ORACLE

Zettalinux

It's not too late to start

Matthew Wilcox
Technical Advisor
Linux Kernel Development
2022-09-14

Safe harbor statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

Copyright © 2022, Oracle and/or its affiliates

Problems We Want To Avoid

1. Large File Summit 2

```
#define _FILE_OFFSET_BITS=128
lloff_t pos;
lllseek()
```

2. `CONFIG_64BIT && CONFIG_HIGHMEM`

Copyright © 2022, Oracle and/or its affiliates

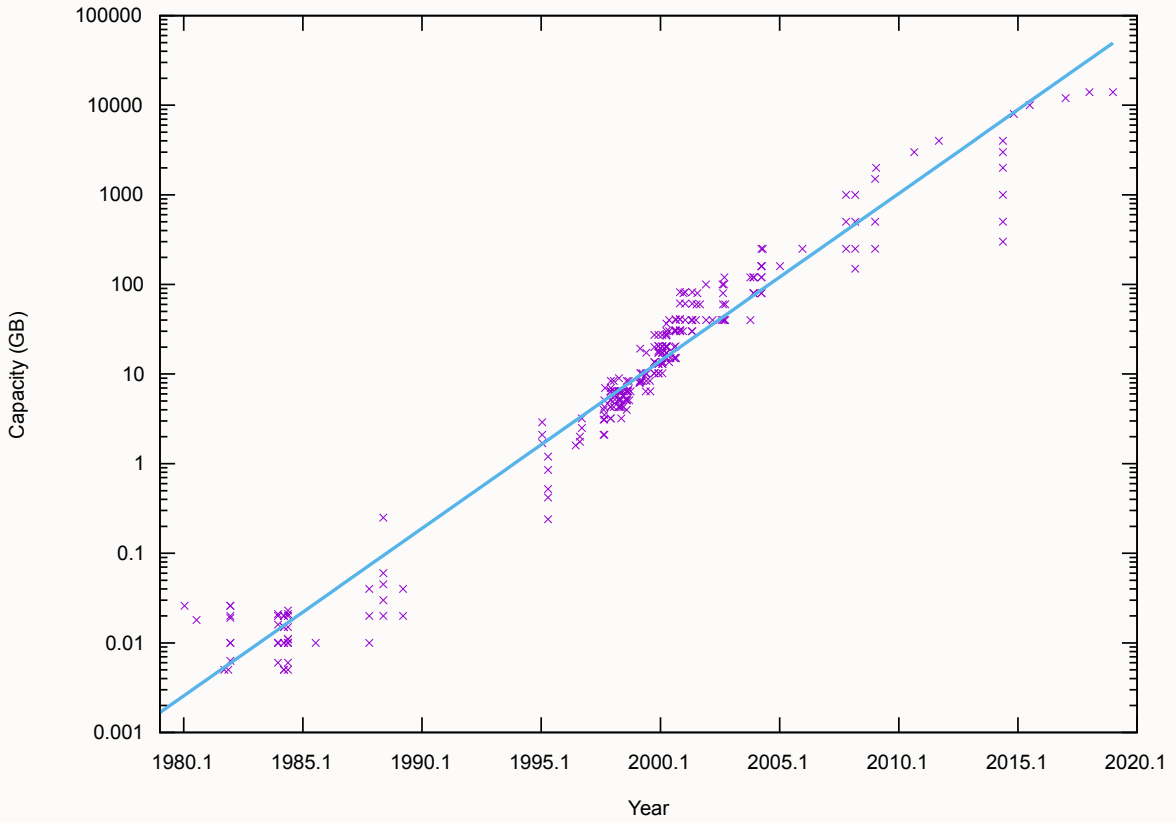
Obvious Solution

- CPUs with 128-bit general-purpose registers

Copyright © 2022, Oracle and/or its affiliates

File sizes

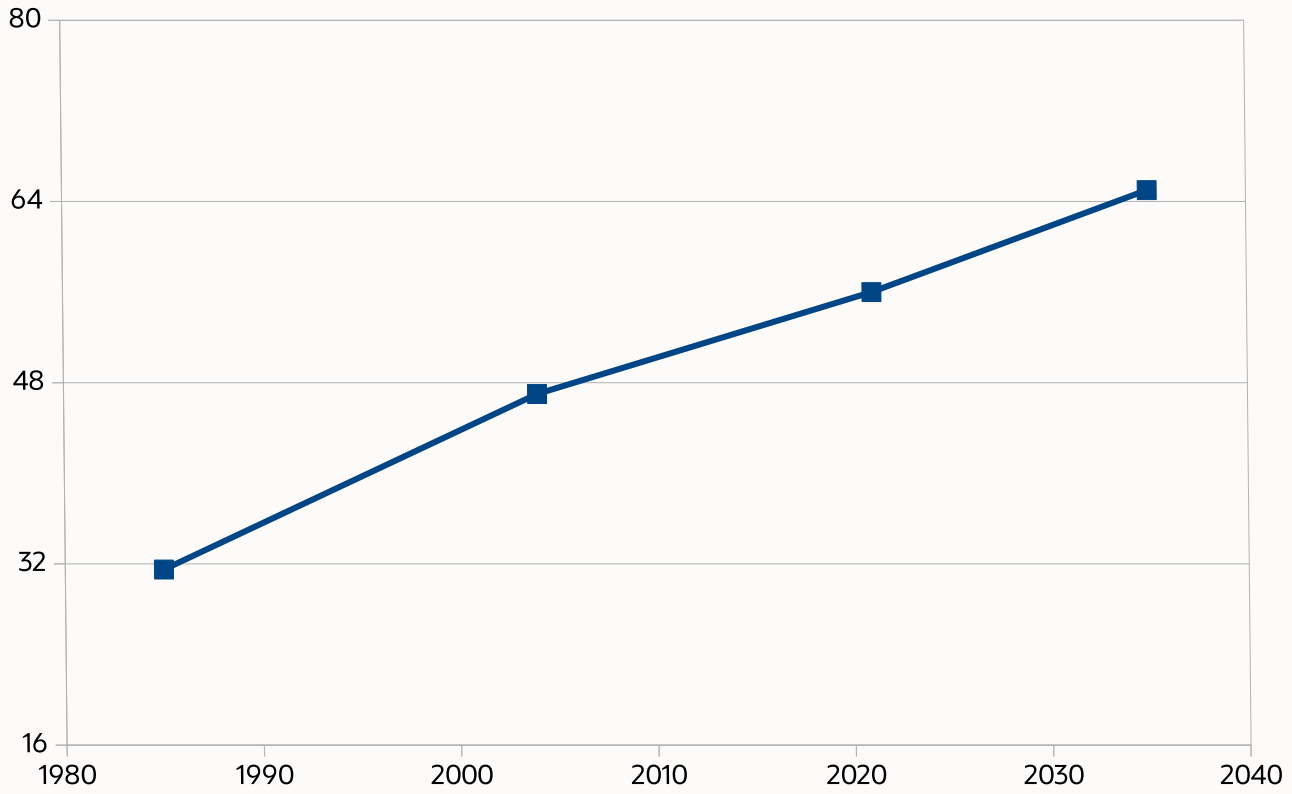
- Customer demand for 8EiB file sizes projected around 2040
- Would like to get OS support in place by 2035
- Very limited arithmetic needed on `off_t`
  - Comparison
  - Addition / subtraction
  - Shift



Copyright © 2022, Oracle and/or its affiliates

User Address Space

- 80386 (1985): 3GiB
- Nocona (2004): 128TiB
- Ice Lake (2021): 64PiB
- Future (2035?): 32EiB



Copyright © 2022, Oracle and/or its affiliates

Pointers

- Address Space Layout Randomization uses address space bits for security
- Using 2MB / 1GB TLB entries reduces ASLR effectiveness – to just 17 bits in the worst case
- ADI/TBI/LAM/MTE/… only useful for debugging, not enough bits for security
- CHERI (Arm/MIPS/RISC-V) uses capability registers to provide better security
  - But does nothing for the impending address space crunch

Copyright © 2022, Oracle and/or its affiliates

In-kernel types

- Decide type sizes
  - `32 bit: ILP32 (int:32, long:32, pointer:32, long long:64)`
  - `64 bit: I32LP64 (int:32, long:64, pointer:64, long long:64)`
  - ~~`128 bit: I32L64P128 (int:32, long:64, pointer:128, long long:128)`~~
  - `128 bit: I32LP128 (int:32, long:128, pointer:128, long long:128)`
- Ask compiler people for a `__int64_t` so we can leave a gap between `int` and `long`
- Encourage switching to Rust types
  - `int` → `i32`
  - `unsigned int` → `u32`
  - `long` → `isize`
  - `unsigned long` → `usize`
  - Use `i64`/`u64` where it makes sense (eg network protocols, hardware registers, storage formats)
- Add a `uaddr_t` for userspace virtual addresses to distinguish them from other `unsigned long`

Copyright © 2022, Oracle and/or its affiliates

UAPI/UABI types

- Some interfaces specified with explicit 64-bit sizes to avoid compat syscalls

```
struct clone_args {
        __aligned_u64 flags;            /* OK */
        __aligned_u64 pidfd;            /* (int *) */
        __aligned_u64 child_tid;        /* (pid_t *) */
        __aligned_u64 parent_tid;       /* (pid_t *) */
        __aligned_u64 exit_signal;      /* OK */
        __aligned_u64 stack;            /* (void *) */
        __aligned_u64 stack_size;       /* OK */
        __aligned_u64 tls;              /* (void *) */
        __aligned_u64 set_tid;          /* (pid_t *) */
        __aligned_u64 set_tid_size;     /* OK */
        __aligned_u64 cgroup;           /* OK */
};
```

- Also DRM & io_uring, probably others
- Suggest a `__ptr_t` for these cases which is somewhat like a `uintptr_t` but is always 32 bytes

Copyright © 2022, Oracle and/or its affiliates

Other Linux concerns

- Only one “compat” personality is supported (ie 32-bit on 64-bit)
- Data structures lovingly laid out by hand
- Alignment of 128-bit types
- `#if BITS_PER_LONG == 64`
- `if (sizeof(unsigned long) == 8)`

Copyright © 2022, Oracle and/or its affiliates

Next Steps

- Leader for this effort
- Qemu RISC-V 128b implementation
- Compiler to match
- Influence CPU companies

Copyright © 2022, Oracle and/or its affiliates

Conclusion

- Avoid Physical Address Extension and Large File Summit hacks by moving to 128-bit registers soon
- CPU companies must start work now, and we need to push them

Copyright © 2022, Oracle and/or its affiliates

ORACLE