

# Q/LS

Q/LS 0014-2014

## 龙芯中科技术有限公司企业标准

---

### 龙芯开发系统 **Linux** 内核开发 详细规范

2014-05-01 发布

2014-06-01 实施

龙芯中科技术有限公司

批准



# 目 次

前 言 .....	2
1 范围 .....	3
2 术语与定义 .....	3
2.1 内核文件分类 .....	3
2.2 模块与文件界定 .....	3
3 Patch 级别与要求 .....	3
4 编程规范 .....	4
4.1 总则 .....	4
4.2 地址空间配置 .....	4
4.3 物理地址空间的访问 .....	4
4.4 全局变量使用 .....	4
4.5 编译约定 .....	4
5 内核的测试指南 .....	5
6 人员角色 .....	6
7 版本库 .....	6
8 开发流程 .....	6
附录 A .....	7

## 前 言

本规范是龙芯中科技术有限公司制定的企业规范，暂无国家相关行业通用规范可参考。

本规范是对龙芯开发系统及其衍生兼容机的 Linux 内核开发、测试及发布过程的技术规定。

本规范的起草单位：龙芯中科技术有限公司。

本规范的主要起草人：高翔，孟小甫，段玮。

本规范审核人：王剑、成修治、王焕东、刘奇、邱吉、黄沛、刘绍宗、胡明昌、乔崇、李文刚。

本规范批准人：胡伟武。

# 龙芯开发系统 Linux 内核开发详细规范

## 1 范围

本规范规定龙芯 2 号、3 号开发系统的内核开发、测试及产品发布的详细要求，嵌入式开发系统不包括在内，但鼓励遵循本规范。

## 2 术语与定义

### 2.1 内核文件分类

内核文件分为三大类：源文件，配置文件及辅助文件。其格式见表 1。

表 1 内核文件分类

分类	名称	格式
源码文件	头文件	.h
	源码文件	.c
	固件文件	.bin
配置文件	功能配置文件	Kconfig, defconfig
	编译配置文件	Makefile, Rules.make
辅助文件	Script 目录	.script

### 2.2 模块与文件界定

根据龙芯平台 Linux 内核实际情况，将内核的源码文件分为三部分：驱动部分、体系架构相关部分以及公共部分。

- a) 驱动部分：包含内核 driver 目录下所有的文件。
- b) 体系架构相关部分：内核源码中与体系架构相关的代码按模块可以分为中断、TLB、Cache、DMA、总线、同步、内存地址空间、例外、时钟、综合等。内核不同版本的详细说明示例以附录形式规定。在内核 2.6.32 版本中，与体系架构相关的各模块涉及的文件示例见附录 A。
- c) 公共部分：未包含在体系结构相关部分与驱动部分的所有内核源文件都属于公共代码。

Kconfig 与 Makefile 等文件都是源码文件对应的配置文件。

附录 A 给出了以 loongson3 为示例的体系架构相关源码文件典型组成与目录列表。

## 3 Patch 级别与要求

内核 Patch 根据其提交的权限控制与技术特点，分为如下 5 级：

- a) 1 级，驱动级。涉及设备驱动的添加与功能完善，源码与配置文件的修改范围限定于 driver 目录，对配置与编译文件的修改一般限定于新增驱动的支持。测试工作以模块级测试为主。
- b) 2 级，架构级。涉及体系架构相关的功能添加与完善。可以修改架构相关文件，但不允许对配置文件进行修改。测试以体系架构相关的白盒测试为主。
- c) 3 级，芯片级。涉及对新增桥片或系统架构（如单路到双路）的支持。允许修改架构和公共部分的源码与配置文件。测试以白盒测试与系统黑盒测试相结合。内核组需要提交书面的设计方案与测试报告报系统部经理审批。
- d) 4 级，公共级。涉及内核公共部分代码的修改，与上游社区版本的同步更新也属于本级范畴。允许对所有源码和配置文件进行修改。内核组需要提交书面的设计方案与测试报告报系统部经理审批，必要时由系统部经理牵头组织公司层面的评审会进行审批。
- e) 5 级，主干级。内核主版本的升级（如 3.10 内核的移植）。对所有源码和配置文件进行修改。

内核组需要提交书面的设计方案与测试报告报系统部经理备案，必须由系统部经理牵头组织公司层面的评审会，由公司主管领导审批。

Patch 的配置控制说明如表 2。

表 2 Patch 的配置控制

级别	级别名	执行权限	源码文件修改范畴		配置文件修改范畴		典型示例
			驱动	公共	驱动	公共	
1 级	驱动级	内核组组长	驱动	√	驱动	√	设备驱动的添加与功能完善
			架构	×	架构	×	
			公共	×	公共	×	
2 级	架构级	内核组组长	驱动	√	驱动	×	与底层体系结构相关的 DMA 功能完善、多核支持等
			架构	√	架构	×	
			公共	×	公共	×	
3 级	芯片级	系统部经理	驱动	×	驱动	×	新增桥片或系统架构支持
			架构	√	架构	√	
			公共	√	公共	√	
4 级	公共级	系统部经理	驱动	√	驱动	√	架构无关的优化，与上游版本更新
			架构	√	架构	√	
			公共	√	公共	√	
5 级	主干级	公司主管领导	驱动	√	驱动	√	新内核版本的移植开发
			架构	√	架构	√	
			公共	√	公共	√	

## 4 编程规范

### 4.1 总则

龙芯 Linux 内核编程除需要遵循标准的 Linux 内核代码编程规范之外，还需要遵循本章要求。

### 4.2 地址空间配置

龙芯平台的地址空间配置由 BIOS 进行，Linux 内核只对其进行验证，不得在内核代码中对芯片的地址空间重新进行配置。

### 4.3 物理地址空间的访问

所有提交的 patch 代码中，对物理地址寄存器的写访问必须要采用统一的宏定义地址的方式进行，禁止对物理地址直接赋值。

驱动代码中不得出现对非设备寄存器的更改访问。

CPU 芯片架构相关的寄存器的地址定义，桥片相关寄存器地址的定义也均需存放固定的头文件中，不得随意在.c 源码文件中添加寄存器地址的宏定义。

### 4.4 全局变量使用

架构与驱动代码中要严格限制全局变量的使用。

驱动代码中不得采用“extern”变量方式对系统全局变量进行修改。

### 4.5 编译约定

开发工程师进行编码时，必须要确定提交的 patch 在固定版本下无 warning 等警告信息出现。内核的版本对应到固定的 GCC 版本。开发组向测试组提交内核版本时，应填写表 3，并进行存档。

### 4.6 Patch 提交的要求

开发工程师提交 patch 时，要进行说明。对 Bug 类的 patch，要符合“定位准确、机理清楚、问题复现、措施有效、举一反三”的要求。

表 3 内核编译构建检查表示例

项目	情况	要求
编译时间		
对应内核版本哈希值		
主机环境		
编译器版本		
是否完全清楚从头编译		
是否出现 warning 信息		
编译过程		
编译系统主机配置		
编译输出文件存放位置		
.....		
编译输出文件存放位置		
编译依赖		
版本信息对应 GIT 哈希值		

## 5 内核的测试指南

内核的测试分为架构测试，驱动测试，系统发布测试。内核测试组应依据表 4 的要求制定详细的测试规范。

表 4 测试规范制定要求

测试分类	测试专项	要求
架构测试	地址空间测试	地址空间全范围扫描测试，确保无地址空洞
	中断测试	确定内核所有中断号处理正常。
	DMA 测试	对内核 DMA 机制的专项测试，包括对 DMA 地址转换的专项测试
	TLB 测试	对系统 TLB 处理模块进行专项测试
	Cache 测试	对系统 Cache 管理接口进行测试
	多核测试	对多核多线程调度机制进行测试
	内存管理测试	对系统的内存管理进行极限空间分配、交换分区等进行压力测试
	BIOS 兼容性测试	对 BIOS 规范的测试，采用小程序/模块来保证
	系统硬件配置	确定芯片及系统的全局参数是否设置为期望目标（重要，用于解决诸如内存控制器参数设置引起的各类系统问题）
驱动测试	网卡	用 netperf 等测试集对网卡驱动进行专项测试。
	硬盘	采用 iozone 等测试集对硬盘驱动部分进行专项测试
	GPU	采用 Xl1perf 等 2D/3D 测试程序对对显示驱动系统进行测试
	USB	对 USB 子模块进行测试。
	EJTAG	对系统 EJTAG 等进行测试
	其它	对串口等驱动进行专项测试
系统发布测试	LTP	验证内核的系统调用功能。
	Spec2000	常规测试。
	Unixbench	内核系统功能专项测试。
	重启	验证系统软硬件配合接口是否正常
	图形 UI 测试	窗口应用是否正常。
	稳定性测试	重负载情况下达到 12*10 功能目标

## 6 人员角色

人员角色配置要求如表 5 所示。

表 5 人员角色配置要求

人员	职责
内核组组长	总体把控内核源码的演进以及日常研发管理工作。
配置管理员	负责确定内核的源码等配置项变更管理。
开发工程师	内核相关模块代码的开发与维护。
测试组长	负责内核测试工作的总体安排与协调。
测试工程师（开发阶段）	负责开发阶段的测试与确认工作，在 Gerrit 流程中，处于 code review 之后、进入受控库的环节。此阶段测试人员需要与研发人员紧密配置，保证进入受控库前，代码能够通过模块和架构级专项测试。
测试工程师（系统发布阶段）	负责系统层面的内核完备性与稳定性测试工作。
发布工程师	负责对外发布包的管理。

## 7 版本库

版本库包含开发库、受控库和存档库三个部分，描述见表 6。

表 6 库描述

库	描述
开发库	内核开发人员自己本地维护，需定期与受控库同步。
受控库	内核的日常开发与演进的主版本库，是开发组与测试组的接口库。由内核负责人进行统一维护，接受来自各开发人员的 patch，经过审核和确认流程后进入受控库。代码的审核采用 Gerrit 系统进行，权限控制和代码审核及测试确认由内核负责人管理。
存档库	阶段性里程碑版本的存档，一般以完成了严格系统级可靠性与稳定性测试作为标准，发布前报系统部经理审批，对外发布的内核版本都源自存档库。

## 8 开发流程

开发所用主要的支撑平台包括 Git 源码库，Gerrit 代码审查系统，Bugzilla 缺陷跟踪系统。开发流程见表 7，其中基本测试项目包含架构测试和驱动测试，本规范不对自测试项目进行规定。

表 7 开发及管理流程

人员角色	研发部	各事业部	相关客户	所属库	管理软件
开发工程师	其他 patch	驱动 patch	驱动 patch	开发库	Git
开发工程师	自测试	自测试			
内核组长	代码审查				Gerrit
测试工程师	基本测试				
内核组长 或评审组	收入受控库			受控库	Git
测试组长	发布测试				
配置管理员 (经批准)	收入存档库			存档库	Bugzilla
发布工程师	发布				



## 附录 A

### 源码文件典型配置

#### A.1 源码文件典型配置示例

内核版本 2.6.32 中涉及到龙芯 3 号 CPU 架构相关目录见表 A.1。根据龙芯 3 号开发系统的软硬件架构,在 arch/mips/loongson3 目录下统一管理与芯片、板卡相关目录。龙芯 3A 与 3B 在操作系统层面可以做到兼容支持,所以在内核代码中对 3A、3B 不做区分。arch/mips/loongson3 目录包含 common、rs780e 与 1s2h 三个下级目录。其中 common 目录存放 3 号开发系统公共的文件,rs780e 目录放置与 3 号+rs780E 系统相关的文件,1s2h 放置与 3 号+2h 系统相关的文件。龙芯 3 号与 rs780e 架构相关文件见表 A.2。

表 A.1 内核 2.6.32 体系架构相关目录列表

目录名	说明
Arch/mips	Mips 架构相关源码文件总目录
Arch/mips/include	Mips 架构相关源码.h 头文件总目录
Arch/mips/include/asm	MIPS 架构汇编级.h 头文件目录
Arch/mips/include/asm/mach-loongson3	与龙芯 3 号板级相关源码目录
Arch/mips/loongson3	龙芯 3 号相关的中断、PCI、初始化源码目录
Arch/mips/loongson3/common	龙芯 3 号下中断处理等通用源码目录
Arch/mips/loongson3/rs780	龙芯 3 号 780E 桥片相关源码目录
Arch/mips/pci	MIPS 架构下 PCI 相关源码目录
Arch/mips/mm	MIPS 架构 TLB、Cache 等存储子系统管理源码目录
Arch/mips/kernel	Mips 架构下内核源码目录

表 A.2 内核 2.6.32 体系架构相关文件列表

模块化分	目录
中断	arch/mips/loongson3/common/irq.c
	arch/mips/loongson3/rs780/irq.c
	arch/mips/kernel/irq.c
	arch/mips/kernel/irq_cpu.c
	arch/mips/kernel/i8259.c
	arch/mips/include/asm/irq_cpu.h
	arch/mips/include/asm/irqflags.h
	arch/mips/include/asm/irq.h
	arch/mips/include/asm/irq_regs.h
	arch/mips/include/asm/mach-loongson3/irqregs.h
TLB	arch/mips/mm/tlb-r4k.c
	arch/mips/mm/tlbex.c
	arch/mips/mm/fault.c
	arch/mips/mm/tlbex-fault.S
	arch/mips/mm/hugetlbpage.c
	arch/mips/mm/pgtable-64.c
	arch/mips/include/asm/tlbflush.h
	arch/mips/include/asm/tlb.h

表 A.2 (续) 体系架构相关部分文件列表

模块化分	目录
TLB	arch/mips/include/asm/pgtable-64.h
Cache 操作	arch/mips/mm/c-loongson3.c
	arch/mips/mm/cache.c
	arch/mips/include/asm/cache.h
	arch/mips/include/asm/cacheflush.h
	arch/mips/include/asm/cacheops.h
	arch/mips/include/asm/cachectl.h
	arch/mips/include/asm/r4kcache.h
DMA	arch/mips/include/asm/dma-mapping.h
	arch/mips/include/asm/dma.h
	arch/mips/mm/dma-default.c
	arch/mips/loongson3/common/dma-swiotlb.c
	arch/mips/include/asm/mach-loongson3/dma-coherence.h
PCI	arch/mips/pci/fixup-loongson3.c
	arch/mips/pci/ops-loongson3.c
	arch/mips/pci/pci.c
	arch/mips/loongson3/common/pci.c
	arch/mips/include/asm/pci.h
	arch/mips/include/asm/mach-loongson3/pci.h
同步、锁	arch/mips/include/asm/spinlock.h
	arch/mips/include/asm/barrier.h
	arch/mips/include/asm/atomic.h
	arch/mips/include/asm/futex.h
	arch/mips/include/asm/cmpxchg.h
	arch/mips/include/asm/mutex.h
	arch/mips/include/asm/spinlock_types.h
	arch/mips/include/asm/signal.h
	arch/mips/include/asm/system.h
	arch/mips/include/asm/wbflush.h
arch/mips/include/asm/bitops.h	
例外处理	arch/mips/mm/cex-gen.S
	arch/mips/kernel/genex.S
	arch/mips/mm/extable.c
	arch/mips/kernel/traps.c
	arch/mips/kernel/unaligned.c
	arch/mips/include/asm/traps.h
时钟	arch/mips/kernel/time.c
	arch/mips/loongson3/common/time.c
	arch/mips/loongson3/rs780/rtc.c
	arch/mips/kernel/csrtc-r4k.c

表 A.2 (续) 体系架构相关部分文件列表

模块化分	目录
时钟	arch/mips/kernel/cevt-r4k.c
	arch/mips/kernel/sync-r4k.c
	arch/mips/include/asm/cevt-r4k.h
	arch/mips/include/asm/r4k-timer.h
	arch/mips/include/asm/cputime.h
	arch/mips/include/asm/time.h
	arch/mips/include/asm/mach-generic/mc146818rtc.h
	arch/mips/include/asm/timex.h
综合	arch/mips/loongson3/common/mem_init_smp.c
	arch/mips/loongson3/common/mem_init_numa.c
	arch/mips/loongson3/common/env.c
	arch/mips/kernel/cpu-probe.c
	arch/mips/kernel/smp.c
	arch/mips/loongson3/common/smp.c
	arch/mips/kernel/syscall.c
	arch/mips/kernel/signal.c
	arch/mips/kernel/process.c
	arch/mips/kernel/r4k_switch.S
	arch/mips/kernel/reset.c
	arch/mips/loongson3/common/init.c