doi: 10.7690/bgzdh.2014.03.027

基于龙芯 1A 平台的 PMON 源码编译和启动分析

吴昌昊，官琴
（中国兵器工业第五八研究所特种电子技术部，四川 绵阳 621000）

摘要：为降低开发人员理解龙芯（MIPS32）和 PMON 源码的难度，对基于龙芯 1A 平台的 PMON 源码编译和启动进行分析。采用流程图和文字相结合的形式，对 PMON 的目录结构、编译方法和执行流程进行研究。结果表明：该方法能清晰地展示 PMON 的编译和启动时期的工作机理，可为相关人员了解 PMON 工作方式和修改编译其源代码提供帮助。

关键词：龙芯；MIPS；PMON；编译；启动
中图分类号：TP316.8　　文献标志码：A

PMON Source Code Compiling and Starting Analysis Based on Loongson 1A Platform

Wu Changhao, Guan Qin
(*Department of Special Electronic Technology, No.* 58 *Research Institute of China Ordnance Industries, Mianyang* 621000, *China*)

Abstract: For decreasing difficulties of researcher understanding Loongson(MIPS32) and PMON source codes, analyze the compiling and starting of PMON source code based on Loongson 1A platform. Combine flow charts with character, research content structure, compiling method and execution process of PMON. The results show that the method can give a clear illustration about PMON's operating principle in compile-time and start-time. This article may help new developers and anyone interested in PMON learn and modify the source code.

Keywords: Loongson; MIPS; PMON; compile; start

0　引言

基本输入输出系统（Basic Input/Output System，BIOS），一般是指 x86 平台一种业界标准的固件接口，用于计算机开机时执行系统各部分的自检，并启动引导程序或装载在内存的操作系统[1]。

龙芯平台使用了名为 PMON（Prom Monitor）的程序作为 BIOS。PMON 具有强大的功能，包括硬件初始化、操作系统引导、硬件测试、程序调试等功能。同时它提供多种操作系统加载方式及多种硬件基础测试工具[2]。

然而，现阶段无论是在嵌入式、家用机、服务器或者是大型机的市场上，龙芯都属于小众，甚至可以说 MIPS 架构都很冷门，直接导致了龙芯平台相关硬件或软件项目参与的人员数量少，也就导致了相关文档资料很少。

为了能让更多人能更快地理解 PMON 源码，笔者针对龙芯 1A 平台的 PMON 代码的编译和启动过程进行分析。文中所使用的 PMON 是指龙芯中科针对龙芯 1A 发布的版本。

1　PMON 历史简介

PMON 原本是由 Phil Bunce 专为 LSI Logic MIPS R3000 评估板设计的一个免费 PROM 调试监控程序。从发布之日起，它就成为了许多 MIPS 平台开发系统的首选固件，但是原作者在 1999 年就停止更新了[3]。

后来出现的 PMON 2000 是由商业公司 Opsycon 开发的，改动巨大。该版本曾经也是最受欢迎的嵌入式系统固件之一[4]。从其官网可以发现，2008 年后 PMON 2000 也不再更新。

现在龙芯平台所用的 PMON 版本，是龙芯中科技术有限公司基于 PMON 2000 修改而来。

2　PMON 目录结构

PMON 的目录结构非常清晰，如表 1 所示。

表 1 中 tools 目录中包含多个工具，但实际上只需要用到里面的 pmoncfg，该程序是用来根据配置文件为 PMON 主程序的生成做准备工作。

最后一行的“zloader.xxx”是指存在于源码根目录中的软链接，比如“zloader.ls1a”。由于这些软链接都是指向同一个目录 zloader，所以使用起来并无差异，只是会影响 zloader 目录中文件的绝对路径值。创建这些软链接的目的是在编译时根据路径名使用不同的 Makefile。

收稿日期：2013-11-25；修回日期：2014-01-20
作者简介：吴昌昊（1990—），男，四川人，本科，助理工程师，从事 Linux、VxWorks 软件开发和多核并行运算研究。

表 1　PMON 目录名称及作用

目录名	作用
conf	通用配置文件
doc	文档
examples	程序例子
fb	SiS 显卡驱动、图形 logo 代码
include	通用头文件
lib	库源码，包含 libc、libm、libz
pmon	PMON 主体源码
sys	系统内核基础代码、部分驱动代码
Targets	目标板级相关的配置文件、头文件、源文件、目标文件
tools	编译辅助工具
x86emu	x86 模拟程序源码
zloader	解压缩程序
zloader.xxx	指向 zloader 目录的软链接

3　PMON 编译方法

PMON 的编译较为简单，以笔者的 Fedora 18 i686 系统为例，流程如图 1。

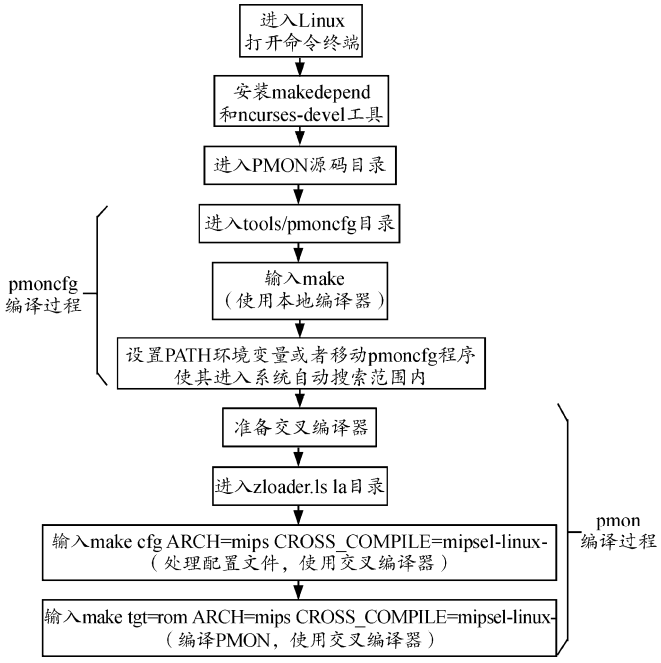

图 1　PMON 编译方法

由图可见，PMON 编译的操作很简单，最终得到 gzrom.bin 二进制文件，该文件可以通过网络、烧写器、EJATG 的方式写入板上 norFlash，然后开机上电后 CPU 就开始执行新编译的 PMON 代码。另外一种 gzram.bin 是可以直接放入内存执行的，暂不讨论。

4　PMON 编译执行流程

在 PMON 编译开始之前，首先得生成一个名为 pmoncfg 的工具，其编译执行流程非常简单，与一般的 C 语言小程序的编译过程很相似，即将多个.c 源文件编译生成.o 目标文件，再链接成可执行程序。但不同的是它利用了词法分析器 flex 和语法分析器 bison，其流程见图 2。

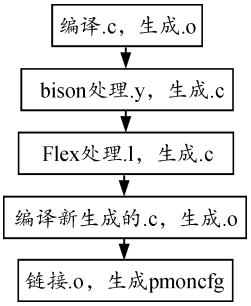

图 2　pmoncfg 工具编译执行流程

PMON 编译期的执行流程是本节的重点，它的编译流程复杂程度很高，涉及到多个 Makefile。这些 Makefile 之间的具体关系见图 3。

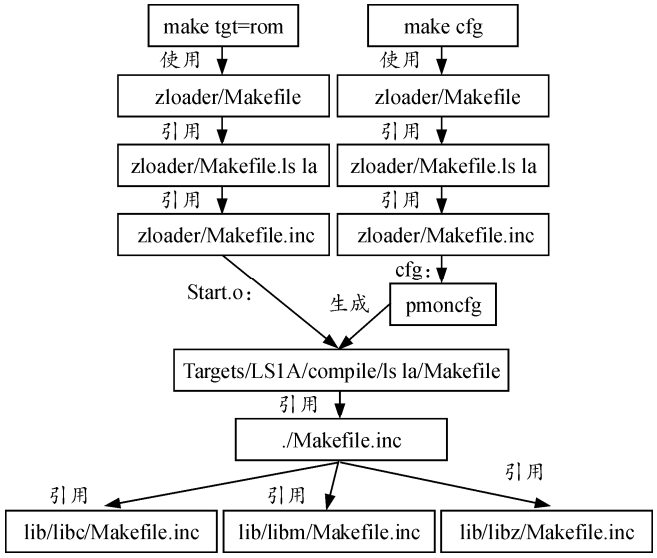

图 3　PMON 编译期各个 Makefile 关系

图中的路径都是对于源码根目录的相对路径。图中“引用”字样代表 makefile 中的“include”指令，即将某个文件原封不动地放进 include 指令所在位置。

在文中第 2 节中提到过“zloader.xxx”字样的软链接会影响使用的 Makefile，被影响的就是图 3 中从上往下数的第 3 层。本例中，使用 zloader.ls1a 进入目录，那么就会使用 Makefile.ls1a。

编译 PMON 最主要的是执行 make cfg(处理配置文件)和 make tgt=rom(生成可执行文件)这 2 步。从图 3 也可知道，make cfg 指令一定要先完成。

make cfg(处理配置文件)步骤的流程最关键的就是执行 pmoncfg 工具根据配置文件生成一个几千行的 Makefile，其余的流程都比较简单而且不影响理解源码，所以文中将其忽略。

make tgt=rom(生成可执行文件)步骤的流程要稍微复杂些，见图 4 所示的简化流程。

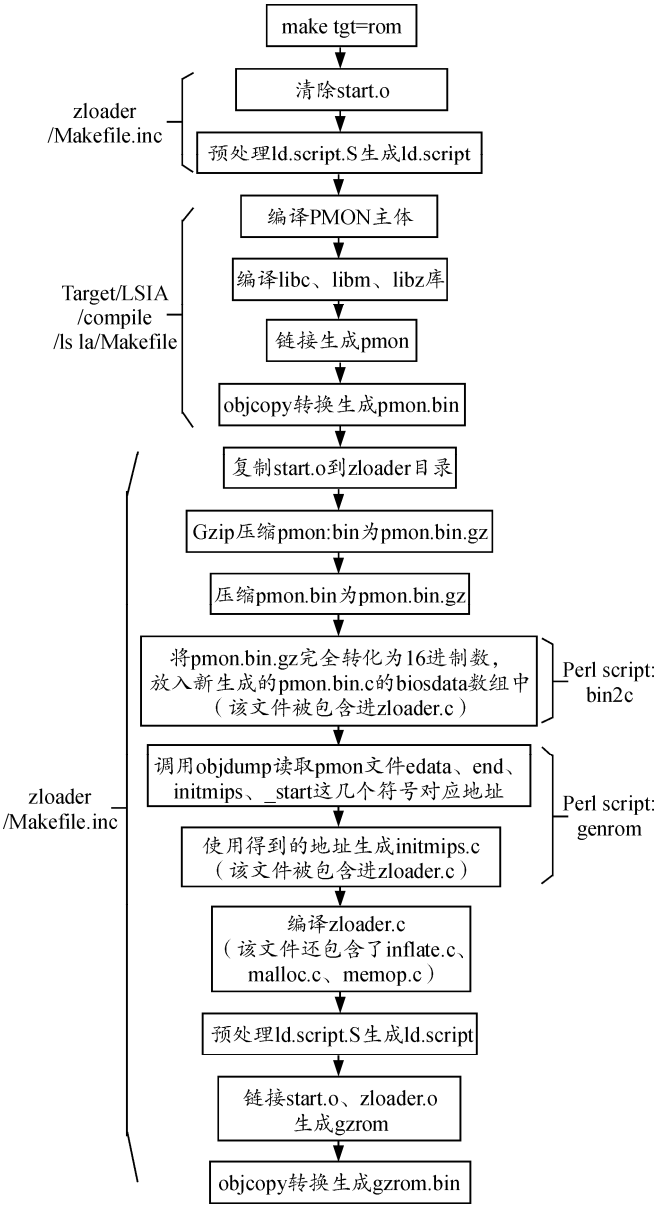

图 4　PMON 二进制文件编译执行流程

从图中可以看到，生成了 2 次 ld.script，实际上这 2 次生成的链接脚本的地址并不相同，也就是说生成的二进制文件起始地址不同。链接过程同样出现了 2 次，第一次是组装 PMON，第二次是组装 zloader。其实从这部分也就可以看出，zloader 并不是 pmon 的一部分，反而是 PMON 成为了 zloader 的一部分，因为最终链接 zloader 时指令较短，其简化后相当于：

mipsel-linux-ld -T ld.script -o gzrom start.o zloader.o

虽然 pmon.bin 已经包含了 start.o，但在 zloader.o 前面同样也包含了 1 次，因为该文件用于 CPU 上电后的关键硬件的初始化，必须保证它最先执行。

该过程还涉及到 2 个 perl 语言编写的脚本，它们的主要功能就是将 zloader 代码和 PMON 二进制内容组装在一起，以便 PMON 启动过程中能正确地在这两者间跳转。

5　PMON 执行流程

龙芯 1A 符合 MIPS32 标准，按照其架构要求，0xA0000000～0xBFFFFFFF（kseg1 段）是唯一在系统重启时能正常工作的内存映射空间[5]，并且地址 0xBFC00000 为重新启动时的入口向量，即 CPU 启动后第一条执行语句所在地址。

对于 kseg1 段的地址，通过将最高 3 位清零的方法，把这些地址映射为相应的物理地址，然后再映射到物理内存中 512 MB 大小的低位[5]。所以启动地址对应的物理地址为 0x1fc00000，且对应于 norFlash 的 0 地址。也就是说，刚开始的 PMON 是直接在 Flash 上执行的。

启动地址的映射关系，这完全是靠硬件设计来完成。在这之后，就要靠 start.S 所包含的源代码来处理接下来的初始化工作，然后跳转到 C 代码的入口继续执行。总的来说，启动过程见图 5。

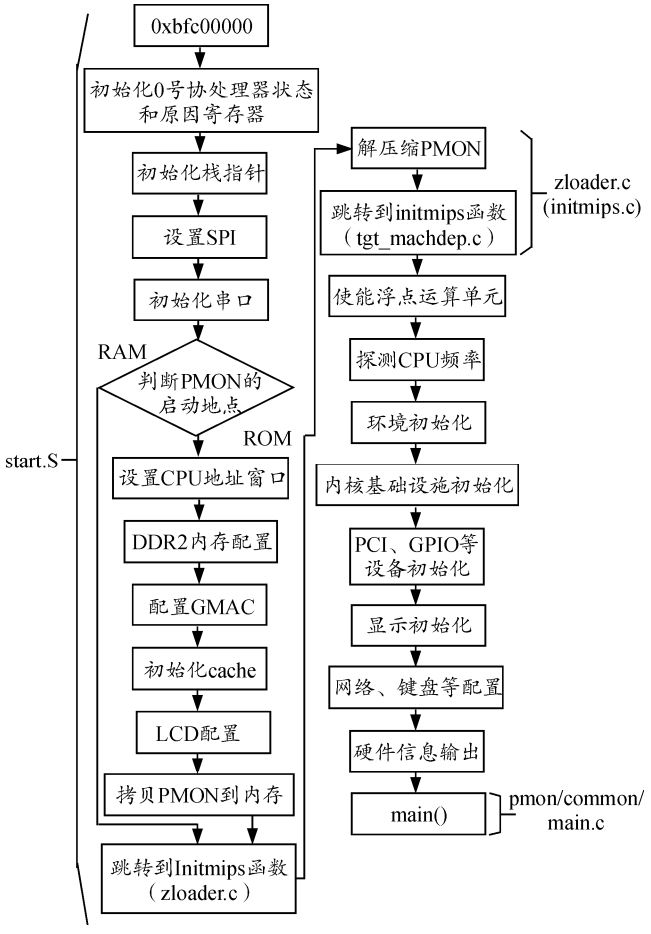

图 5　PMON 启动流程

从图中首先可以看出：一般情况下，PMON 会直接在 Flash 中运行，一段时间后才会将自身拷贝到内存中，并且还需要通过 zloader 将压缩过的 PMON 主程序解压[6]。

而且，PMON 还存在另一种启动流程，该流程是指以 tgt=ram 为编译目标时所得到的，完全运行在内存中，一般用作调试，笔者暂不讨论该情况。

在串口初始化之后，就会看到 PMON 输出的第一行字符串“PMON2000 MIPS Initializing. Standby…”。从这之后，就能不断地在串口终端上看到 PMON 的启动情况。

图中出现了 2 个 initmips 函数，第 1 个存在于自动生成的 initmips.c 中，而且该文件又被包含进 zloader.c，该函数作为 zloader 的入口，同时也是第一个执行的 C 代码程序；第 2 个存在于 tgt_machdep.c 文件，作为 PMON 主体的入口，直接从 zloader 的 realinitmips()函数中跳转过来，同时也对应于串口输出信息“OK,Booting Bios”之后。

initmips()函数（tgt_machdep.c）大部分的工作都是配置相关硬件、系统环境，在最后通过调用 main.c 提供的 main()函数，正式进入 PMON 的世界。

6 结束语

笔者用流程图和文字相结合的方式，较为清晰地描述了 PMON 编译时期和启动时期的执行流程，可为相关人员提供一定的帮助。

参考文献：

[1] Wikipedia. BIOS[EB/OL]. Wikipedia, 2004(2013-11-12). http://zh.wikipedia.org/zh/BIOS.
[2] 龙芯开源社区. PMON[EB/OL]. 龙芯开源百科(2012-6-20). http://www.loongson.cn/dev/wiki/pmon.
[3] LinuxMIPS. PMON[EB/OL]. LinuxMIPS(2010-2-8). http://www.linux-mips.org/wiki/pmon.
[4] LinuxMIPS. PMON 2000[EB/OL]. LinuxMIPS(2013-1-24). http://www.linux-mips.org/wiki/pmon_2000.
[5] Dominic Sweetman. MIPS 体系结构透视[M]. 2 版. 北京: 机械工业出版社, 2008: 33-35.
[6] 吴昌昊. 基于 Zentyal 的数据过滤程序[J]. 兵工自动化, 2013, 32(7): 89-92.

****

（上接第 92 页）

5.2 实验数据

为了验证优化后的效果，笔者将改进前全程直线运输策略比赛与改进后策略的进球数进行比较。由于 2 个策略的进球方法不一样，笔者将目的区域设定为易进球区，5 min 2 个策略进球的数量作为实验数据。如表 1～表 3 所示，进球数明显提高。

表 1 第 1 次测试数据

时间/min	改进前策略进球数	改进后策略进球数
1	0	0
2	0	2
3	0	4
4	1	5
5	2	7

表 2 第 2 次测试数据

时间/min	改进前策略进球数	改进后策略进球数
1	0	0
2	0	2
3	0	4
4	1	6
5	1	8

表 3 第 3 次测试数据

时间/min	改进前策略进球数	改进后策略进球数
1	0	0
2	0	2
3	0	3
4	1	5
5	2	7

5.3 实验结果分析

经过实验数据分析，优化后的策略比之前策略的进球数提高了 3 倍，并且不会出现运球过程中尾部碰触边界弹鱼的情况。能流畅地运输水球，在上半场将 7～8 个水球顶入己方球门，运输效率提高了 90%。实验证明，本策略是行之有效的。

6 结论

优化后的抢球大作战策略充分考虑了比赛场地的地形以及机器鱼和水球的实时信息，可以将整个地形运用起来，在不同的区域执行不同的策略，加强了机器鱼的运输和绕障能力。这只是机器人仿真研究中的冰山一角，笔者将通过深入研究，将机器人协作研究运用到实际生活中去。

参考文献：

[1] 谢广明. 机器人水球比赛项目推介书[M]. 北京: 北京大学工学院, 2009: 1-5.
[2] 黄永安, 马路, 刘惠敏. Matlab7.0/simulink 6.0 建模仿真开发与高级工程应用[M]. 北京: 清华大学出版社, 2007: 1-75.
[3] 龙海楠, 李淑琴, 安永跃. 仿真机器鱼抢球大作战策略的研究[J]. 计算机仿真, 2012, 30(7): 312-315.
[4] 安永跃, 李淑琴, 龙海楠, 等. 机器鱼仿真水球斯诺克比赛策略[J]. 兵工自动化, 2012, 31(11): 52-54.
[5] 陈晓, 李淑琴, 谢广明. 基于启发式路径评估的仿真机器鱼策略研究[J]. 北京信息科技大学学报, 2012, 28(1): 79-82.