Linux内核最新进展

海航云资深架构师张健

Outline

- Who I am
- Live with upstream kernel
- Who writes the Linux kernel
- Architecture relative features
- Memory features
- Staging features: ILP32

Who I am?

- Linux kernel developer
- Distributed storage developer
- Open source lover
- Wechat public account: 敏达生活

Live with upstream kernel

- Why
  - Minimize migration effort
    - E.g. migrate from kernel 4.4(16.04) to kernel 4.15(18.04) after three years LTS.
  - Collaborate with community
- How
  - Mailing list
  - IRC(e.g. irccloud)
  - Conference
- What
  - Feature
  - Pull request
  - Commit message
  - Code

Live with upstream kernel

- kernel newbie: https://kernelnewbies.org/Linux_4.16
- LWN: merge window
  - 4.16 Part1: https://lwn.net/Articles/746129/
  - 4.16 Part2: https://lwn.net/Articles/746791/
  - 4.17 Part1: https://lwn.net/Articles/750928/
  - 4.17 Part2: https://lwn.net/Articles/751482/

Who writes the Linux kernel?

No.1	Hobbyists	7558417(11.53%)
No.2	Intel	6159531(9.39%)
No.3	Red Hat	6074666(9.26%)
No.4	Unknown	5775254(8.81%)
No.5	Novell	3229108(4.92%)
No.6	AMD	2620724(4.00%)
No.7	IBM	2601454(3.97%)
No.8	Broadcom	1848568(2.82%)
No.9	Linaro	1407847(2.15%)
No.10	Samsung	1370381(2.09%)

Who writes the Linux kernel

ref: http://www.remword.com/kps_result/all_whole_line.htm

Most active 4.16 employers

By changesets			By lines changed		
Intel	1424	10.6%	AMD	97644	14.2%
Red Hat	971	7.2%	Intel	73566	10.7%
(Unknown)	962	7.2%	(Unknown)	33700	4.9%
(None)	895	6.7%	Red Hat	33027	4.8%
AMD	677	5.0%	(None)	31155	4.5%
IBM	566	4.2%	IBM	26329	3.8%
Linaro	524	3.9%	Linaro	25245	3.7%
Renesas Electronics	373	2.8%	(Consultant)	20772	3.0%
Mellanox	366	2.7%	Cavium	18173	2.6%
Google	365	2.7%	Samsung	16587	2.4%
SUSE	337	2.5%	ARM	16368	2.4%
(Consultant)	333	2.5%	Broadcom	13868	2.0%
ARM	328	2.4%	Texas Instruments	13597	2.0%
Oracle	320	2.4%	Code Aurora Forum	13437	2.0%
Huawei Technologies	295	2.2%	Oracle	13335	1.9%
Samsung	272	2.0%	Bootlin	13038	1.9%
Texas Instruments	233	1.7%	Mellanox	12999	1.9%
Broadcom	201	1.5%	Google	12281	1.8%
Netronome Systems	192	1.4%	Huawei Technologies	11781	1.7%

ref:
https://lwn.net/Articles/750054/

Who writes the Linux kernel

Meltdown and Spectre

- ISA and side-channel attack
  - Ref: google I/O 1'31'': https://mp.weixin.qq.com/s/bUVQVk3W6BM4WNNbFnJ4HQ
- Upstream status
  - Addressing Meltdown and Spectre in the kernel: https://lwn.net/Articles/743265/
  - Meltdown/Spectre mitigation for 4.15 and beyond: https://lwn.net/Articles/744287/
  - Meltdown and Spectre mitigations — a February update: https://lwn.net/Articles/746551/
- Variant4: https://bugs.chromium.org/p/project-zero/issues/detail?id=1528&from=timeline
  - From google project zero as well

Spectre vulnerability(variant 1): "bounds-check bypass"

```
BamvordeMacBook-Pro:linux bamvor$ git log --oneline --grep array_index.*nospec
56986016cb8c powerpc/64s: Wire up cpu_show_spectre_v1()
85a2d939c059 Merge branch 'x86-pti-for-linus' of git://git.kernel.org/pub/scm/linux/kernel/git/tip/tip
eb6174f6d1be nospec: Include <asm/barrier.h> dependency
b98c6a160a05 nospec: Allow index argument to have const-qualified type
1d91c1d2c80c nospec: Kill array_index_nospec_mask_check()
d4667ca14261 Merge branch 'x86-pti-for-linus' of git://git.kernel.org/pub/scm/linux/kernel/git/tip/tip
8fa80c503b48 nospec: Move array_index_nospec() parameter checking into separate macro
be3233fbfcb8 x86/speculation: Fix up array_index_nospec_mask() asm constraint
dff839f27dc8 Merge branch 'for-linus' of git://git.kernel.org/pub/scm/linux/kernel/git/s390/linux
c0136321924d Merge tag 'arm64-upstream' of git://git.kernel.org/pub/scm/linux/kernel/git/arm64/linux
6314d90e6493 arm64: entry: Ensure branch through syscall table is bounded under speculation
022620eed3d0 arm64: Implement array_index_mask_nospec()
e2dd833389cc s390: add optimized array_index_mask_nospec
085331dfc6bb x86/kvm: Update spectre-v1 mitigation
edfbae53dab8 x86/spectre: Report get_user mitigation for spectre_v1
259d8c1e9843 nl80211: Sanitize array index in parse_txq_params
2fbd7af5af86 x86/syscall: Sanitize syscall table de-references under speculation
c7f631cb07e7 x86/get_user: Use pointer masking to limit speculation
b3bbfb3fb5d2 x86: Introduce __uaccess_begin_nospec() and uaccess_try_nospec
babdde2698d4 x86: Implement array_index_mask_nospec
f3804203306e array_index_nospec: Sanitize speculative array de-references
f84a56f73ddd Documentation: Document array_index_nospec
```

Spectre vulnerability(variant 1): "bounds-check bypass" cont

```
if (within_bounds(index)) {
    value = array[index];
    if (some_function_of(value))
        execute_externally_visible_action();
}
```

git show babdde2698d4

SBB: Integer Subtraction with Borrow. ref: https://c9x.me/x86/html/file_module_x86_id_286.html

```
diff --git a/arch/x86/include/asm/barrier.h b/arch/x86/include/asm/barrier.h
index 7fb336210e1b..173b38f5fe88 100644
--- a/arch/x86/include/asm/barrier.h
+++ b/arch/x86/include/asm/barrier.h
@@ -24,6 +24,30 @@
 #define wmb()  asm volatile("sfence" ::: "memory")
 #endif

+/**
+ * array_index_mask_nospec() - generate a mask that is ~0UL when the
+ *     bounds check succeeds and 0 otherwise
+ * @index: array element index
+ * @size: number of elements in array
+ *
+ * Returns:
+ *     0 - (index < size)
+ */
+static inline unsigned long array_index_mask_nospec(unsigned long index,
+               unsigned long size)
+{
+       unsigned long mask;
+
+       asm ("cmp %1,%2; sbb %0,%0;"
+                       :"=r" (mask)
+                       :"r"(size),"r" (index)
+                       :"cc");
+       return mask;
+}
+
+/* Override the default implementation from linux/nospec.h. */
+#define array_index_mask_nospec array_index_mask_nospec
+
```

Spectre vulnerability(variant 1): “bounds-check bypass” cont(arm64)

git show 022620eed3d0

Arm armv8 C6-661:
SBC: Subtract with Carry subtracts a register vaue and the value of NOT (Carry flag) from a register value, and writes the result to the destination register

```
diff --git a/arch/arm64/include/asm/barrier.h b/arch/arm64/include/asm/barrier.h
index c0a846d2c602..f11518af96a9 100644
--- a/arch/arm64/include/asm/barrier.h
+++ b/arch/arm64/include/asm/barrier.h
@@ -41,6 +41,27 @@
 #define dma_rmb()      dmb(oshld)
 #define dma_wmb()      dmb(oshst)

+/*
+ * Generate a mask for array_index__nospec() that is ~0UL when 0 <= idx < sz
+ * and 0 otherwise.
+ */
+#define array_index_mask_nospec array_index_mask_nospec
+static inline unsigned long array_index_mask_nospec(unsigned long idx,
+                                                    unsigned long sz)
+{
+       unsigned long mask;
+
+       asm volatile(
+       "       cmp     %1, %2\n"
+       "       sbc     %0, xzr, xzr\n"
+       : "=r" (mask)
+       : "r" (idx), "Ir" (sz)
+       : "cc");
+
+       csdb();
+       return mask;
+}
+
```

Spectre vulnerability(variant 1): “bounds-check bypass” cont(arm64)

https://patchwork.kernel.org/patch/10200747/

“For CPUs capable of data value prediction, CSDB waits for any outstanding predictions to architecturally resolve before allowing speculative execution to continue. Provide macros to expose it to the arch code.”

```
diff --git a/arch/arm64/include/asm/assembler.h b/arch/arm64/include/asm/assembler.h
index 794fe8122602..709adde9e80f 100644
--- a/arch/arm64/include/asm/assembler.h
+++ b/arch/arm64/include/asm/assembler.h
@@ -116,6 +116,13 @@
        .endm

 /*
+ * Value prediction barrier
+ */
+       .macro  csdb
+       hint    #20
+       .endm
+
+/*
  * NOP sequence
  */
        .macro  nops, num
diff --git a/arch/arm64/include/asm/barrier.h b/arch/arm64/include/asm/barrier.h
index 77651c49ef44..c0a846d2c602 100644
--- a/arch/arm64/include/asm/barrier.h
+++ b/arch/arm64/include/asm/barrier.h
@@ -32,6 +32,7 @@
 #define dsb(opt)        asm volatile("dsb " #opt : : : "memory")

 #define psb_csync()     asm volatile("hint #17" : : : "memory")
+#define csdb()          asm volatile("hint #20" : : : "memory")

 #define mb()            dsb(sy)
 #define rmb()           dsb(ld)
```

SCMI: ARM System Control and Management Interface

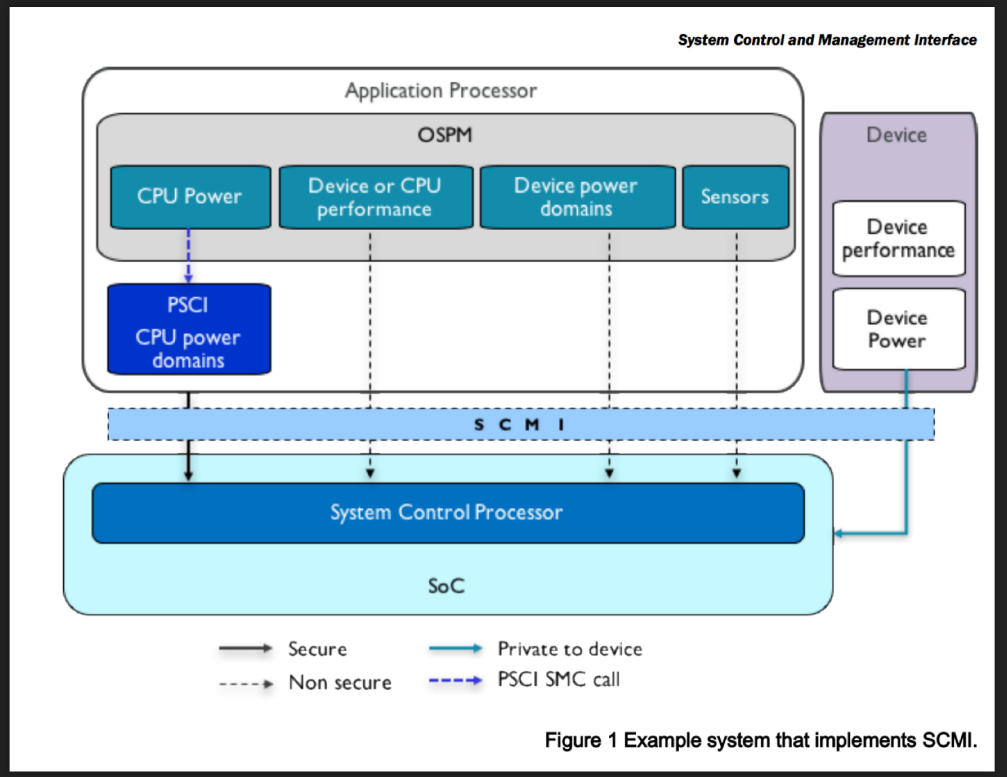
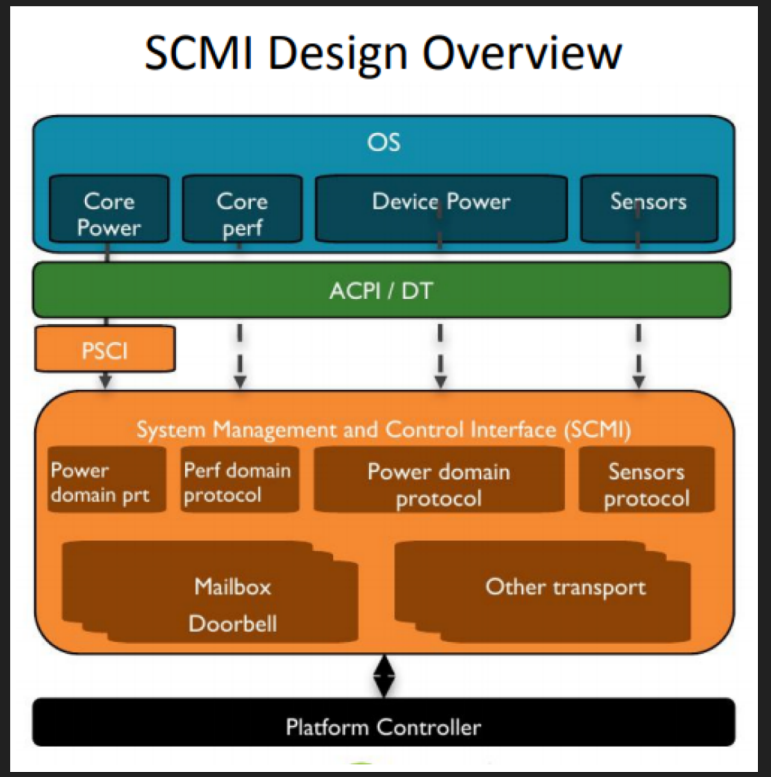


Figure 1 Example system that implements SCMI.



Ref:

http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.den0056a/index.html

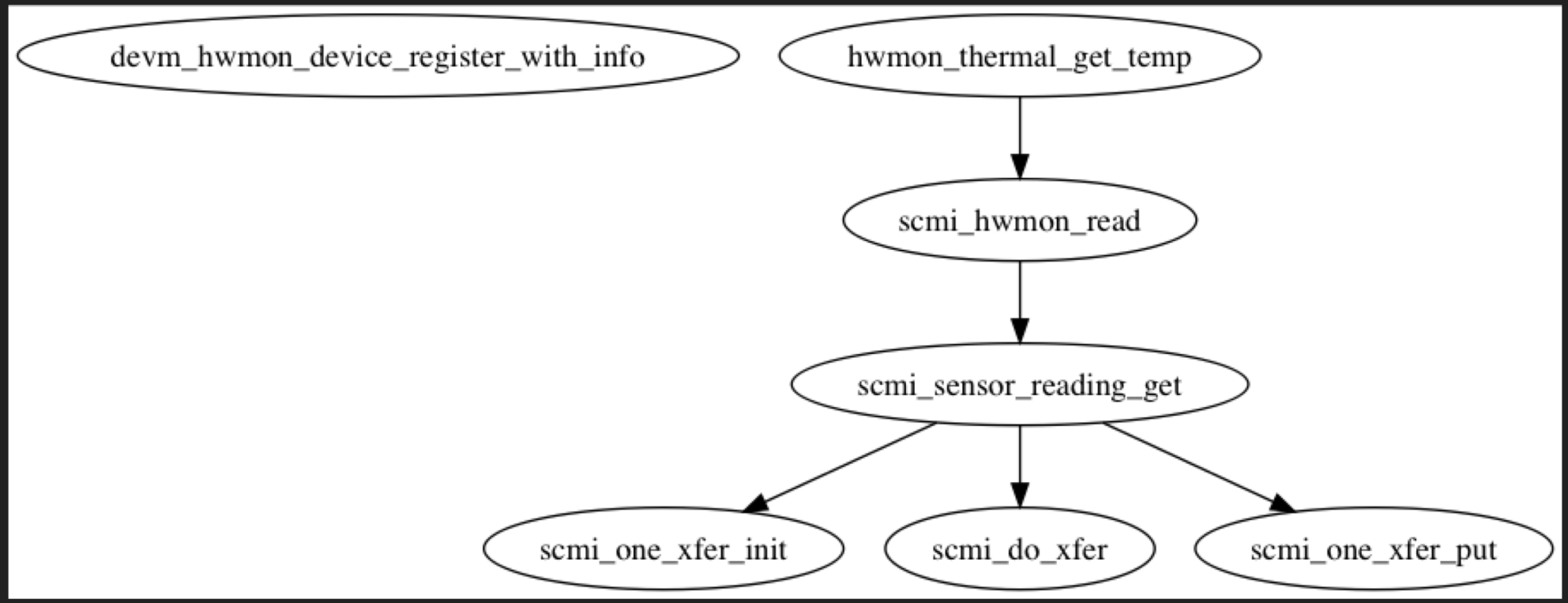
https://www.linuxplumbersconf.org/2017/ocw/system/presentations/4759/original/fast_dvfs_scmi.pdf

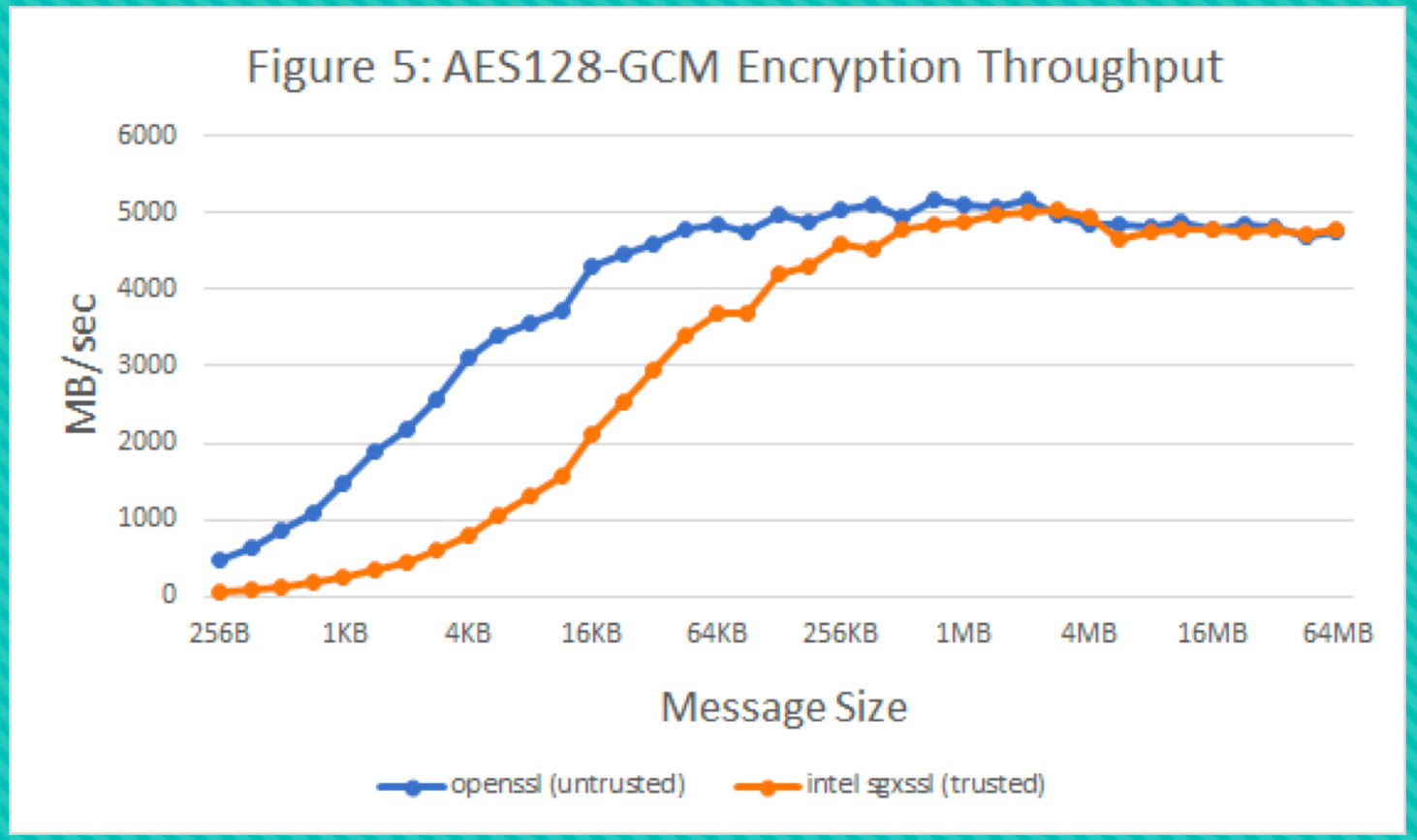
SCMI: ARM System Control and Management Interface

```
* tag 'scmi-updates-4.17' of ssh://gitolite.kernel.org/pub/scm/linux/kernel/git/sudeep.holla/linux:
  cpufreq: scmi: add support for fast frequency switching
  cpufreq: add support for CPU DVFS based on SCMI message protocol
  hwmon: add support for sensors exported via ARM SCMI
  hwmon: (core) Add hwmon_max to hwmon_sensor_types enumeration
  clk: add support for clocks provided by SCMI
  firmware: arm_scmi: add device power domain support using genpd
  firmware: arm_scmi: add per-protocol channels support using idr objects
  firmware: arm_scmi: refactor in preparation to support per-protocol channels
  firmware: arm_scmi: add option for polling based performance domain operations
  firmware: arm_scmi: add support for polling based SCMI transfers
  firmware: arm_scmi: probe and initialise all the supported protocols
  firmware: arm_scmi: add initial support for sensor protocol
  firmware: arm_scmi: add initial support for power protocol
  firmware: arm_scmi: add initial support for clock protocol
  firmware: arm_scmi: add initial support for performance protocol
  firmware: arm_scmi: add scmi protocol bus to enumerate protocol devices
  firmware: arm_scmi: add common infrastructure and support for base protocol
  firmware: arm_scmi: add basic driver infrastructure for SCMI
  dt-bindings: arm: add support for ARM System Control and Management Interface(SCMI) protocol
  dt-bindings: mailbox: add support for mailbox client shared memory
```

Ref:
git log f46f11dc1e86270935041fbc3920ba71a050a5fd^!

SCMI: ARM System Control and Management Interface





Memory encryption: Intel SGX

Ref: https://medium.com/@danny_harnik/impressions-of-intel-sgx-performance-22442093595a

Memory encryption: AMD SEV

- Ref: http://amd-dev.wpengine.netdna-cdn.com/wordpress/media/2013/12/AMD_Memory_Encryption_Whitepaper_v7-Public.pdf

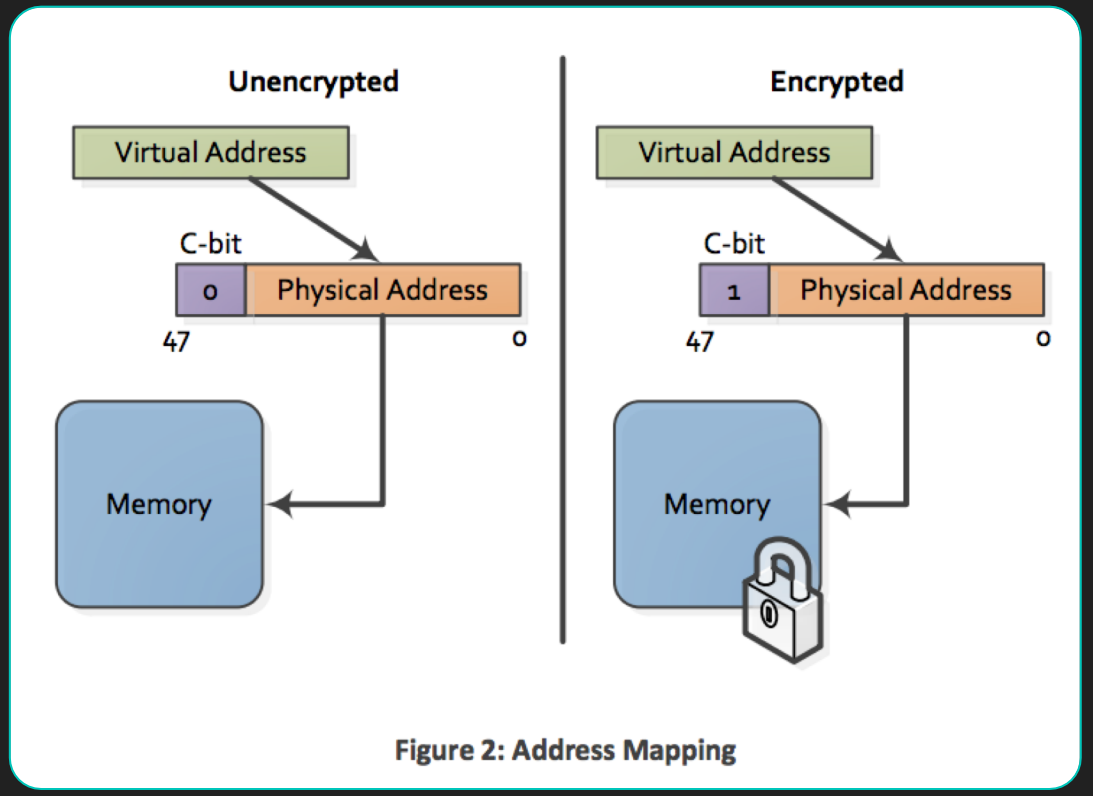


Figure 2: Address Mapping

```
@@ -1365,6 +1369,13 @@ unsigned long do_mmap(struct file *file, unsigned long addr,
        if (offset_in_page(addr))
                return addr;

+       if (flags & MAP_FIXED_NOREPLACE) {
+               struct vm_area_struct *vma = find_vma(mm, addr);
+
+               if (vma && vma->vm_start <= addr)
+                       return -EEXIST;
+       }
+
        if (prot == PROT_EXEC) {
                pkey = execute_only_pkey(mm);
                if (pkey < 0)
```

Memory: Introduce MAP_FIXED_NOREPLACE

- git log --oneline --grep MAP_FIXED_NOREPLACE
- Commit: a4ff8e8620d3 mm: introduce MAP_FIXED_NOREPLACE

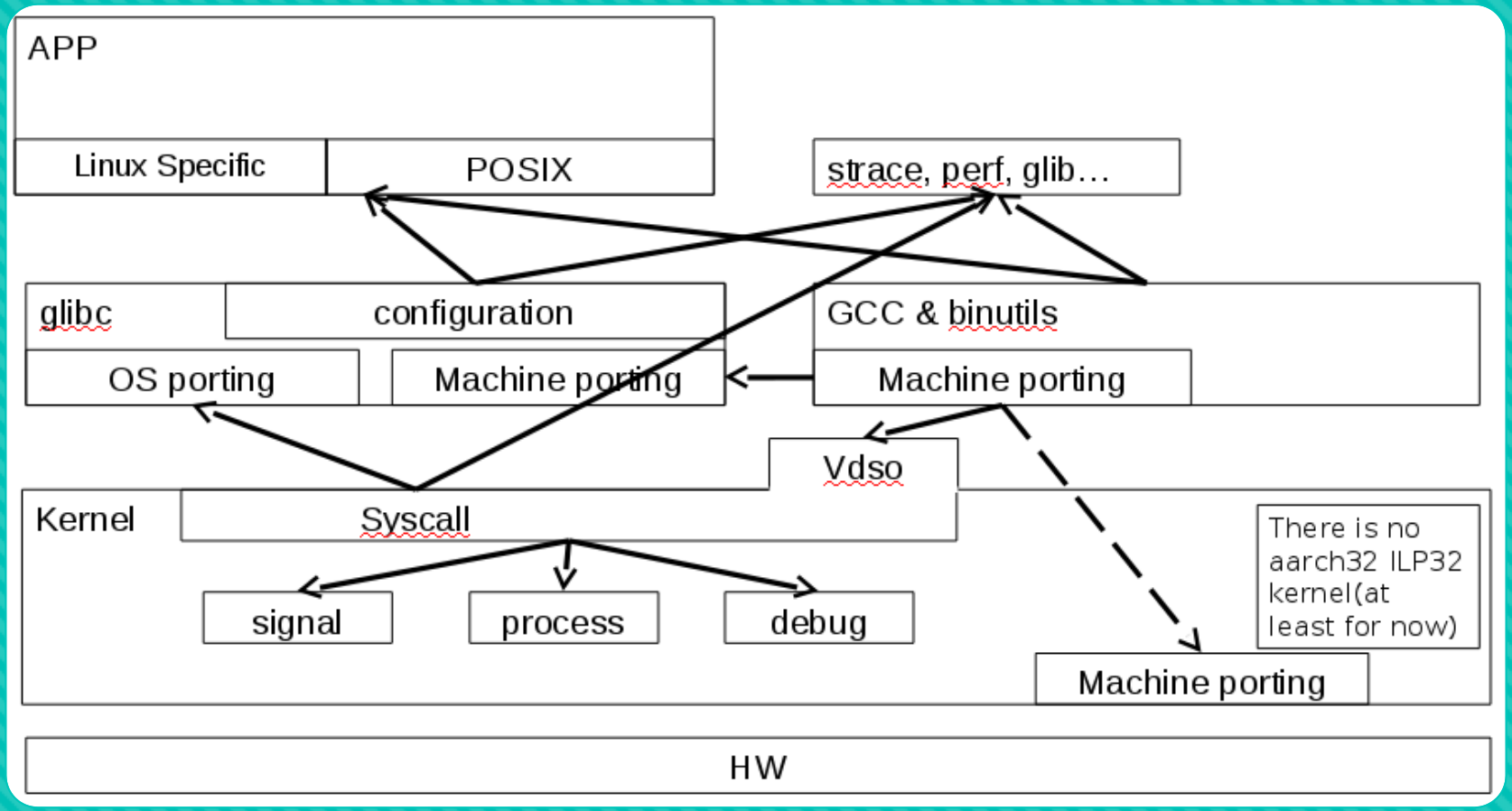
Staging features: ILP32

```
88c5962bd33e (yury/ilp32-4.16) arm64: ilp32: Make the Kconfig option default y
2a3c10a41c44 arm64:ilp32: add ARM64_ILP32 to Kconfig
48a17244fe5d arm64:ilp32: add vdso-ilp32 and use for signal return
2b7dc0f6757e arm64: ptrace: handle ptrace_request differently for aarch32 and ilp32
cf4bc5d1e683 arm64: ilp32: introduce ilp32-specific sigframe and ucontext
b812dab0b44a arm64: signal32: move ilp32 and aarch32 common code to separated file
a6c4e745df66 arm64: signal: share lp64 signal structures and routines to ilp32
b7f787f07098 arm64: ilp32: add sys_ilp32.c and a separate table (in entry.S) to use it
aa0ab84b0b73 arm64: ilp32: share aarch32 syscall handlers
69c86cf3bb1f arm64: ilp32: introduce binfmt_ilp32.c
227d777e6f53 arm64: change compat_elf_hwcap and compat_elf_hwcap2 prefix to a32
d9be9fce0063 arm64: introduce binfmt_elf32.c
f5446bbe63eb arm64: ilp32: add is_ilp32_compat_{task,thread} and TIF_32BIT_AARCH64
9004838923c3 arm64: introduce is_a32_task and is_a32_thread (for AArch32 compat)
0458b36fb77f arm64: uapi: set __BITS_PER_LONG correctly for ILP32 and LP64
f00e69817d12 arm64: rename functions that reference compat term
0b99ff727d47 arm64: rename COMPAT to AARCH32_EL0 in Kconfig
a043e409d947 arm64: ilp32: add documentation on the ILP32 ABI for ARM64
a5a574c17606 thread: move thread bits accessors to separated file
29a5b16a8111 asm-generic: Drop getrlimit and setrlimit syscalls from default list
d28a53977821 32-bit userspace ABI: introduce ARCH_32BIT_OFF_T config option
e2da3e9f6fea compat ABI: use non-compat openat and open_by_handle_at variants
c98d3dba8f8a ptrace: Add compat PTRACE_{G,S}ETSIGMASK handlers
7c3403693e46 arm64: signal: Make parse_user_sigframe() independent of rt_sigframe layout
0adb32858b0b (tag: v4.16) Linux 4.16
```

Ref: https://github.com/norov/linux/tree/ilp32-4.16

	ILP32	LP64	LLP64	ILP64
char	8	8	8	8
short	16	16	16	16
int	32	32	32	64
long	32	64	32	64
long long	64	64	64	64
size_t	32	64	64	64
pointer	32	64	64	64
	arm aarch32 aarch64 ILP32 x32 n32	aarch64 LP64	64bit windows	

Staging features: ILP32



Staging features: ILP32

Staging features: ILP32

```
commit 48a17244fe5d663d7b9138175b066b90edd9de46
Author: Philipp Tomsich <philipp.tomsich@theobroma-systems.com>
Date:   Tue May 24 03:04:51 2016 +0300

    arm64:ilp32: add vdso-ilp32 and use for signal return

    ILP32 VDSO exports following symbols:
     __kernel_rt_sigreturn;
     __kernel_gettimeofday;
     __kernel_clock_gettime;
     __kernel_clock_getres.

    What shared object to use, kernel selects depending on result of
    is_ilp32_compat_task() in arch/arm64/kernel/vdso.c, so it substitutes
    correct pages and spec.

    Adjusted to move the data page before code pages in sync with
    commit 601255ae3c98 ("arm64: vdso: move data page before code pages")

    Signed-off-by: Philipp Tomsich <philipp.tomsich@theobroma-systems.com>
    Signed-off-by: Christoph Muellner <christoph.muellner@theobroma-systems.com>
    Signed-off-by: Yury Norov <ynorov@caviumnetworks.com>
    Signed-off-by: Bamvor Jian Zhang <bamv2005@gmail.com>
```

Thanks

- 半瓦平时有随手记笔记的习惯，公众号《敏达生活》。只分享自己有体会的信息，希望能促进价值信息流动。任何建议欢迎给我留言或添加我的微信：
- 春风吹又生—-梳理中国CPU
- Linux自动化部署工具综述（Linux自动化部署工具系列之一）
- 比较操作系统镜像制作方式（Linux自动化部署工具系列之二）
- ARM生态系统的盛会Linaro connect（之一）：arm64 server和端侧AI
- ARM生态系统的盛会Linaro connect（之二）：arm64 workstation和低成本调试工具
- 内核测试小整理