Linux 内核原理

Amlogic Beijing
Zhouzhi
2009-12-7

主题

- Linux System Start Steps
- Interrupter Manage
- Memory Manage
- Cache Manage
- Task Manage
- Linux Kernel Module

Linux System Start Steps

Uboot

- Init DDR
- Prepare boot paramters

Load kernel

- Load kernel image form nand/tftp server
- Uncompress the kernel, and get the start section address

Goto kernel

- Goto the Kernel's start setion
- /arch/arc/proc/arc700/head.S->line37

Linux System Start Steps

Prepare C Start

- Save the kernel parameters
- Clear bss, setup stack

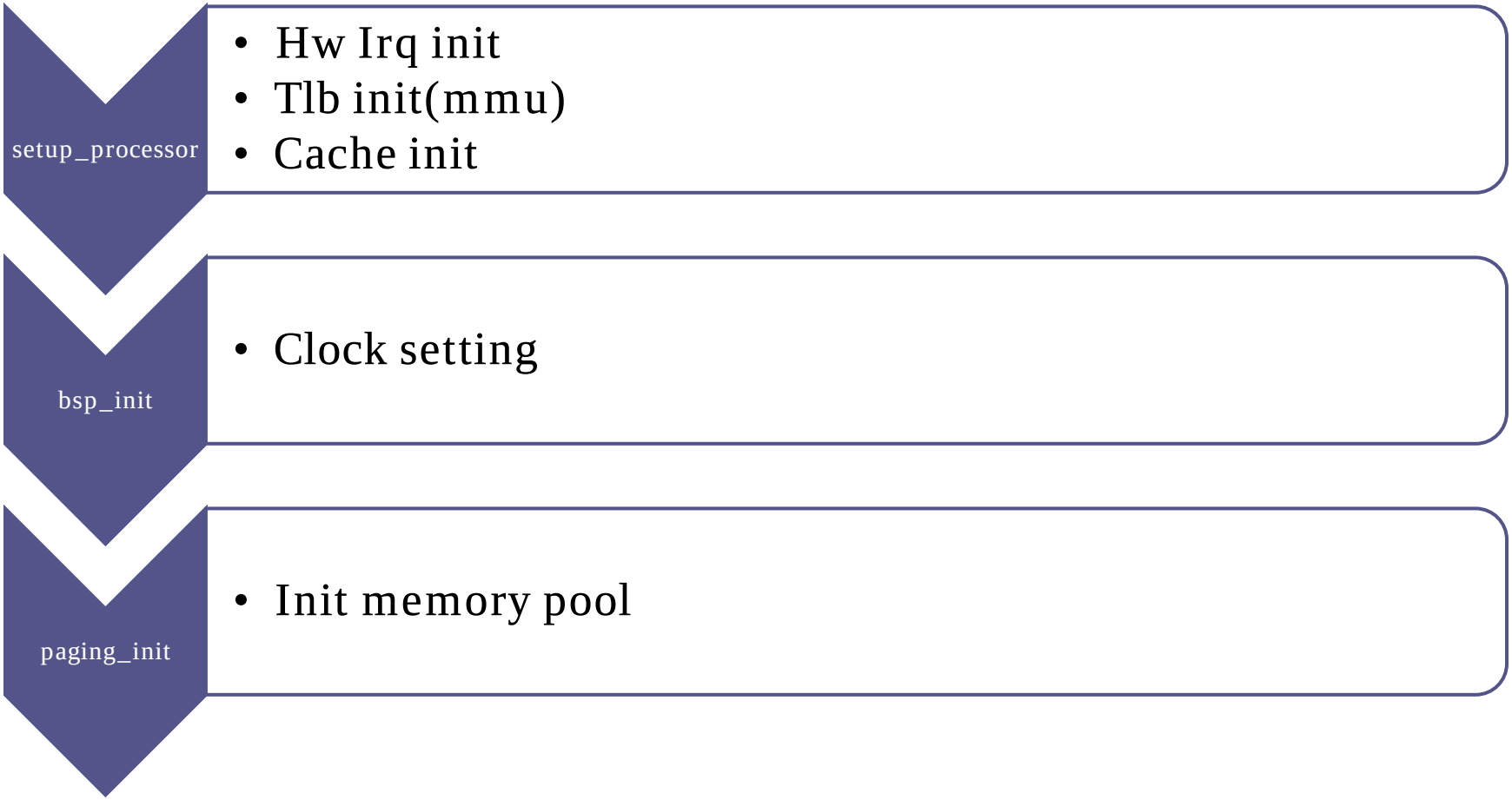
Start_kernel

- start_kernel
- Init/main.c

Setup_arch

- Parser the boot command line

Linux System Start Steps



Linux System Start Steps

System init

- task init->Irq init->Timer init->Softirq init
- Mm_init->console_init->...

rest_init

- kernel_init(start a init thread)
- Schedule(switch to kernel_init thread)

kernel_init

- Smp init
- do_basic_setup

Linux System Start Steps

do_basic_setup

- init_workqueues();
- usermodehelper_init();
- driver_init();
- init_irq_proc();
- do_initcalls();

do_initcalls

•#define pure_initcall(fn)	__define_initcall("0",fn,0)	
•#define core_initcall(fn)	__define_initcall("1",fn,1)	
•#define core_initcall_sync(fn)	__define_initcall("1s",fn,1s)	
•#define postcore_initcall(fn)	__define_initcall("2",fn,2)	
•#define postcore_initcall_sync(fn)	__define_initcall("2s",fn,2s)	
•#define arch_initcall(fn)	__define_initcall("3",fn,3)	->bsp_init(),modules parameters init
•#define arch_initcall_sync(fn)	__define_initcall("3s",fn,3s)	
•#define subsys_initcall(fn)	__define_initcall("4",fn,4)	
•#define subsys_initcall_sync(fn)	__define_initcall("4s",fn,4s)	
•#define fs_initcall(fn)	__define_initcall("5",fn,5)	
•#define fs_initcall_sync(fn)	__define_initcall("5s",fn,5s)	
•#define rootfs_initcall(fn)	__define_initcall("rootfs",fn,rootfs)	
•#define device_initcall(fn)	__define_initcall("6",fn,6)	→Drivers init
•#define device_initcall_sync(fn)	__define_initcall("6s",fn,6s)	
•#define late_initcall(fn)	__define_initcall("7",fn,7)	->alsa driver init
•#define late_initcall_sync(fn)	__define_initcall("7s",fn,7s)	

Linux System Start Steps

Drivers init

- obj-y += uart/
- obj-y += nand/
- obj-y += net/
- obj-y += usb/
- obj-y += amports/
- obj-y += i2c/
- obj-y += input/
- obj-$(CONFIG_CARDREADER) += cardreader/
- obj-y += audiodsp/
- obj-y += sound/
- /drivers/amlogic/Makefile

Linux System Start Steps

prepare_namespace

- driver_probe_done
- Prepare root device.(root=/dev/nfs)

mount_root

- mount_nfs_root
- change_floppy
- mount_block_root(hd/nand/spi)

init_post

- free_initmem();
- Open init console
- run_init_process(execute_command); /*init=/init*/
- run_init_process("/sbin/init");
- run_init_process("/etc/init");
- run_init_process("/bin/init");
- run_init_process("/bin/sh");

Linux System Start Steps

init_main

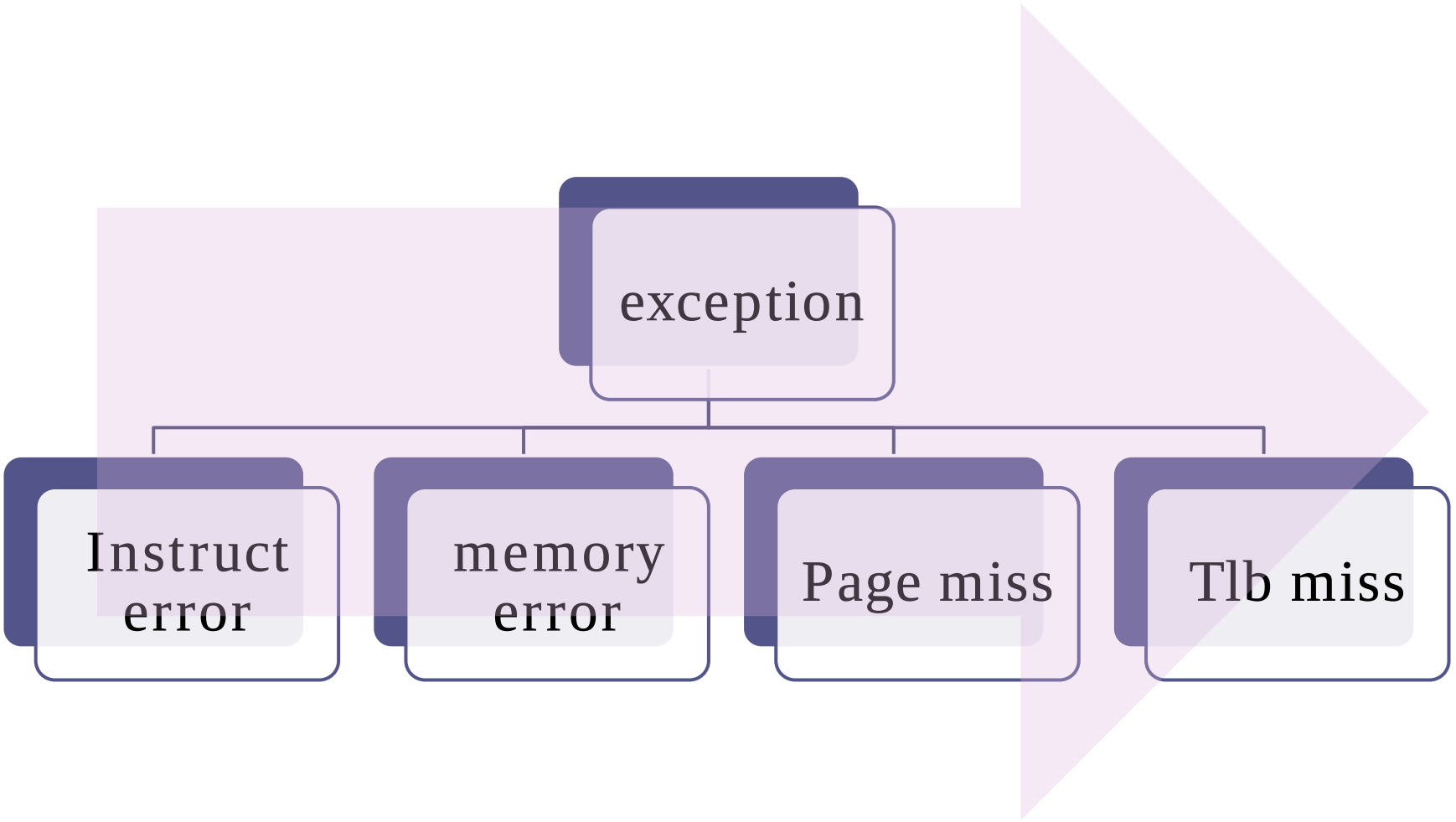
- Busybox/init/init.c
- parse_inittab
- ::sysinit:/etc/init.d/rcS

rcS

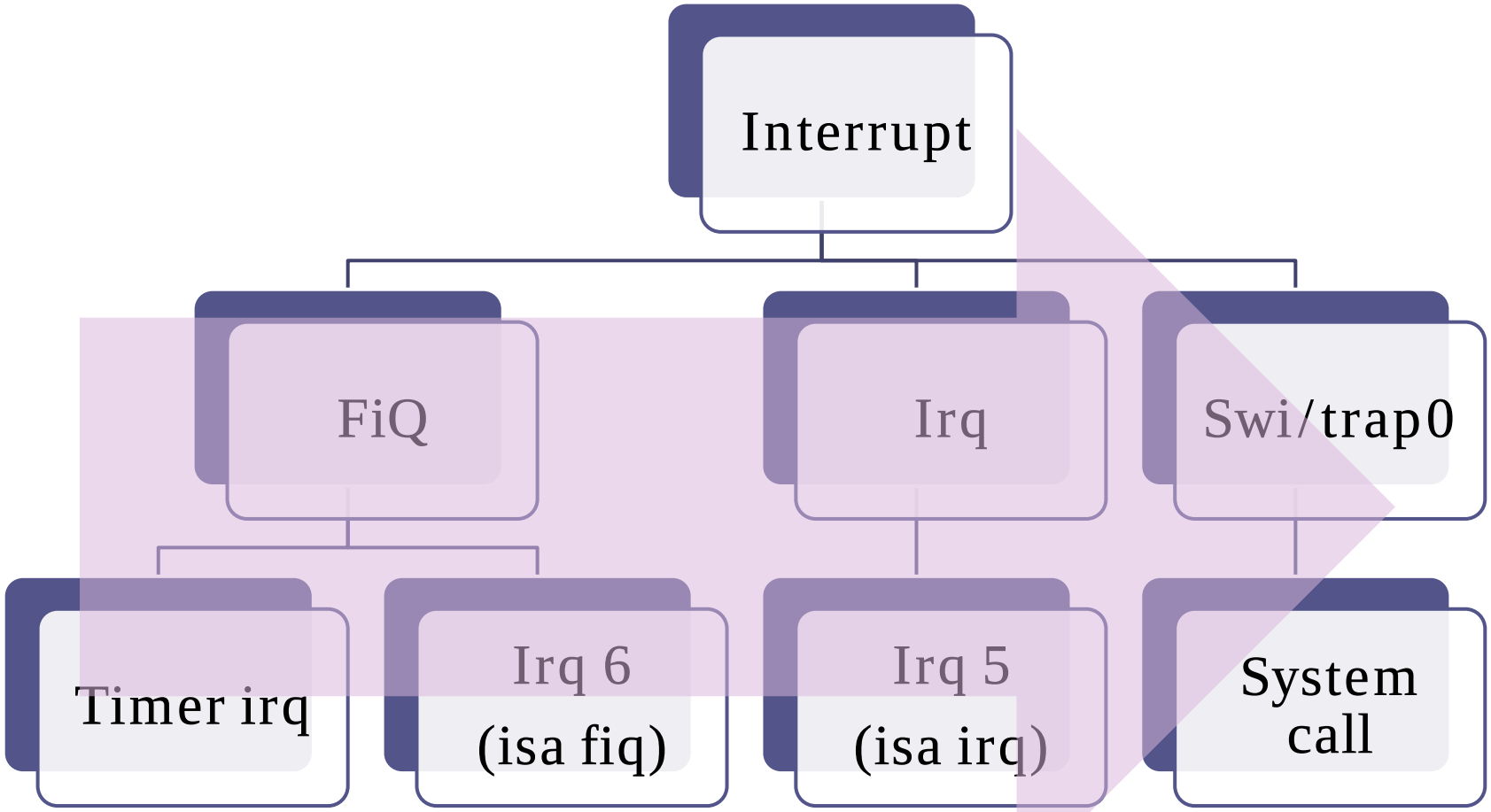
- /etc/init.d/rcS
- Insmod modules
- Mount proc, tmpfs,u sbfs, devfs, sysfs, devpts
- Netconfig
- Telnetd
- Changed the hotplug for mdev

Interrupt Manage

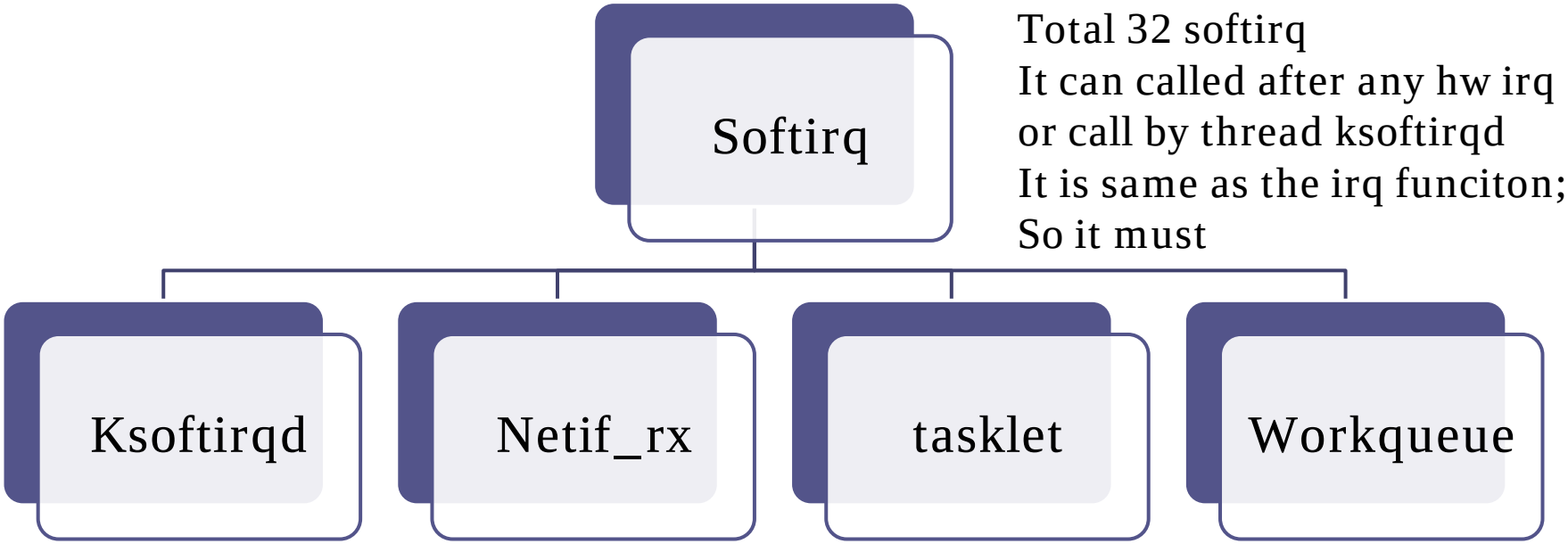
Interrupt Manage



Interrupt Manage



Interrupt Manage



Total 32 softirq
It can called after any hw irq
or call by thread ksoftirqd
It is same as the irq funciton;
So it must

Interrupt Manage

- Isa irq manager
  - #define AM_ISA_GEN_IRQ(v) (CORE_IRQ+v)
  - #define AM_ISA_GEN_IRQ_MAX() AM_ISA_GEN_IRQ(32)
  - #define AM_ISA_GEN1_IRQ(v)
    (AM_ISA_GEN_IRQ_MAX() +v)
  - #define AM_ISA_GEN1_IRQ_MAX() AM_ISA_GEN1_IRQ(16)
  - #define AM_ISA_GPIO_IRQ(v)
    (AM_ISA_GEN1_IRQ_MAX()+v)

  - #define AM_ISA_GPIO_IRQ_MAX() AM_ISA_GPIO_IRQ(32)
  - #define AM_ISA_AMRISC_IRQ(v)
    (AM_ISA_GPIO_IRQ_MAX()+v)
  - #define AM_ISA_AMRISC_IRQ_MAX()
    AM_ISA_AMRISC_IRQ(32)
  - #define AM_ISA_IRQ_MAX
    AM_ISA_AMRISC_IRQ(32)

Interrupt Manage

- Request hw isa irq
  - int request_irq(unsigned int irq, irqreturn_t (*handler)(int, void *),unsigned long flags, const char *name, void *dev_id)
  - Irq:irq number, AM_ISA_GEN_IRQ(31)
  - Flags:
    - IRQF_SHARED
    - IRQ_ISA_FAST /*fiq, fast irq*/
  - dev_id: do not used 0,

Memory Manage

- Slab->normal memory manager system;
- Slob->Designed for embeded system memory malloc; it is fast and low cost for less memory;
- Slub->huge memory manager system
- Physic memory manage
  - Page (mmu mange unit,8K size for arc)
  - __get_free_pages();
  - __free_pages();

Memory Manage

- Kmem_cache
  - kmem_cache_create
  - kmem_cache_destroy
  - kmem_cache_alloc
  - kmem_cache_zalloc
  - kmem_cache_free
- Used for mass kernel const size object: task_struct, file inode, skb header,kernel object; It is the final interface for malloc

Memory Manage

- Many user malloc functions:
  - kmalloc(size_t size, gfp_t flags);
    - Flags: __GFP_ZERO;GFP_NOWAIT; GFP_KERNEL ...
  - void *kzalloc(size_t size, gfp_t flags);
    - Same as kmalloc(size, flags | __GFP_ZERO);
  - void kfree(const void *);
  - size_t ksize(const void *);
    - The return size is always (2^n);

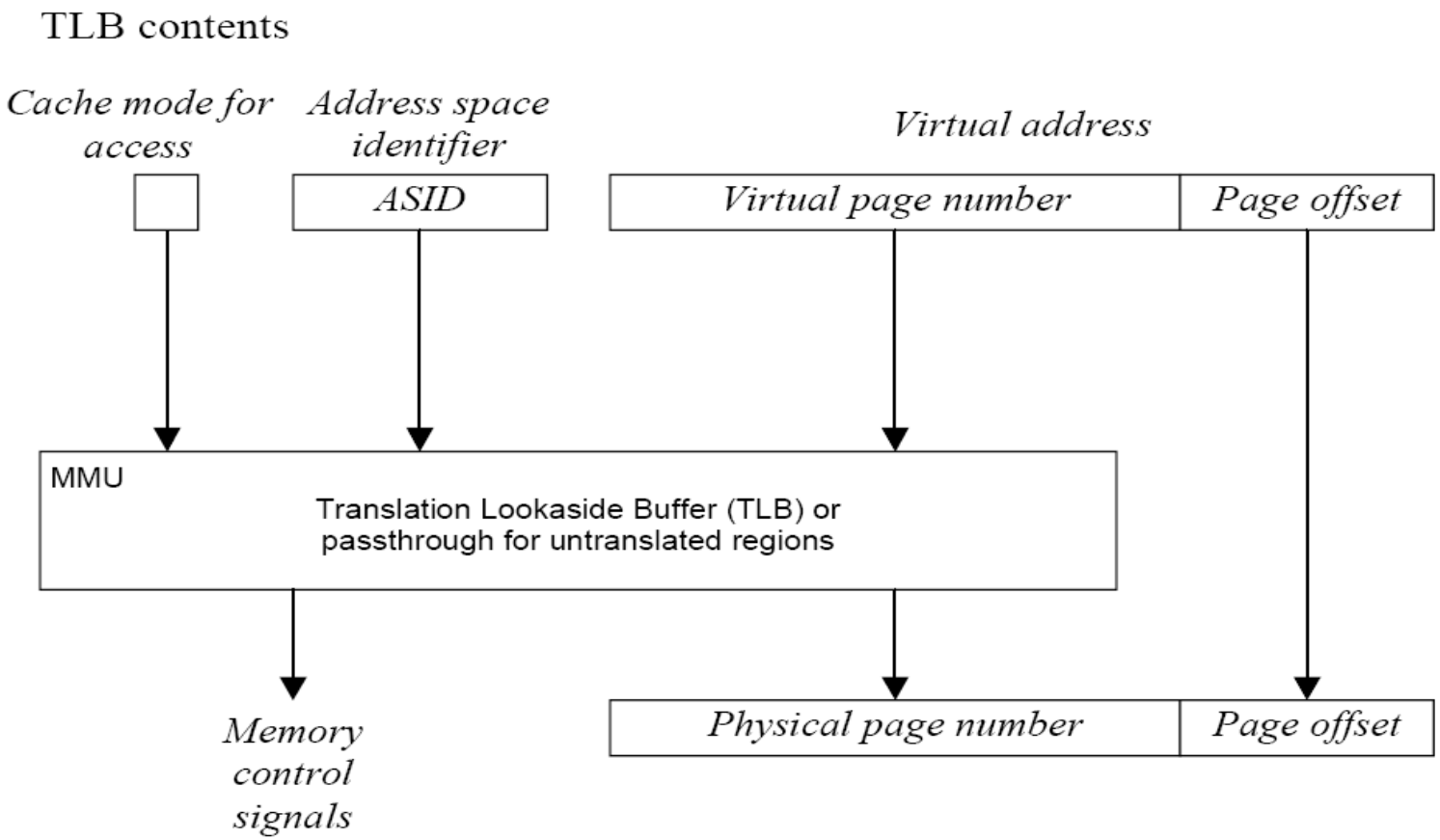
- Memory Manage-Memory mapping

Addr	option	purpose
0x00000000~ 0x20000000	Virtual address	A programs memory space, code and heaps
0x20000000~ 0x60000000	Virtual address	Mem maping address, for shared libraries, and stack
0x60000000~ 0x70000000	Virtual address	Kernel VM address, for kernel mmap();
0x70000000~ 0x80000000	Reversd	Keep space between the Virtual and Kernel phy memory;
0x80000000 0xc0000000	DDR	Ddr memory space
0xc0000000	Register address	No cache memory space, for register

Memory Manage

- Sample Mapping
  - 00010000-00066000 r-xp 00000000 00:0a 4607596 /bin/busybox
  - 00066000-00068000 rw-p 00054000 00:0a 4607596 /bin/busybox
  - 00068000-00082000 rwxp 00068000 00:00 0 [heap]
  - 20000000-20008000 r-xp 00000000 00:0a 4608002 /lib/ld-uClibc-0.9.29.so
  - 20008000-2000a000 rw-p 00006000 00:0a 4608002 /lib/ld-uClibc-0.9.29.so
  - 2000a000-2001e000 rwxp 2000a000 00:00 0
  - 2001e000-20020000 rw-p 2001e000 00:00 0
  - 20022000-200ae000 r-xp 00000000 00:0a 4609441 /lib/libuClibc-0.9.29.so
  - 200ae000-200b0000 rw-p 0008a000 00:0a 4609441 /lib/libuClibc-0.9.29.so
  - 200b0000-200ca000 rw-p 200b0000 00:00 0
  - 5ff1c000-5ff46000 rwxp 5ffd6000 00:00 0 [stack]

Memory Manage-(HW MMU)



Memory Manage-(HW MMU)

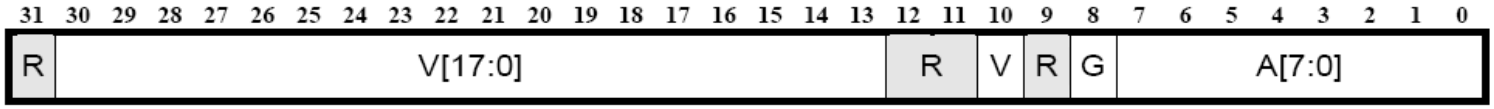


Figure 3 TLBPD0 Page Descriptor

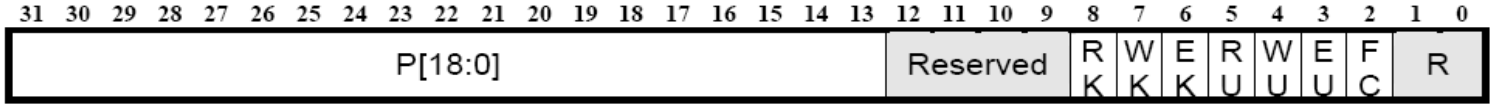


Figure 4 TLBPD1 Page Descriptor

- V[17:0] - Virtual Page Number
- V - Valid
- G - Global
- A[7:0] - Address Space Identifier ASID
- P[18:0] - Physical Page Number
- RK, WK, EK - Kernel Mode Permission Bits
- RU, WU, EU - User Mode Permission Bits
- FC - Cached/Uncached Flag

Memory Manage-(SW MMU)

address	31--------24	23---------------13	12 ---------0
name	pgd_offset	pte_offset	Offset_in_page
Description	Page group dir	Page table	offset

- Address
  - Page_size=(8192)
  - Pgd_offset=(address >>24)
  - Pte_offset =(address >>13) & (2048 -1)
  - PGD=(task->mm->pgd | (Address >>24));
  - PMD=(PGD);
  - PTE= (PGD+ ((address) >> 13) & (2048- 1))

Memory Manage-(SW MMU)

- MAP(0x81000000→0x0)
- Task->mm->pgd

PDG	PGD0(0<<24=0)			
PTE	Pte0	pte1	……	pte2047
V-address	PGD | 0(0<<13)	PGD | 0x2000(1<<13)		
Value of PTE	0x81000000 | Permits_bits			
	……			

Cache Manage

- Cache Line Size(32Bytes)
- Dcache
- Icache
- flush(write back)
- Invalidate_flush(write back and clear)
- Invalidate(clear dcache data)
- ahb_cache(invalidate ahb cache before DMA start)

Cache Manage

- Basic functions(Don’t used it if not necessary)
  - flush_dcache_range()
  - flush_dcache_all()
  - flush_and_inv_dcache_all()
  - flush_and_inv_dcache_range()
  - inv_dcache_range()
  - inv_dcache_all()
  - invalidate_ahb_cache();
  - …

Task Manage

- Kernel basic task functions
  - kthread_create();
  - kthread_should_stop();(call by thread itself,it return ture,return;)
  - kthread_stop;(call by others to stop a thread);
  - Schedule();(main task switch fucntion,It can used for switch to other task)

Task Manage

- Task wait some thing completion
  - wait_for_completion();
  - wait_for_completion_timeout();
  - Complete();//wake once
  - complete_all();//wake all the task;

Task Manage

- mutex
  - mutex_unlock()
  - mutex_lock()
  - mutex_init()
  - ...
- spin
  - spin_lock()
  - spin_unlock()
  - spin_lock_irqsave()
  - spin_unlock_irqrestore()
  - ...

Task Manage

- semaphore
  - Down();
  - Up();
  - down_killable();
  - down_interruptible();
  - down_timeout();

Task Manage

- workqueue(thread)
  - INIT_WORK();
  - DECLARE_WORK(n, f);
  - schedule_work();
  - schedule_delayed_work();
  - cancel_delayed_work();
  - ...

Task Manage

- Tasklet
- Run in Softirq, It is fast than workqueue
  - tasklet_init();
  - tasklet_schedule();
  - tasklet_kill();

Task Manage

- timer
  - init_timer();
  - mod_timer();
  - del_timer();
  - schedule_timeout(); //sleep util time out
  - ....

Linux Kernel Module

- Simple test module
  - test_module_init(void)
  - {
    - prink(“test modules init\n”);
  - }
  - test_module_exit (void)
  - {
    - prink(“test modules init\n”);
  - }
  - module_init(test_module_init);
  - module_exit(test_module_exit);

Thanks

Mail:Zhi.zhou@amlogic.com
msn:rising_o@msn.com
Skype:rising_o