

嵌入式

Wednesday, April 25, 2018 8:51 PM

本人观点: !!! 未来是属于arm的, 信不信由你 !!!

M内核 (stm32(M) 等等) 是单片机, 一般是标准C开发, 专用下载器下载, (一般)裸机无系统, 直接操控硬件, 适合简单算法、高速实时。

A内核 (树莓派(A7、A53)、ZYNQ|PYNQ(A9) 等等) 是处理器, 跑操作系统, 高级语言开发, 不需要专用下载器, 不直接去控制硬件 (可控但效率低), 适合多任务复杂算法, 实时性较弱。

嵌入式的话, 本人建议偏软件, 没有为什么~~~当然, 应该是先M后A, 慢慢来……

嵌入式软件最重要的是嵌入式操作系统, 就可以具体到linux, 学的话, 可以拿stm32、树莓派等平台, 边实践边学……

1、操作系统与裸机的区别

裸机运行的程序代码, 一般由一个main函数中的while死循环和各种中断服务程序组成, 平时CPU执行while循环中的代码, 出现其他事件时, 跳转到中断服务程序进行处理, 没有多任务、线程的概念。

引入操作系统后, 程序执行时可以把一个应用程序分割为多个任务, 每个任务完成一部分工作, 并且每个任务都可以写成死循环。操作系统根据任务的优先级, 通过调度器使CPU分时执行各个任务, 保证每个任务都能够得到运行。若调度方法优良, 则可使个任务看起来是并行执行的, 减少了CPU的空闲时间, 提高了CPU的利用率。由操作系统的任务管理衍生出相应的CPU管理、内存管理, 它们分别负责分配任务对CPU的占有权和管理任务所占有的内存空间。在linux操作系统中, 还具有文件管理、I/O设备管理的功能。

2、stm32能运行什么操作系统, 能运行linux系统吗?

首先操作系统有两种: 一种是用MMU (Memory Managment Unit, 即存储管理单元) 的, 另一种是不用MMU的。

用MMU的是Windows、MacOS、Linux、Android;

不用MMU的是FreeRTOS、VxWorks、ucLinux、uc/OS。

其次CPU也有两种: 带MMU的和不带MMU的。

带MMU的有Cortex-A、ARM9、ARM11系列, 不带MMU的有Cortex-M系列 (和前述自行联系)

STM32是Cortex-M3系列, 所以目前还不能运行Linux (不考虑ucLinux), 但是stm32可以运行uc/OS、ucLinux、FreeRTOS、VxWorks等操作系统。其中uc/OS操作系统是开源代码, 且小巧而严谨, 具有很高的实时性, 可以作为学习linux等高级操作系统的基础, 实际项目中也经常使用uc/OS开发。

3、建议直接A系列, Linux……