Linux Kernel Debugging Tools & Techniques

Friday, December 20, 2024 10:33 AM

1. Trace_printk
2. Dynamic printk
3. Debugfs
4. Trace-cmd, Ftrace, LTTng, TraceCompass
5. Kernel oops messages
6. Kprobe, kretprobe
7. Mem debug:
1. KASAN
2. KMEMSAN
3. Memleak
8. Kdump, kexec, crash
9. Kgdb

Kunit ---> Kernel unit test cases.

Configs

- ll arch/x86/configs
- Make help; Make localmodconfig
- Make menuconfig
    - General Setup --> kernel .config support
    - Kernel hacking

Kernel_debug setup

18a19
> CONFIG_CONSTRUCTORS=y
29c30
< CONFIG_LOCALVERSION=""

> CONFIG_LOCALVERSION="-Prithvi_Local_mod_config"
248c249
< # CONFIG_EXPERT is not set

> CONFIG_EXPERT=y
279a281
> # CONFIG_DEBUG_RSEQ is not set
282c284
< CONFIG_GUEST_PERF_EVENTS=y

```
---
> # CONFIG_PC104 is not set
308a311
> CONFIG_GENERIC_CSUM=y
316a320
> CONFIG_KASAN_SHADOW_OFFSET=0xdfffffc0000000000
379a384
> # CONFIG_PROCESSOR_SELECT is not set
530a536
> # CONFIG_SUSPEND_SKIP_SYNC is not set
544a551
> # CONFIG_DPM_WATCHDOG is not set
586a594
> # CONFIG_ACPI_REDUCED_HARDWARE_ONLY is not set
664a673,674
> # CONFIG_PCI_CNB20LE_QUIRK is not set
> # CONFIG_ISA_BUS is not set
680,706c690
< CONFIG_HAVE_KVM_PFNCACHE=y
< CONFIG_HAVE_KVM_IRQCHIP=y
< CONFIG_HAVE_KVM_IRQFD=y
< CONFIG_HAVE_KVM_IRQ_ROUTING=y
< CONFIG_HAVE_KVM_DIRTY_RING=y
< CONFIG_HAVE_KVM_DIRTY_RING_TSO=y
< CONFIG_HAVE_KVM_DIRTY_RING_ACQ_REL=y
< CONFIG_HAVE_KVM_EVENTFD=y
< CONFIG_KVM_MMIO=y
< CONFIG_KVM_ASYNC_PF=y
< CONFIG_HAVE_KVM_MSI=y
< CONFIG_HAVE_KVM_CPU_RELAX_INTERCEPT=y
< CONFIG_KVM_VFIO=y
< CONFIG_KVM_GENERIC_DIRTYLOG_READ_PROTECT=y
< CONFIG_KVM_COMPAT=y
< CONFIG_HAVE_KVM_IRQ_BYPASS=y
< CONFIG_HAVE_KVM_NO_POLL=y
< CONFIG_KVM_XFER_TO_GUEST_WORK=y
< CONFIG_HAVE_KVM_PM_NOTIFIER=y
< CONFIG_KVM_GENERIC_HARDWARE_ENABLING=y
< CONFIG_VIRTUALIZATION=y
< CONFIG_KVM=m
< # CONFIG_KVM_INTEL is not set
< CONFIG_KVM_AMD=m
< CONFIG_KVM_AMD_SEV=y
< CONFIG_KVM_SMM=y
< # CONFIG_KVM_XEN is not set
---
> # CONFIG_VIRTUALIZATION is not set
737d720
< CONFIG_USER_RETURN_NOTIFIER=y
899a883
> # CONFIG_TRIM_UNUSED_KSYMS is not set
962d945
< CONFIG_PREEMPT_NOTIFIERS=y
```

```
1020a1004
> # CONFIG_SLUB_TINY is not set
1085a1070
> CONFIG_ARCH_HAS_ZONE_DMA_SET=y
1619c1604
< CONFIG_RFKILL_INPUT=y
---
> # CONFIG_RFKILL_INPUT is not set
1671a1657,1661
> # CONFIG_PCIE_BUS_TUNE_OFF is not set
> CONFIG_PCIE_BUS_DEFAULT=y
> # CONFIG_PCIE_BUS_SAFE is not set
> # CONFIG_PCIE_BUS_PERFORMANCE is not set
> # CONFIG_PCIE_BUS_PEER2PEER is not set
2778a2769
> # CONFIG_TTY_PRINTK is not set
3017a3009
> # CONFIG_GPIO_SYSFS is not set
3554a3547,3548
> # CONFIG_DRM_DEBUG_DP_MST_TOPOLOGY_REFS is not set
> CONFIG_DRM_DEBUG_MODESET_LOCK=y
3556a3551
> # CONFIG_DRM_FBDEV_LEAK_PHYS_SMEM is not set
4056a4052
> # CONFIG_USB_OTG_DISABLE_EXTERNAL_HUB is not set
4423d4418
< CONFIG_IRQ_BYPASS_MANAGER=m
5452a5448
> # CONFIG_FORCE_NR_CPUS is not set
5478a5475
> CONFIG_STACKDEPOT_ALWAYS_INIT=y
5513,5514c5510,5511
< # CONFIG_DEBUG_INFO_DWARF_TOOLCHAIN_DEFAULT is not set
< CONFIG_DEBUG_INFO_DWARF4=y
---
> CONFIG_DEBUG_INFO_DWARF_TOOLCHAIN_DEFAULT=y
> # CONFIG_DEBUG_INFO_DWARF4 is not set
5520,5521c5517,5518
< CONFIG_DEBUG_INFO_BTF=y
< # CONFIG_GDB_SCRIPTS is not set
---
> # CONFIG_DEBUG_INFO_BTF is not set
> CONFIG_GDB_SCRIPTS=y
5527a5525
> # CONFIG_DEBUG_FORCE_FUNCTION_ALIGN_64B is not set
5528a5527
> # CONFIG_VMLINUX_MAP is not set
5556c5555,5566
< # CONFIG_UBSAN is not set
---
> CONFIG_UBSAN=y
> # CONFIG_UBSAN_TRAP is not set
> CONFIG_CC_HAS_UBSAN_BOUNDS_STRICT=y
```

```
> CONFIG_UBSAN_BOUNDS=y
> CONFIG_UBSAN_BOUNDS_STRICT=y
> CONFIG_UBSAN_SHIFT=y
> # CONFIG_UBSAN_DIV_ZERO is not set
> CONFIG_UBSAN_BOOL=y
> CONFIG_UBSAN_ENUM=y
> # CONFIG_UBSAN_ALIGNMENT is not set
> CONFIG_UBSAN_SANITIZE_ALL=y
> # CONFIG_TEST_UBSAN is not set
5585c5595,5598
< # CONFIG_DEBUG_KMEMLEAK is not set
---
> CONFIG_DEBUG_KMEMLEAK=y
> CONFIG_DEBUG_KMEMLEAK_MEM_POOL_SIZE=16000
> CONFIG_DEBUG_KMEMLEAK_DEFAULT_OFF=y
> CONFIG_DEBUG_KMEMLEAK_AUTO_SCAN=y
5596c5609
< CONFIG_DEBUG_MEMORY_INIT=y
---
> # CONFIG_DEBUG_MEMORY_INIT is not set
5602c5615,5621
< # CONFIG_KASAN is not set
---
> CONFIG_KASAN=y
> CONFIG_KASAN_GENERIC=y
> CONFIG_KASAN_OUTLINE=y
> # CONFIG_KASAN_INLINE is not set
> CONFIG_KASAN_STACK=y
> CONFIG_KASAN_VMALLOC=y
> CONFIG_KASAN_MODULE_TEST=m
5604c5623,5628
< # CONFIG_KFENCE is not set
---
> CONFIG_KFENCE=y
> CONFIG_KFENCE_SAMPLE_INTERVAL=100
> CONFIG_KFENCE_NUM_OBJECTS=255
> # CONFIG_KFENCE_DEFERRABLE is not set
> # CONFIG_KFENCE_STATIC_KEYS is not set
> CONFIG_KFENCE_STRESS_TEST_FAULTS=0
5651c5675,5676
< # CONFIG_PROVE_LOCKING is not set
---
> CONFIG_PROVE_LOCKING=y
> # CONFIG_PROVE_RAW_LOCK_NESTING is not set
5653,5659c5678,5691
< # CONFIG_DEBUG_RT_MUTEXES is not set
< # CONFIG_DEBUG_SPINLOCK is not set
< # CONFIG_DEBUG_MUTEXES is not set
< # CONFIG_DEBUG_WW_MUTEX_SLOWPATH is not set
< # CONFIG_DEBUG_RWSEMS is not set
< # CONFIG_DEBUG_LOCK_ALLOC is not set
< # CONFIG_DEBUG_ATOMIC_SLEEP is not set
---
```

> CONFIG_DEBUG_RT_MUTEXES=y
> CONFIG_DEBUG_SPINLOCK=y
> CONFIG_DEBUG_MUTEXES=y
> CONFIG_DEBUG_WW_MUTEX_SLOWPATH=y
> CONFIG_DEBUG_RWSEMS=y
> CONFIG_DEBUG_LOCK_ALLOC=y
> CONFIG_LOCKDEP=y
> CONFIG_LOCKDEP_BITS=15
> CONFIG_LOCKDEP_CHAINS_BITS=16
> CONFIG_LOCKDEP_STACK_TRACE_BITS=19
> CONFIG_LOCKDEP_STACK_TRACE_HASH_BITS=14
> CONFIG_LOCKDEP_CIRCULAR_QUEUE_BITS=12
> # CONFIG_DEBUG_LOCKDEP is not set
> CONFIG_DEBUG_ATOMIC_SLEEP=y
5666a5699,5700
> CONFIG_TRACE_IRQFLAGS=y
> CONFIG_TRACE_IRQFLAGS_NMI=y
5688a5723
> CONFIG_PROVE_RCU=y
5727a5763
> CONFIG_PREEMPTIRQ_TRACEPOINTS=y
5756d5791
< CONFIG_PROBE_EVENTS_BTF_ARGS=y

Instrumenting kernel

Pr_fmt() should be the first line of code in the source code file

For device drivers
Use dev_foo()
Journalctl
Ps -ef | grep systemd-jou

Systemd is the new init
Systemd-journalctl archives the ring buffer logs to drive storage.

cat /proc/sys/kernel/printk

Proc file system does not reside on disk.

Console device

Ignore loglevel [knl]

Changing kernel command line.

Mandatory system components
Bootloader
Kernel
Root file system
Dtb --> (embedded systems only)

Docs.kernel.org : --> All options of kernel command line are described.

Convenient.h
Dump_stack()

Ccflags-y += -Og

Hardirq --> invokes hardirq_handler() --> **Top half**

Bottom half --> implemented soft_irq_handler ()

Rate Limiting : kernel algorithm to limit emitted printks from device drivers.

For e.g. **pr_debug_ratelimited**()

Proc/sys/kernel/printk_ratelimit 5
Proc/sys/kernel/printk_ratelimit_burst 10

Openembedded.org

Printk not available on early kernel

For that there is another early kernel printf.

CONFIG_DEBUG_LL

Early_printk.c implements early printk --> **early_serial_write**()

Dynamic Debug
Zcat /proc/config.gz | grep DYNAMIC

Which trace printk's are off or on
 wc -l /proc/dynamic_debug/control
grep "=p" /proc/dynamic_debug/control | wc -l

Echo "file sound/pci/intel8x0.c +p" > /sys/kernel/debug/dynamic_debug/control

In case of debugging when printk messages fail to reach hard drive.

__Log_buf is the symbol of the buffer.

grep __log_buf

Debugfs : Allows userspace code to print out critical kernel space structures

Linux kernel debugging part 2 (ebook)

__Kernel communication pathways__

- **Without writing c code**
- ~~Proc file system~~ -- After Kernel v2.6 driver creators cannot use.
- Sysfs file system
    - Limitation can use only **one entry per sysfs file**
- **Debugfs file system**
    - **No limitations**
- **Writing C code**
- Netlink socket
- Ioctl

Cat for read
Echo for write.

Unix/Linux philosophy.
Every thing is a process. If it is not a process then it is a **file**