目标软件工程的编译状态为编译完成时,还可以根据该目标软件工程对应的编译日志文件中的第二特征字段,确定该目标软件工程的编译结果,其中,第二特征字段为:用于描述软件工程的编译结果的字段。

[0087] 另外,还可以基于所有目标软件工程的编译结果,输出针对所有目标软件工程的编译结果汇总信息。

[0088] 具体的,针对每一目标软件工程,在确定该目标软件工程的编译结果后,可以实时将编译结果写入脚本日志文件,这样当所有目标软件工程均编译完成后,可以读取脚本日志文件,对全部目标软件工程的编译结果进行汇总,并把汇总情况输出到电子设备的屏幕中。

[0089] 可见,本发明实施例提供的方案,利用编译脚本进行自动化编译,不需要人工参与,即可实现对目标软件工程的源代码的编译,提高了编译效率,并且,采用多进程机制并行对多个软件工程进行编译,进一步提高了编译效率。

[0090] 下面以单片机工程为例对本发明实施例所提供的代码编译方法进行描述。具体的,编译脚本和IDE工具均运行在本地主机中,且运行平台为Windows操作系统,IDE工具为Keil uVision 4.0。

[0091] 为了实现自动化、多进程编译,编译脚本能够实现其正常功能的前提包括以下三点:1、IDE工具支持命令行调用方式;2、所有单片机工程均在同一根目录下(下面均以root_path表示);3、编译脚本运行前,需要放在root_path下。

[0092] 在以上三点前提下,本实施例进行编译的具体步骤如下:

[0093] 步骤a1、在本地主机中安装Keil uVision 4.0;

[0094] 步骤a2、将编译脚本放在根目录root_path下;

[0095] 步骤a3、运行编译脚本;

[0096] 步骤a4、编译脚本自动获取root_path的绝对路径,然后生成单片机工程的工程主文件(工程主文件为.uvproj格式的文件)的路径,同时指定编译日志文件的名称与单片机工程名称相对应;

[0097] 例如,单片机工程D0_100VB的工程主文件为:D0_100VB.uvproj,则该工程主文件的路径可以表示为:%root_path%D0_100VB/D0_100VB.uvproj,该单片机工程的编译日志文件名称可以指定为:pjt_D0_100VB.log;

[0098] 步骤a5、使用命令行START/B Uv4.exe调起IDE工具Uv4.exe(Uv4.exe为Keil的编译程序),其中/B选项用于实现无编译界面的后台编译;

[0099] 步骤a6、使用命令行“-b+工程主文件路径”指定待编译的单片机工程,以使IDE工具基于工程主文件路径为目标软件工程开启编译进程,接着使用命令行“-o+编译日志文件名称”将指定的编译日志文件名称传递给IDE工具,以使编译进程根据所指定的名称来创建编译日志文件;

[0100] 例如,一个单片机工程的命令行语句可以表示为:START/B Uv4.exe-b%local_path%D0_100VB\D0_100VB.uvproj-o../D_H1_100VD.log;

[0101] 步骤a7、顺序查询每个单片机工程的编译日志文件是否已经存在,若该单片机工程的编译日志文件尚不存在,重复本步骤来检查下一单片机工程,若该单片机工程的编译日志文件已经存在,执行步骤a8;

[0102] 步骤a8、根据特征字段判断该单片机工程的编译状态(编译进行中、编译完成);若该单片机工程的编译状态为编译完成,执行步骤a9,若该单片机工程的编译状态为编译进行中,返回执行步骤a7;

[0103] 步骤a9、继续根据特征字段判断该单片机工程的编译结果(编译成功、编译失败);第一次有单片机工程编译完成时,编译脚本创建一个空的脚本日志文件,把该单片机工程的编译结果以追加方式写入脚本日志文件,例如脚本日志文件名称为build_all_result.log;判断是否所有单片机工程均编译完成,如果是执行步骤a10,否则返回执行步骤a7;

[0104] 步骤a10、读取脚本日志文件,对所有单片机工程的编译结果进行汇总,然后以一定形式把全部单片机工程的编译结果输出到屏幕;

[0105] 步骤a11、删除上述编译过程中产生的编译日志文件(pjt_xxx.log)和脚本日志文件(build_all_result.log);

[0106] 步骤a12、输出编译完成提示信息到屏幕,等待用户退出脚本。

[0107] 可见,本实施例所提供的代码编译方法,仅需要运行编译脚本就可以实现所有单片机工程的自动化编译,相比于现有技术中需要由程序开发人员利用IDE工具进行编译的方式,节省了人力。此外,本实施例引入多进程编译方式,通过多个单片机工程的并行编译,缩短了所有单片机工程编译完成的时间,提高了程序开发人员调试源代码的效率。另外,当单片机工程的源代码与单片机工程保存在同一本地主机时,这种情况可以称之为本地编译;当单片机工程的源代码保存在远端服务器、单片机工程保存在本地主机时,这种情况可以称之为远程编译。相比于本地编译,远程编译会受到网速、网络阻塞状态以及网关设备等因素的影响。本实施例所提供的代码编译方法可以提高本地编译和远程编译的效率,并且远程编译效率提升尤为明显,编译时间可以缩短65%~85%。

[0108] 与上述的代码编译方法相对应,本发明实施例提供了一种代码编译装置。与图1所示的方法实施例相对应,图2为本发明实施例提供的一种代码编译装置的结构示意图,该装置可以包括:

[0109] 获得模块201,用于编译脚本获得待编译的目标软件工程的工程主文件的目标路径;

[0110] 编译模块202,用于调起集成开发环境IDE工具,并将所述目标路径传递给所述IDE工具,以使所述IDE工具基于所述目标路径,为所述目标软件工程开启编译进程,以通过所述编译进程将所述目标软件工程对应的源代码编译为可执行文件。

[0111] 可见,本发明实施例提供的方案,利用编译脚本进行自动化编译,不需要人工参与,由编译脚本直接调起IDE工具并将待编译的目标软件工程的工程主文件的目标路径传递给IDE工具,IDE工具基于目标路径为目标软件工程开启编译进程,从而实现对目标软件工程的源代码的编译,提高了编译效率。

[0112] 作为本发明的一个具体实施例,所述待编译的目标软件工程为多个;

[0113] 所述编译模块202将所述目标路径传递给所述IDE工具,以使所述IDE工具基于所述目标路径,为所述目标软件工程开启编译进程,以通过所述编译进程将所述目标软件工程对应的源代码编译为可执行文件的步骤,具体可以为:

[0114] 将各个目标软件工程的目标路径传递给所述IDE工具,以使所述IDE工具基于各个目标软件工程的目标路径,分别为各个目标软件工程开启编译进程,以通过各个编译进程并行将各个目标软件工程对应的源代码编译为可执行文件。

[0115] 作为本发明的一个具体实施例,所述编译脚本存储在所述目标软件工程所在的根目录下;

[0116] 所述获得模块201,可以包括:

[0117] 获得子模块,用于编译脚本获得所述编译脚本所在的根目录;

[0118] 生成子模块,用于根据所述根目录,生成所述根目录下的所述目标软件工程的工程主文件的目标路径。

[0119] 作为本发明的一个具体实施例,所述装置还可以包括:

[0120] 第一确定模块,用于根据所述目标软件工程对应的编译日志文件中的第一特征字段,确定所述目标软件工程的编译状态,其中,所述第一特征字段为:用于描述软件工程的编译状态的字段。

[0121] 作为本发明的一个具体实施例,所述装置还可以包括:

[0122] 第二确定模块,用于在确定所述目标软件工程的编译状态为编译完成时,根据所述目标软件工程对应的编译日志文件中的第二特征字段,确定所述目标软件工程的编译结果,其中,所述第二特征字段为:用于描述软件工程的编译结果的字段。

[0123] 作为本发明的一个具体实施例,所述装置还可以包括:

[0124] 输出模块,用于基于所述目标软件工程的编译结果,输出针对所述目标软件工程的编译结果汇总信息。

[0125] 本发明实施例还提供了一种电子设备,如图3所示,包括处理器301、通信接口302、存储器303和通信总线304,其中,处理器301,通信接口302,存储器303通过通信总线304完成相互间的通信,

[0126] 存储器303,用于存放计算机程序;

[0127] 处理器301,用于执行存储器303上所存放的程序时,实现如下步骤:

[0128] 编译脚本获得待编译的目标软件工程的工程主文件的目标路径;

[0129] 调起集成开发环境IDE工具,并将所述目标路径传递给所述IDE工具,以使所述IDE工具基于所述目标路径,为所述目标软件工程开启编译进程,以通过所述编译进程将所述目标软件工程对应的源代码编译为可执行文件。

[0130] 关于该方法各个步骤的具体实现以及相关解释内容可以参见上述图1所示的方法实施例,在此不做赘述。

[0131] 另外,处理器301执行存储器303上所存放的程序而实现的代码编译方法的其他实现方式,与前述方法实施例部分所提及的实现方式相同,这里也不再赘述。

[0132] 本发明实施例提供的电子设备,利用编译脚本进行自动化编译,不需要人工参与,由编译脚本直接调起IDE工具并将待编译的目标软件工程的工程主文件的目标路径传递给IDE工具,IDE工具基于目标路径为目标软件工程开启编译进程,从而实现对目标软件工程的源代码的编译,提高了编译效率。

[0133] 上述电子设备提到的通信总线可以是外设部件互连标准 (Peripheral Component Interconnect, PCI) 总线或扩展工业标准结构 (Extended Industry Standard Architecture, EISA) 总线等。该通信总线可以分为地址总线、数据总线、控制总线等。为便于表示,图中仅用一条粗线表示,但并不表示仅有一根总线或一种类型的总线。

[0134] 通信接口用于上述电子设备与其他设备之间的通信。

[0135] 存储器可以包括随机存取存储器 (Random Access Memory, RAM),也可以包括非易失性存储器 (Non-Volatile Memory, NVM),例如至少一个磁盘存储器。可选的,存储器还可以是至少一个位于远离前述处理器的存储装置。

[0136] 上述的处理器可以是通用处理器,包括中央处理器 (Central Processing Unit, CPU)、网络处理器 (Network Processor, NP) 等;还可以是数字信号处理器 (Digital Signal Processing, DSP)、专用集成电路 (Application Specific Integrated Circuit, ASIC)、现场可编程门阵列 (Field-Programmable Gate Array, FPGA) 或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件。

[0137] 在本发明提供的又一实施例中,还提供了一种计算机可读存储介质,该计算机可读存储介质中存储有指令,当其在计算机上运行时,使得计算机执行上述实施例中任一所述的代码编译方法。

[0138] 通过运行本发明实施例提供的计算机程序产品,利用编译脚本进行自动化编译,不需要人工参与,由编译脚本直接调起IDE工具并将待编译的目标软件工程的工程主文件的目标路径传递给IDE工具,IDE工具基于目标路径为目标软件工程开启编译进程,从而实现对目标软件工程的源代码的编译,提高了编译效率。

[0139] 需要说明的是,在本文中,诸如第一和第二等之类的关系术语仅仅用来将一个实体或者操作与另一个实体或操作区分开来,而不一定要求或者暗示这些实体或操作之间存在任何这种实际的关系或者顺序。而且,术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含,从而使得包括一系列要素的过程、方法、物品或者设备不仅包括那些要素,而且还包括没有明确列出的其他要素,或者是还包括为这种过程、方法、物品或者设备

所固有的要素。在没有更多限制的情况下,由语句“包括一个……”限定的要素,并不排除在包括所述要素的过程、方法、物品或者设备中还存在另外的相同要素。

[0140] 本说明书中的各个实施例均采用相关的方式描述,各个实施例之间相同相似的部分互相参见即可,每个实施例重点说明的都是与其他实施例的不同之处。尤其,对于装置、电子设备以及计算机可读存储介质实施例而言,由于其基本相似于方法实施例,所以描述的比较简单,相关之处参见方法实施例的部分说明即可。

[0141] 以上所述仅为本发明的较佳实施例而已,并非用于限定本发明的保护范围。凡在本发明的精神和原则之内所作的任何修改、等同替换、改进等,均包含在本发明的保护范围内。

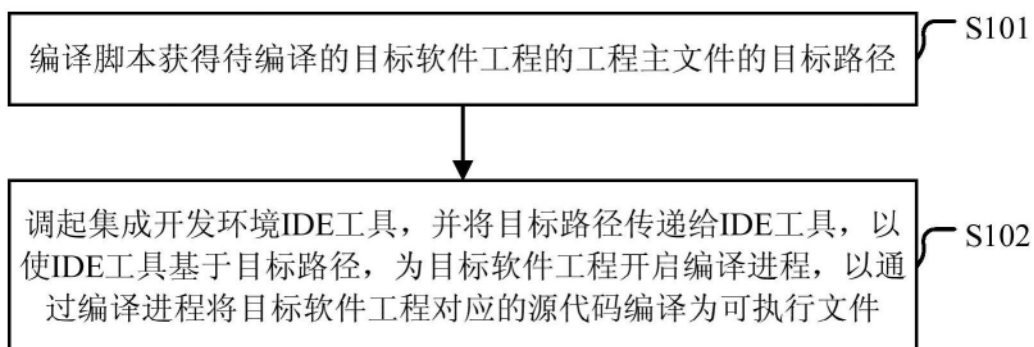


图1



图2

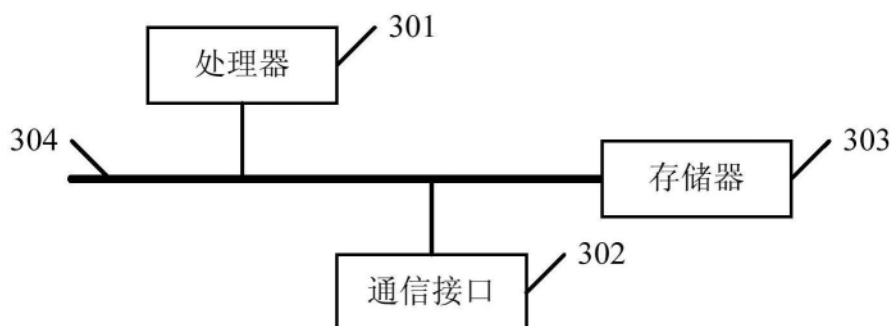


图3