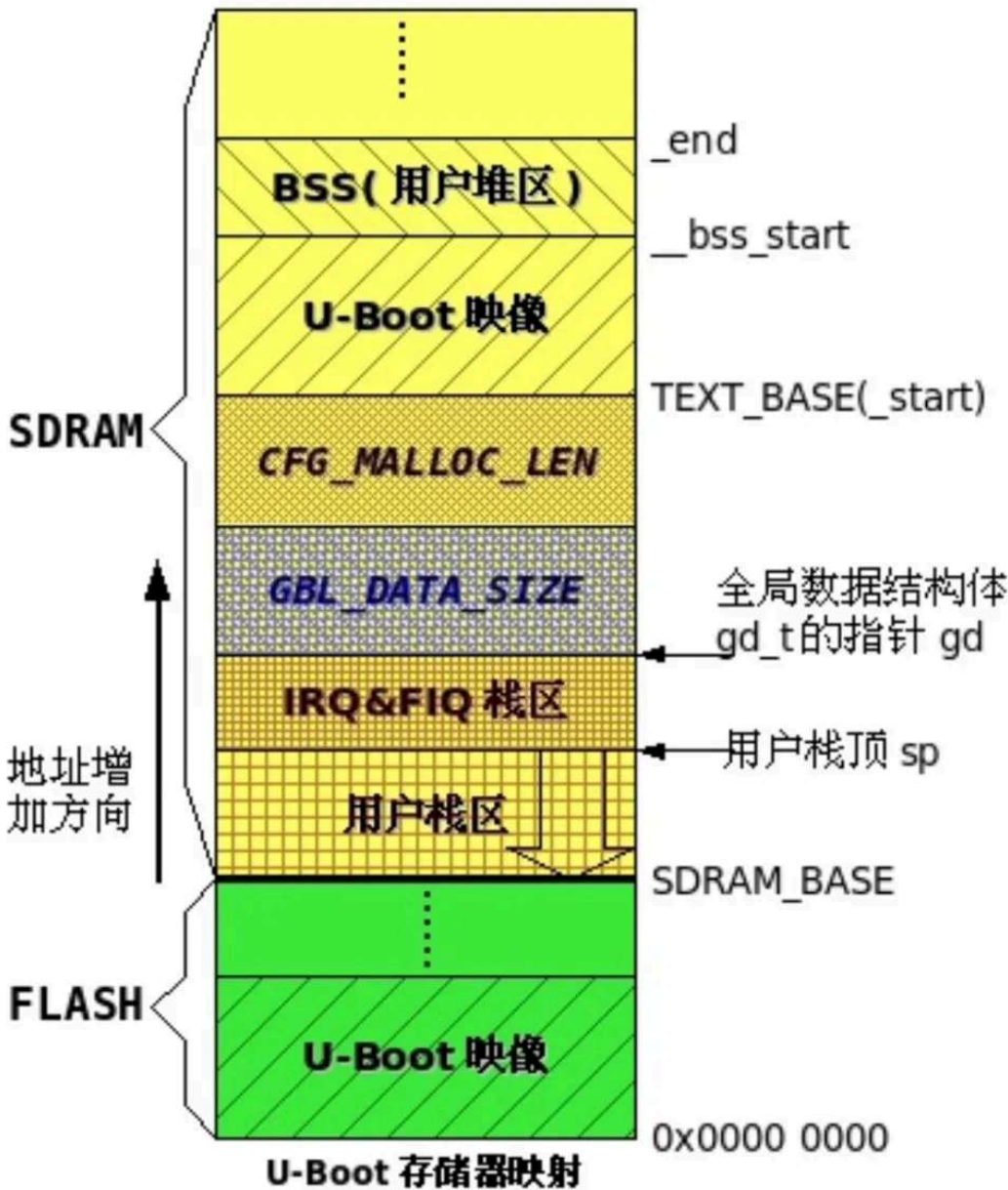


深度剖析U-Boot：嵌入式系统中的开源引导加载程序

原创 嵌入式工程师成长 嵌入式工程师成长日记 2025年03月09日 00:05 江苏

一、U-Boot的功能与作用



U-Boot内存使用情况

1. 硬件初始化：U-Boot在系统启动时承担着初始化硬件设备的重任。这包括初始化CPU、内存控制器、时钟、串口等基本硬件组件。

例如，在基于ARM架构的开发板上，U-Boot会设置CPU的工作频率，配置内存的时序和大小，使硬件处于可正常工作的状态，为后续操作系统的加载和运行做好准备。

2. 加载操作系统内核：这是U-Boot的核心功能之一。它可以从各种存储设备，如NAND Flash、SD卡、硬盘等，读取操作系统内核镜像，并将其加载到内存中的指定位置。

在加载过程中，U-Boot会根据不同的存储设备和文件系统格式，采用相应的读取策略。比如，对于NAND Flash，由于其存储特性，U-Boot需要处理坏块管理等问题，以确保内核镜像的正确读取。加载完成后，U-Boot会将系统控制权交给操作系统内核，启动操作系统。

3. 环境变量管理：U-Boot支持环境变量的设置和管理。环境变量包含了系统启动和运行所需的各种参数，如内核启动参数、网络配置信息等。开发者可以通过U-Boot的命令界面或配置文件，灵活地设置和修改这些环境变量。

例如，在需要通过网络启动系统时，开发者可以设置网络相关的环境变量，包括IP地址、服务器地址等，使U-Boot能够从网络服务器获取内核镜像和文件系统。

二、U-Boot的工作流程

1. 第一阶段启动：这一阶段通常在硬件复位后立即执行，主要负责完成最基本的硬件初

始化工作，如设置CPU的工作模式、初始化内存的最小可用部分等。

第一阶段的代码通常存放在开发板的片内ROM或Flash的特定区域，并且是用汇编语言编写的，以确保高效执行。由于第一阶段的资源有限，它的主要任务是为第二阶段的启动做好准备，如初始化必要的硬件设备，使系统能够进入第二阶段的启动流程。

2. 第二阶段启动：在第一阶段完成初步的硬件初始化后，U-Boot进入第二阶段。这一阶段主要是用C语言编写的，功能更加丰富和复杂。

第二阶段会进一步初始化硬件设备，如完整地初始化内存、初始化各种外部设备控制器等。然后，U-Boot会加载环境变量，根据环境变量的设置来决定如何加载操作系统内核。它可以从本地存储设备读取内核镜像，也可以通过网络从服务器获取。

最后，U-Boot将内核镜像加载到内存中的合适位置，并传递启动参数，将系统控制权交给内核，完成整个启动过程。

三、U-Boot的移植

移植流程：通常包括以下几个关键步骤：首先，要对目标硬件平台进行深入分析，了解其硬件架构、寄存器配置等信息。然后，修改U-Boot的源代码，使其能够适配目标硬件。这包括修改与硬件相关的初始化代码、设备驱动代码等。

例如，在将U-Boot移植到一款新的ARM开发板时，需要根据开发板的具体硬件连接，修改时钟初始化代码、GPIO驱动代码等。接着，进行交叉编译，使用针对目标硬件平台的交叉编译工具链，将修改后的U-Boot源代码编译成可在目标硬件上运行的二进制文件。

最后，通过调试工具，如JTAG，将编译好的U-Boot下载到目标硬件上进行调试和测试，逐步解决移植过程中出现的问题。

四、U-Boot在嵌入式系统中的应用案例

智能家居设备：在智能家居领域，许多设备都采用了嵌入式系统，U-Boot在其中发挥着重要作用。例如，智能摄像头的启动过程中，U-Boot负责初始化摄像头的图像传感器、网络模块、存储设备等硬件组件，然后加载操作系统内核和相关的应用程序。

通过U-Boot的环境变量设置，可以方便地配置摄像头的网络参数，实现远程监控功能。同时，U-Boot的可定制性使得开发者能够根据不同的摄像头硬件平台和功能需求，对其进行个性化定制，提高产品的竞争力。

STM32 106

STM32 · 目录

上一篇

每日分享02：嵌入式软件面试题，这些八股文你掌握了吗？

下一篇

2025小米嵌入式软件开发岗位校招面试题（面经），快来试试看吧！

个人观点，仅供参考