一个开放源码的嵌入式仿真环境 — SkyEye

SkyEye 项目开发小组

http://hpclab.cs.tsinghua.edu.cn/~skyeye/

I hear and I forget,
I see and I remember,
I do and I understand.

摘要：

本文是对开放源码的嵌入式仿真环境 SkyEye 的综述。首先介绍了 SkyEye 的背景、目标和存在的意义；接下来对目前 SkyEye 本身进行了简要的技术分析；然后对 SkyEye 上已经移植成功的操作系统、典型应用进行了介绍；最后讲述了 SkyEye 正在开展的工作和将来的发展计划。

一、 SkyEye 的目的与意义

1. SkyEye 产生的背景

操作系统是软件产业的基础和龙头，它能左右软件产业发展的方向，是世界软件产业最大的利润来源。对于后 PC 时代和 pervasive computing（普适计算）而言，嵌入式系统将无处不在，其中关键的软件核心技术包括嵌入式操作系统和网络互连中间件等。

所谓嵌入式系统一般指非 PC 系统，它包括硬件和软件两部分。硬件包括处理器／微处理器、存储器及外设器件和 I／O 端口、图形控制器等。软件部分包括操作系统软件（要求实时和多任务操作）和应用程序编程。嵌入式操作系统可以广泛应用于 PDA、掌上电脑、手机、信息家电（网络冰箱、机顶盒）等 嵌入式设备。

对于一般的嵌入式系统的软件开发和学习，一般不能脱离一个具体的硬件环境。对于想研究嵌入式 linux 等操作系统和一些底层系统软件（如 TCP/IP 等）的研究和开发人员，可能存在如下几方面的问题：(1) 经常苦于经费不足，缺少足够的硬件开发板和完善的软件开发环境，相关的书籍对一些最新软件的分析还不够全面，无法深入研究和开发嵌入式软件。(2) 高层次的软件设计和开发一般不用太考虑底层硬件的实现细节，如果直接处于一个具体的硬件环境下，在开发和研究中可能会陷入硬件的具体细节中不能自拔，而不能把精力放到高层次的软件设计和开发上。(3) 如果硬件开发环境不太稳定（这种情况经常见到），且对具体的硬件不是很了解，则可能在排除问题上花费大量的不必要的时间。

对于想了解、学习一般操作系统的实现原理，linux/uclinux 操作系统或 TCP/IP 等系统级软件的实现的人员，目前一般采用的方法是看书和读源代码，这是一种静态的学习方法，但效率较低，比较枯燥，缺少一种动态和亲自实践的感觉。要想深入分析和开发软件，就要动手编程，不能只是看看书，读读代码，只有通过亲手实践才能够掌握软件设计的核心内容。

如何进行动态分析和研究？以相对简单的 uC/OS-II操作系统为例 :其作者已经写了一本介绍 uC/OS-II操作系统的书籍，且也有中文版发行。但看完书后，你能够真正掌握 uC/OS-II 的实现和操作系统的核心技术吗？我想，可能大部分操作系统的初学者只是知道（**不等于了解、掌握**）书本上讲的内容，而且书中的例子基于 Intel x86, 但 x86 结构的复杂性进一步限制了大家对一些与硬件相关的 uC/OS-II 核心部分（如任务切换、中断处理等）的掌握。如

果能够有一个试验环境，大家能够在这个试验环境中一步一步地看到 uC/OS-II 是如何执行的，每一行 C 代码或汇编代码执行的结果是什么，相信对大家全面了解 uC/OS-II 有很好的帮助。如果大家能够亲自动手，写出一些基于 uC/OS-II 的应用或者直接修改 uC/OS-II 代码并成功运行，我想如果到了这一步，大家可以算是真正地掌握 uC/OS-II。

上面所指出的问题和需求促使了 SkyEye 项目的诞生。

2. SkyEye 的目标和意义

SkyEye 是一个开源软件（opensource software）项目，中文名字是“天目”。SkyEye 的目标是在通用的 Linux 和 Windows 平台实现一个仿真集成开发环境，模拟常见的嵌入式计算机系统（目前支持基于 arm7tdmi 的 AT91 开发 板，带网络仿真功能）；可在 SkyEye 上运行 uclinux 以及 uC/OS-II 等多种嵌入式操作系统和各种系统软件（如 TCP/IP，图形子系统，文件子系统等），并可对它们进行源码级的分析和测试。SkyEye 的推出具有下面两方面的意义：

- 通过 SkyEye 仿真集成环境可以很方便地进入到嵌入式系统软件学习和开发的广阔天地中。尤其对于缺少嵌入式硬件开发环境和软件开发环境的用户来说，它将是一个非常有效的学习工具和开发手段，因为 SkyEye 的整个软件系统都是 Open Source 的，且基于 GPL 协议（uCOS-II 除外）。因此，如果您想学习 Liunx 操作系统或者进行嵌入式系统开发，但苦于没有硬件支持和 money 的话，欢迎使用我们的 SkyEye 仿真环境软件！
- 如何你想研究与具体硬件无关的系统软件（如 TCP/IP 协议栈等），采用 SkyEye 可以有效地提高工作效率，因为你可以直接在 uCOS-II 和 uClinux for SkyEye 上进行开发和调试，而与具体硬件打交道的各种 driver 已经存在，且有源码级调试环境，你只需关心高层的逻辑设计和实现就可以了。
- SkyEye 本身作为一个开放式的项目体系，可以划分为多个独立的子项目系统。通过参与 SkyEye 的 各个子项目，与大家共同交流、协作，可以帮助您进一步学习、分析、精通 Linux 内核，掌握 ARM 嵌入式 CPU 编 程。因此，如果您想成为一个 Liunx 高手、嵌入式系统专家的话，欢迎加入到我们开发者的行列中来！

在 32 位嵌入式 CPU 领域中，ARM 系列 CPU 所占比重相当大，而 arm7tdmi 是其中最广泛的一个 CPU，因此 SkyEye 首先选择了 arm7tdmi 作为仿真的目标 CPU。目前在 SkyEye 上可运行并进行源码级调试 uclinux、uC/OS-II 操作系统和 LwIP(一个著名的嵌入式 TCP/IP 实现)等系统软件。SkyEye 可用于学习，分析，开发 uclinux、uC/OS-II 操作系统内核和 TCP/IP 实现，了解 ARM 嵌入式 CPU 编程。而这一切都可在一个集成环境中完成。如果能够改进 SkyEye 本身，则大家对 ARM，8019 ethernet 网络芯片等硬件的了解也会更深入。SkyEye 项目是一个自由软件（Open Source Software）项目，遵循 GPL 协议。SkyEye 可运行在 Linux 和 Windows 平台。它并不能取代开发板等硬件的功能，但通过它可以比较容易进入到嵌入时软件的广阔天地中。由于 SkyEye 建立在 GDB 基础之上，使用者可以方便地使用 GDB 提供的各种调试手段对 SkyEye 仿真系统上的软件进行源码级的调试，还可以进行各种分析，如执行热点分析、程序执行覆盖度分析等。由于 SkyEye 提供了源代码和相关文档，有经验的用户完全可以修改和扩充 SkyEye 来满足自己的需求。

3. 可以在 SkyEye 上做什么

在 SkyEye 上可以做什么事情呢？许多人下载了 SkyEye 和相关在 SkyEye 上运行的操作系统和应用程序，在自己的计算机上安装并且运行成功。但他们不知道下一步该干什么。其实在 SkyEye 环境中可以干许多事情：

1. 对于嵌入式操作系统的初学者和入门的学生而言，他们可以先看一些有关操作系统和嵌入式操作系统方面的教材和书籍，如与 uC/OS、Minix、uClinux、Linux

相关的书籍等。然后可以在 SkyEye 上开发一些简单的应用程序例子（如进程间通信、进程优先级、死锁情况，网络应用等），对某些操作系统功能（如进程调度、内存管理、网络子系统、文件子系统等）进行简单的修改和扩展，并通过 SkyEye 进行运行和调试，看看会发生什么情况。

2. 对于有一定经验的软件工程师而言，在 SkyEye 上完成一定的应用系统原型开发是值得一做的事情。比如移植或开发一个文件子系统或网络子系统到一个特定的操作系统中，相信比在一个真实的开发板上开发要容易一些。在 SkyEye 上进行一些操作系统级的移植和开发（如移植 RTLinux、RTAI 等其它操作系统到 SkyEye 上）也是很有挑战性的工作。
3. 对于硬件工程师而言，对 SkyEye 进行扩充，设计新的硬件仿真（如 USB、IDE 硬盘等），使得 SkyEye 的硬件仿真功能更加强大，支持更多功能的软件，是很有意义的事情。

二、 SkyEye 的技术分析

1. SkyEye 总体结构

SkyEye 基于 GDB /armulator，模仿了一个完整的嵌入式计算机系统－Atmel91X40，目前包括 CPU、内存、I/O 寄存器、时钟、UART、网络芯片，将来还会有 MMU、Cache, LCD、USB 等各种硬件。在 SkyEye 上运行的操作系统和各种系统软件“ 意识 ”不到它们是在一个虚拟的计算机系统上运行。

SkyEye 从总体上分为四个层次：

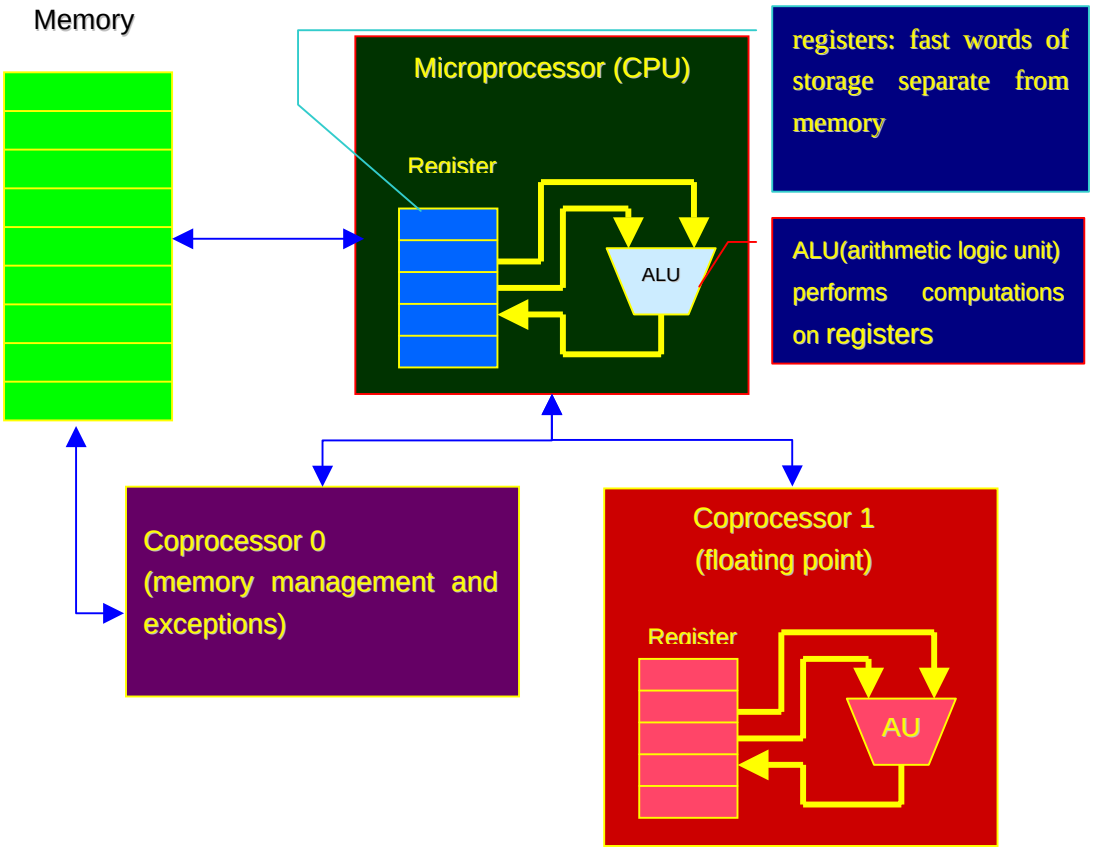
- 用户接口模块：包括命令行用户界面和图形用户界面，即是处理用户的输入命令，并把相关调试数据输出给用户。
- 符号处理模块：主要处理执行文件的头信息，解释执行文件中内嵌的 debuger 调试信息，对符号表的管理，对源代码表达式的解析，定位源代码中的语句位置和机器码的位置关系等。
- 目标控制模块：主要完成执行控制（如中断程序的执行，设置中断条件等），程序栈结构分析，对具体目标硬件的控制（如本地调试、远程调试和仿真调试的控制）。
- 目标仿真模块：主要是模仿计算机系统中的主要硬件（包括 CPU、内存和各种硬件外设等）的执行，对执行文件的机器指令进行解释，并仿真执行每一条机器指令，产生响应的硬件响应。

2. SkyEye 模拟的硬件介绍

目前 SkyEye 仿真的 CPU 包括不带 MMU 的 AT91X40 和带 MMU 的 ARM720T，它们都是基于由 ARM 公司提供相关的 ARM7TDMI CPU 内核。SkyEye 还模拟了其它硬件外设，如串口、网络芯片、内存、时钟等。

ARM 公司自 1990 年正式成立以来，一直以 IP(Intelligence Property)提供者的身份向各大半导体制造商出售知识产权，而自己不介入芯片的生产销售，加上其设计的芯核具有功耗低、成本低等显著优点，因此获得众多的半导体厂家和整机厂商的大力支持，在 32 位嵌入式应用领域获得了很大的成功，目前已经占有 75%以上的 32 位 RISC 嵌入 式产品市场。在低功耗、低成本的嵌入式应用领域确立了市场领导地位。现在设计、生产 ARM 芯片的国际大公司已经超过 50 多家，国内中兴通讯和华为通讯等公司也已经购买 ARM 公司的技术用于通讯专用芯片的设计。目前非常流行的 ARM 芯核有 ARM7TDMI，StrongARM ARM720T，ARM9TDMI，ARM922T，ARM940T，ARM10 等。

ARM RISC architecture



Atmel91X40 是美国 Atmel 公司生产的 AT91 系列微控制器中的一员，具有 ARM7TDMI 核、大容量 Flash 存储 器以及片内 SRAM 和外围 DRAM。这种微控制器的特点是高性能的 32 位 RISC 体系结构、高密度的指令集、低功耗以及实时性，扩充的 Flash 存储器还增加了开发者使用的灵活性。除此以外，大量的内部分组寄存器加速了对异常的处理过程，从而使其更适合于实时控制的应用。8 级基于向量的优先级中断控制器和外围数据控制器 PDC 大大增强了实时器件的性能。此器件适用于开发工业自动化系统、MP3、销售终端、GPS 接收机以及无线网络产品等对功耗敏感且要求具有实时性的产品。

3. MMU/Cache 仿真

除了支持仿真不带 MMU 的 ARM7TDMICPU 内核和基于 ARM7TDMI 的 Atmel91X40 CPU 外，目前 SkyEye 也支持仿真带 MMU 的 ARM720TCPU。MMU（Memory Management Unit）即存储器管理单元，是用来管理虚拟内存系统的器件。MMU 通常是 CPU 的一部分，本身有少量存储空间存放从虚拟地址到物理地址的匹配表。此表称作 TLB(translation lookaside buffers)。（TLB 表中保存的是虚址及其对应的物理地址，权限，域和映射类型）。如果没有查到，则进行查找 translation table，称为 translation table walk（ttw）。经过 ttw 后将查到的信息保存到 TLB。然后根据 TLB 表项的物理地址进行读写。所有数据请求都送往 MMU，由 MMU 决定数据是在 RAM 内还是在大容量存储器设备内。如果数据不在存储空间内，MMU 将产生页面错误中断。

MMU 的两个主要功能是：

① 将虚地址转换成物理地址。

② 控制存储器存取允许。MMU 关掉时，虚地址直接输出到物理地址总线。

MMU/cache 的仿真主要是仿真控制 MMU/cache 的寄存器结构、TLB 结构、cache 结构、

translation table walk 的控制逻辑以及在此基础上内存的读取操作。MMU/cache 的仿真主要是依据 ARM720T 处理器的体系结构进行的，而且 MMU/cache 的仿真与 ARM Linux 的移植是分不开的，所以仿真的过程中需要参照 ARM Linux 和 ARM720T 对 skyeye ARM 仿真的部分进行相应的修改。

三、 Skyeye 上已经移植成功的 OS

1. uCLinux

Linux 是一种很受欢迎的操作系统，它与 UNIX 系统兼容，开放源代码。它原本被设计为桌面系统，现在广泛应用于服务器领域。而更大的影响在于它正逐渐的应用于嵌入式设备。uClinux 正是在这种氛围下产生的。在 uClinux 这个英文单词中 u 表示 Micro，即小的意思，C 表示 Control，即控制的意思，所以 uClinux 就是 Micro-Control-Linux，字面上的理解就是"针对微控制领域而设计的 Linux 系统"。uClinux 与标准的 Linux 的一个关键区别是 uClinux 只支持不带 MMU 的 CPU，而标准的 Linux 只支持带 MMU 的 CPU（主要是 32 位嵌入式 MMUless CPU）。为了减少 OS 的尺寸，uClinux 采用了如下做法：1. 删除不需要的内核功能，重新配置内核；2．uClinux 的根（root）文件系统采用 romfs 文件系统；3．uClinux 的应用程序库为精简的 uClibc，而不是庞大的 glibc

UClinux 主要针对没有 MMU 的处理器，即 uClinux 不能使用处理器的虚拟内存管理技术。uClinux 系统对于内存的访问是直接的，（它对地址的访问不需要经过 MMU，而是直接送到地址线上输出），所有程序中访问的地址都是实际的物理地址。操作系统对内存空间没有保护（这实际上是很多嵌入式系统的特点），各个进程实际上共享一个运行空间（没有独立的地址转换表）。

uClinux 对内存的管理减少同时就给开发人员提出了更高的要求。如果从易用性这一点来说，uClinux 的内存管理是一种倒退，退回了到了 UNIX 早期或是 Dos 系统时代，开发人员不得不参与系统的内存管理。从内存的访问角度来看，开发人员的权利增大了（开发人员在编程时可以访问任意的地址空间），但与此同时系统的安全性也大为下降。虽然 uClinux 的内存管理与标准 Linux 系统相比功能相差很多，但这是目前嵌入式设备的现状，在嵌入式设备中，由于成本等敏感因素的影响，普偏的采用不带有 MMU 的处理器，这决定了系统没有足够的硬件支持实现虚拟存储管理技术。

uClinux 没有 MMU 管理存储器，在实现多个进程时（fork 调用生成子进程）需要实现数据保护。uClinux 的多进程管理通过 vfork 来实现。这意味着 uClinux 系统 fork 调用完程后，要么子进程代替父进程执行（此时父进程已经 sleep）直到子进程调用 exit 退出，要么调用 exec 执行一个新的进程。uClinux 的这种多进程实现机制同它的内存管理紧密相关。uClinux 针对 noMMU 处理器开发，所以被迫使用一种 flat 方式的内存管理模式，启动新的应用程序时系统必须为应用程序分配存储空间，并立即把应用程序加载到内存。缺少了 MMU 的内存重映射机制，uClinux 必须在可执行文件加载阶段对可执行文件进行处理，使得程序执行时能够直接使用物理内存。

目前全面分析 uClinux 的书籍还较少，不过除了上述内存管理和进程管理上与标准的 Linux 有较大差别外，在其它方面二者基本一致。所以参考目前已有的 Linux 方面的分析文章对理解 uClinux 也是很有帮助的。如果要学习 Linux 设备驱动程序的开发，推荐看《Linux device driver 2》中英文版；如果要深入学习 Linux2.4.x 的实现，推荐看《Linux 内核源代码情景分析》；如果要深入了解 UNIX 的内部实现机制和相关理论，推荐看《Unix Internal》中英文版；浙江大学也出了不少有关 Linux 的书籍，建议参考阅读。

2. uC/OS-II

SkyEye 作为一个基于 Atmel 91x40 开发板的仿真环境，在它上面可以移植各种适合于嵌入式开发应用的操作系统。将 uC/OS-II 移植到 SkyEye 上是我们对此做的又一次尝试。uC/OS-II 是一个简单、高效的嵌入式实时操作系统内核。自从 1992 年以来，已经被应用到各种嵌入式系统中。目前它可以支持 x86、ARM、PowerPC、MIPS 等众多体系结构，并有上百个商业应用实例，其稳定性和可用性是经过实践验证的。同时，它的源代码公开，任何人都可以从 www.ucos-ii.com 的网站上获得全部源码以及其在各种体系结构平台上的移植范例。无论是通过学习 uC/OS-II 来了解实时操作系统的构造，或者是直接使用它来针对具体应用进行开发，都是非常方便和可行的。

最新的 uC/OS-II 的内核已经到了 2.61 版。实际上，2.0 版以上的内核都已经具有可抢占的实时多任务调度功能，另外它还提供了许多系统服务，例如信号量、消息队列、邮箱、内存管理、时间函数等，这些功能可以根据不同的需求进行裁减。可以说，uC/OS-II 是一个具备现代操作系统特点的 RTOS，同时由于它的结构清晰，注解详尽，使其具有了良好的可扩展性和可移植性。因此，它被广泛地应用于各种架构的微处理器上。

有关于 uC/OS-II 实时操作系统的分析和介绍，可以参考清华大学勐贝贝老师翻译的那本《uC/OS-II — 源码公开的实时嵌入式操作系统》。这本书非常详尽的介绍了关于 uC/OS-II 的基本概念、内核结构、任务管理、内存管理等重要内容，同时还分析了在 80x86 上的移植范例。

3. ARM Linux

ARM Linux 属于 Linux 标准发行内核中的一个分支，支持大量带 MMU 的 ARM 系列 CPU，如 ARM720T、Intel StrongARM 等。我们移植 ARM Linux 的目的，一方面是要在 Skyeye 上仿真带 MMU 支持的 ARM CPU，目前 CPU 型号初步定在 CPU core 为 ARM7TDMI，并带 MMU 支持的 ARM720T；另一方面，要让所仿真的环境，可以运行带 MMU 支持的操作系统，这里我们选择了 ARM Linux 作为移植对象，并且选择了体系结构为 clpx711x 的 ARM Linux 内核，它支持 ARM720T，UART 串口驱动等，符合 Skyeye 目前支持的硬件。ARM Linux 的移植，目前主要集中在启动，MMU 支持，中断处理（主要是时钟中断），UART，异常处理等方面。有关 ARM Linux 的更多信息可访问 http://www.arm.linux.org.uk/ .

4. 更多的 OS Porting

目前移植在 SkyEye 上的 uC/OS-II 版本为 2.51 版。移植 uC/OS-II 的工作更多的是为了验证 SkyEye 的内部仿真逻辑的正确性。而在 SkyEye 上的 uClinux 版本是 20020927, 在 SkyEye 上调试 uClinux 可有效地了解 uClinux 的实现细节，得到书本上无法得到的知识和经验。

实际上基于 SkyEye 的 OS Porting 的工作可以延续下去，可能你已经自己写过一个简单的基于优先级的任务调度器或者一个相对完整的操作系统内核，那么你都可以基于 SkyEye 这个仿真环境在还没有硬件的条件下做前期的验证工作，这样能大大地提高自己开发 OS 的效率和正确性。另一方面，目前已经存在不少优秀的操作系统，如 MINIX 等，把它们移植到 SkyEye 上，会大大提高移植人员的实际动手能力和对操作系统的理解能力。

欢迎对嵌入式操作系统感兴趣的人在 SkyEye 上研究、开发并调试各种操作系统。

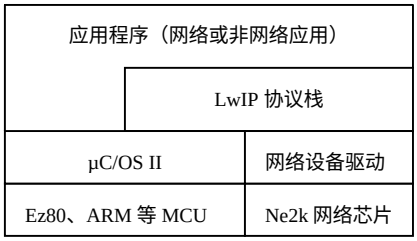
四、 SkyEye 目前支持的网络协议栈

1. tcp/ip on uCLinux

uCLinux 上面有完整的 tcp/ip 协议栈，但缺少对 SkyEye 的网络仿真芯片（仿真兼容 Ne2k 的 8019as）的驱动程序。驱动程序主要包括了初始化，中断处理，接收数据包的处理，发送数据包的处理等工作。驱动程序的具体分析可参考源代码和相关分析文档。

2. lwip on uC/OS-II

μC/OS-II 是一个富有开放色彩的 RTOS，但是它目前的一些第三方 TCP/IP 支持都是完全商业化的，很少给出源代码，这影响了μC/OS II 的研究和推广。通过把开放源代码的 TCP/IP 协议栈 LwIP 移植到μC/OS-II 上来，就获得了一套可免费研究、学习的嵌入式网络软件平台。系统示意图如下图：



LwIP 是瑞士计算机科学院（Swedish Institute of Computer Science）的 Adam Dunkels 等开发的一套用于嵌入式系统的开放源代码 TCP/IP 协议栈。LwIP 的含义是 Light Weight(轻型)IP 协议。LwIP 可以移植到操作系统上，也可以在无操作系统的情况下独立运行。我们目前使用的是 LwIP 的最新稳定版 V0.5.3。

LwIP 协议栈在设计时就考虑到了将来的移植问题，因此把所有与硬件、OS、编译器相关的部份独立出来，放在/src/arch 目录下。因此 LwIP 在μC/OS-II 上的实现就是修改这个目录下的文件，其它的文件一般不应该修改。主要修改和实现的是以下几部份：

与 CPU 或编译器相关的 include 文件

与 OS 相关的一些结构和函数

lib_arch 中库函数的实现

网络芯片驱动程序

我们在 Skyeye 中所仿真的网络芯片是 Ne2k，所以目前实现的网络设备驱动是针对 Ne2k 的，其它类型的网络芯片驱动可以在 LwIP 的网站上找到。

做完以上代码移植的工作后，LwIP 就可以顺利运行在 uC/OS-II 下了，在 Skyeye 中可以从 host 主机 ping 通 Skyeye 上运行的 uC/OS-II＋LwIP，这说明 ARP 协议和 ICMP 协议都运行正常。应用层目前实现了一个 tcp echo server，见下文。

3. 更多的 TCP/IP 协议栈

uC/OS-II 是一个简单的多任务内核，它缺少很多基本的功能如协议栈。但是它的简单也往往成为移植和实验各种协议栈的优点，你不需要考虑太多的问题。它提供的多任务调度、任务间通信、内存管理功能足以实现任何嵌入式 TCP/IP 协议栈，而又没有 linux 那样烦琐，所以在 uC/OS-II 中移植或试验新的协议栈是很方便的。

除了 LwIP 之外，还有很多开源的嵌入式 TCP/IP 协议栈，它们各有各的特点，有些功能较全面，有些占用资源出奇地少，能适合不同场合的需要，而且多接触一些不同的思路，对协议和编码能有更好的理解。其中的代表有 tinyTCP、ucIP、uIP 等，它们的基本思路应该都是一致的，Skyeye 小组希望能有更多对 TCP/IP 协议栈感兴趣的人参与进来，特别是 ucIP，它实际上是专门为 uC/OS-II 设计的，不过我们还没有时间去分析使用它。这些都要留给广大爱好者去做了。

五、 SkyEye 上已经实现的几个简单应用

1. Echo server on lwip

做完 LwIP 的移植修改工作以后，就可以在μC/OSII 中初始化 LwIP，并创建 TCP 或 UDP

任务进行测试了。值得注意的是 LwIP 的初始化必须在μC/OSII 完全启动之后也就是在任务中进行，因为它的初始化用到了信号量等 OS 相关的操作。

LwIP 提供了两种 API（Application Program's Interfaces，应用程序接口），供用户使用这个协议栈。

第一种是 BSD API，这种 API 非常象 BSD 标准 UNIX 中的 socket API，所以这样命名。它跟普通的 socket api 一样是基于 open-read-write-close 模型的。它提供了一种标准的方法来使用 LWIP 协议栈。但这种 API 效率较低，消耗资源较多，因此不推荐使用。

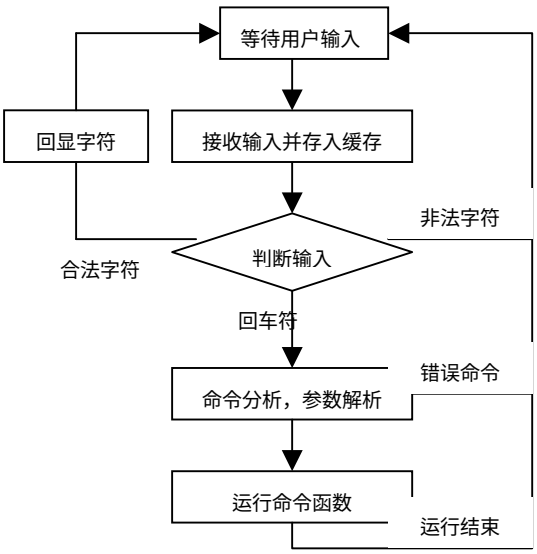
第二种称为 RAW API，这种 API 接口实际上是直接使用了 LwIP 协议栈中的回调函数（callback functions），从而应用程序和协议栈代码更好地集成在一起，运行在同一个线程即任务中。RAW API 相对于普通 BSD API 来说，速度更快，消耗内存更少，唯一的缺点是编程复杂，对程序员的要求高，考虑到我们是在嵌入式设备中工作，这样的缺点是可以接受的。事实上，BSD API 就是用 RAW API 来实现的。

利用 Berkley API 实现的 tcpecho_thread 是一个 TCP echo 服务器，监听 7 号端口，这实际上是实现了 RFC 中最简单的 echo 协议：收到什么，就往回发同样的内容。 编译运行后，用 ping ip 地址 命令可以得到 ICMP reply 响应。用 telnet ip 地址 7（登录 7 号端口） 命令可以看到 echo server 的回显效果。说明 ARP、ICMP、IP、TCP 协议都已正确运行。

2. Genie-shell for uC/OS-II

uC/OS-II 只提供了操作系统内核，用户要自己添加文件处理、人机界面、网络接口等重要部分。其中 Shell(人机界面)提供了人与机器交互的界面，是系统必不可少的重要组成部分，大部分人认识 OS 都是从这里开始的。

由于 Skyeye 下的仿真串口 USART 已经实现了中断方式的接收（实际是从键盘接收输入），而且串口输出（实际上是输出到终端屏幕）也已经实现，所以实现一个类似 DOS 或 Bash 的简化版 Shell 并不困难。其本质思想就是：Shell 作为一个 uC/OS-II 下的任务，工作于内核之外，占用一个任务号。它接收用户输入的字符，存储到缓冲区，并回显在屏幕上，以回车键为用户输入的结束信号，随后解析用户输入的命令名称、参数，调用相应的命令函数。一直到这个命令函数运行返回，才继续 Shell 的人机交互界面。其流程图如下：



我们目前在 uC/OS-II 下实现的 Shell 起名为 Genie Shell，非常简单，只实现了最基本的命令输入、解析参数、调用命令函数功能，以及两条示例性的命令。这个 Shell 的特色是采

用了一些面向对象的思路来实现 Shell 的各种命令。Genie Shell 把每个命令看成一个对象，对象的属性是命令名，而对象的方法就是命令的执行函数本身。用户输入命令及参数后，将参数传递给对象的方法并执行。在 shell 中增加一条命令，就是增加一个对象，并实现这个对象的方法。在 c 语言中实际是一个带函数指针的结构体。

3. Applications on uClinux

uClinux-dist-20020816 和 uClinux-dist-20020927 已经可以在 SkyEye 上运行。目前已经在 uClinux 上成功地执行了 shell、ping、ifconfig、telnet 客户端（没有对应用程序进行任何修改）等应用程序，其它的应用还没有测试，相信网络方面的应用程序执行没有问题。

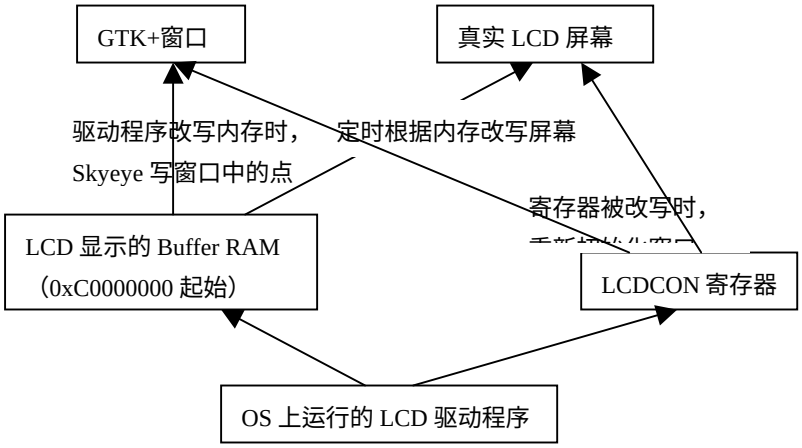
六、 SkyEye 正在开展的工作计划

1. LCD 仿真

作为单片机系统重要的输出设备，LCD 屏幕用的非常广泛。因此我们觉得有必要为 Skyeye 加入 LCD 控制器设备仿真，并使用 GTK+图形函数库在 Xwindow 中模拟出一个 LCD 屏幕，这样 Skyeye 上运行的 OS 中，LCD 驱动程序象驱动真正的 LCD 控制器一样发送控制命令，而 Skyeye 解释这些控制命令，并相应地在 GTK+绘出的窗口中画不同灰度或颜色的点。这里对 GTK+的使用仅限于绘制窗口和画点，因为这是 LCD 屏幕的最基本动作，其它所有的工作如画图，GUI 的实现都应该由应用程序利用画点的功能去实现，与 Skyeye 本身无关。

LCD 的 buffer ram 是映射到内存 ram 中的，LCD 屏幕的每个点对应了内存中的一位或几位，各个点所对应的内存按行序连续存放。LCD 上显示什么内容，完全由这块内存决定。LCD 控制器会根据这块内存去刷新屏幕显示，刷新频率等参数要根据具体使用的 LCD 屏幕来决定。OS 中的 LCD 驱动程序要往 LCD 屏幕上写东西，就是写这块内存。

Skyeye，LCD 仿真部份的示意图如下（包括与真实情况的比较）：



2. IDE 集成开发环境

IDE 集成开发环境是 SkyEye 的一个重要组成部分。由于 SkyEye 的用户接口是基于 GDB 的，所有支持 GDB 的 IDE 集成开发环境都可以很容易地移植为 SkyEye 的 IDE 集成开发环境。目前的 Windows 版本借鉴 Dev-Cpp 的源代码，使用 Delphi 编写（这样便于以后移植到 Kylix 中），主要与 cygwin 版本的 skyeye 进行通讯，以进行方便的、可视化的调试。目前的 Linux 版本是修改过的 DDD，以后可能使用 Windows 版本的移植。将来可能会采用功能强

大，可扩展性强的 Eclipse 集成开发环境作为 SkyEye 的 IDE 集成开发环境。

3. 更多的参与和支持

开发小组的主要联系和讨论方式是 maillist、jabber IRC 及时讨论、和 SkyEye 主页上的留言版。 采用小组分散式开发方式和 sourceforge 提供的开发环境，一般一个小组从（1到5人不等，由小组长负责），用 CVS 实现版本管理，强调分析和设计文档的写作工作。

SkyEye 的开发工作还很多，主要集中在 SkyEye 仿真环境的开发和基于 SkyEye 仿真环境的上层系统软件的开发。目前已经规划的子项目有：

项目	起始时间	进展度	成员 (*表示需要人手)	迫切度
实现初始版 SkyEye on Cygwin/window & linux	2002-12	100%	陈渝	*****
设计实现 uoosii for SkyEye 软件结构框架	2003-02	100%	李明	*****
分析实现 ucosii for Skyeye	2002-12	100%	陈渝、李明	*****
设计实现 ucosii for SkyEye 的 I/O 功能	2002-01	100%	李明	*****
分析实现 LwIP for ucosii for SkyEye	2003-01	100%	杨晔、李明、陈渝	*****
设计实现 SkyEye 的网络仿真	2003-01	100%	杨晔、陈渝	*****
设计实现 shell for ucosii on SkyEye	2003-02	100%	杨晔	*****
设计实现 uart ISR for ucosii on SkyEye	2003-02	100%	杨晔	*****
分析 redboot/armboot 对 AT91EV40 的支持	2002-12	90%	尹首一	*****
uClinux2.4.x on 开发板 AT91EV40	2003-02	85	尹首一	*****
分析 GDB5.0 for armulator 对 AT91EV40 的支持	2002-12	85%	陈渝、杨晔、王利明	*****
分析 uclinux2.4.x 对 GDB5.0 for armulator 的支持	2002-12	70%	陈渝、尹首一	*****
分析 uclinux2.4.x 对 AT91EV40 的支持	2002-12	70%	陈渝、尹首一	*****
分析基于 GDB5.0 for armulator 的结构	2002-12	50%	陈渝、杨晔、王利明 *	*****
设计实现 uclinux net driver for SkyEye	2003-02	50%	陈渝、尹首一 *	*****
分析和调试 DEV-C++源代码	2003-01	50%	王宇轩、韩怡昕 *	*****

实现 MMU 模拟模块 for SkyEye	2003-03	90%	王利民 宋振宇 楼克刚 *	*****
实现 SkyEye 支持仿真 Linux	2003-03	90%	王利民 宋振宇 楼克刚 *	*****
实现 LCD 模拟模块 for SkyEye	2003-03	10%	杨晔、康烁、赵俊良 *	*****
实现 DEV-C++对 SkyEye 的支持	2003-03	10%	王宇轩*	*****
在 Cygwin/windows 上实现 SkyEye 的网络仿真功能	N/A	0%	N/A *	N/A
分析和调试 Eclipse 源代码	N/A	0%	N/A *	N/A
设计实现 文件子系统 for ucosii on SkyEye	N/A	0%	N/A *	N/A
设计实现 图形子系统 for ucosii on SkyEye	N/A	0%	N/A *	N/A
设计实现各种应用 for ucosii on SkyEye	N/A	0%	N/A *	N/A
分析 RTAI 对 uclinux2.4.x 的支持（68knoMMU）	N/A	0%	N/A *	N/A
在 SkyEye 上安装并运行 uclinux2.5.x	N/A	0%	N/A *	N/A
在 SkyEye 上实现 armnoMMU RTAI+uclinux2.5.x	N/A	0%	N/A *	N/A

欢迎感兴趣，有热情和有一定空闲时间的朋友积极参加。请与项目负责人陈渝联系，
mailto: chenyu-tmlinux@hpclab.cs.tsinghua.edu.cn

SkyEye 项目开发小组
2003-3-14