

seL4: формальная верификация ядра ОС

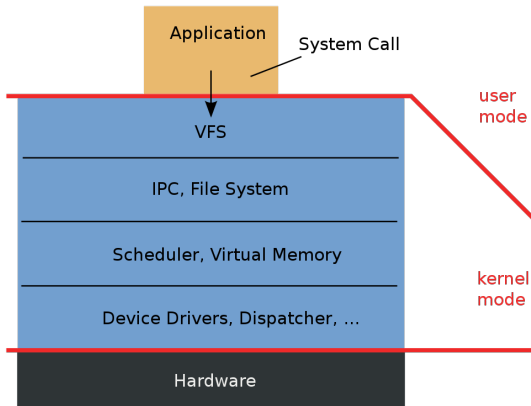
Реализует минимальную функциональность:

- управление памятью
- планирование времени
- доступ к устройствам
- коммуникации процессов

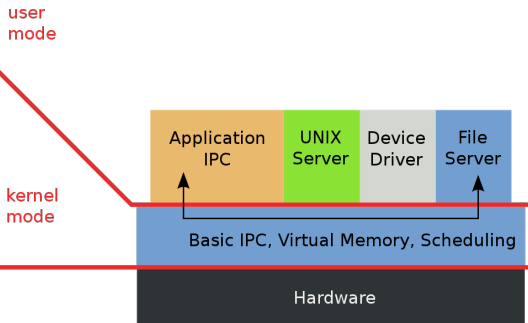
Прочие компоненты работают в пространстве пользователя.

Различие моно- и микро- ядер

Monolithic Kernel based Operating System



Microkernel based Operating System



История L4

L1 – компилятор ELAN (на основе Algol 68)

L2 – ОС EUMEL, эмулировалась на VM

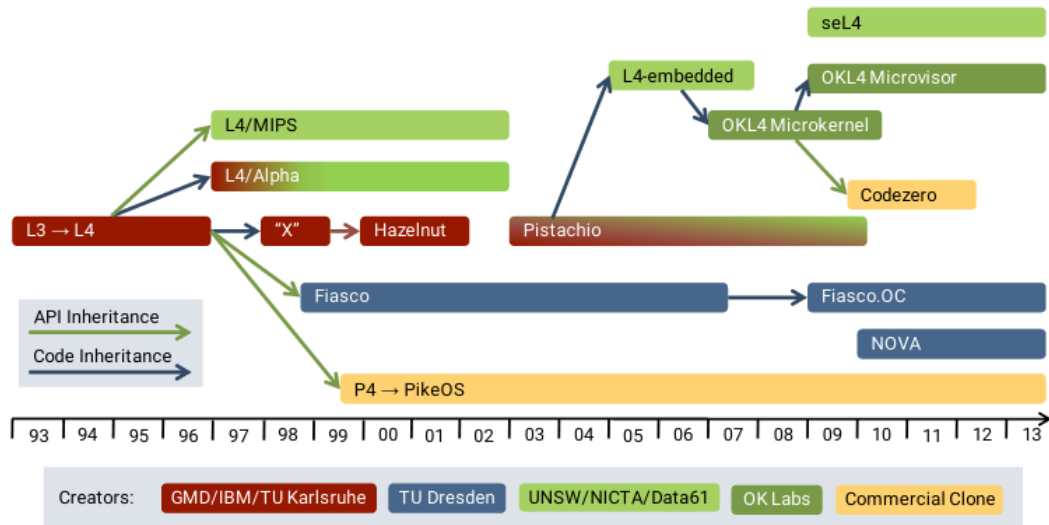
L3 – совместимая с EUMEL, но для x86

L4 – уход от принципов Mach (замена BSD UNIX от CMU)



Рис. 1: Jochen Liedtke 4/16

Семейство L4



- Абстрактная спецификация (интерфейсы, эффекты системных вызовов)
- Прототип на Haskell, определяющий исполнимую спецификацию (QEMU)
- Реализация на C и ASM (no GC, оптимизации, ограниченная семантика)
- Инструментарий, сопоставляющий коду его представление в Isabelle/HOL

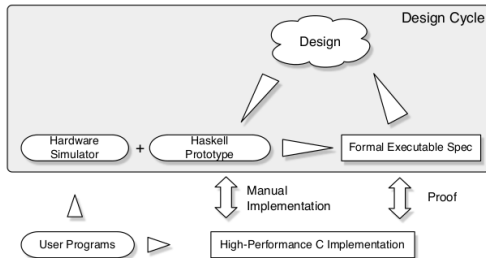


Figure 1: The seL4 design process

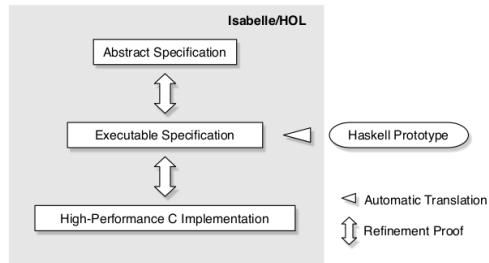
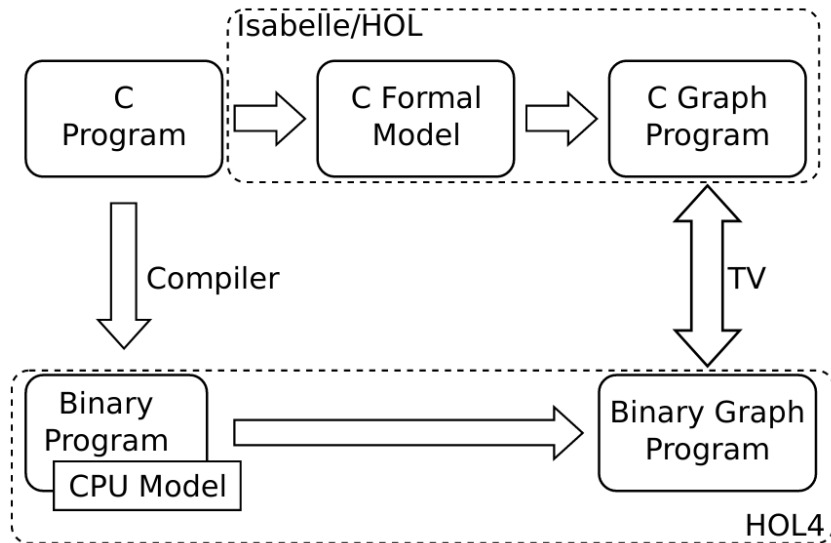


Figure 2: The refinement layers in the verification of seL4

Доказательство строится на гипотезе о корректности:

- компилятора (GCC) и архитектуры (ARMv6)
- кода сборки и загрузчика
- оборудования, в т.ч. управления кэшем



Запрещено использование следующих возможностей языка:

- архитектурно-зависимый размер типов
- применение оператора & к локальным переменным (на стеке)
- недетерминизм при исполнении функций с побочными эффектами
- вызов функций через указатели
- goto и fall-through switch
- union, оптимизации bitfields
- приведение типов указателей к void*

Программа π считается корректной, если для любых входных данных, удовлетворяющих предусловию ϕ , результат работы программ удовлетворяет постусловию ψ .

Запись вида $\phi \{ \pi \} \psi$ – триада Хоара.

(Много формализмов, аксиомы, правила действия операторов.)

Основная цель – доказать выполнение реализацией спецификации, а т.е.:

- сохранение инвариантов
- безопасное переиспользование блоков памяти
- прерывания запрещены в ядре (polling)
- единый стек

PROPERTIES OF THE ANALYZED seL4 BINARY.

Code size (bytes)	42,120
Number of instructions	10,271
Number of functions	228
Number of basic blocks	2,384
Number of loops	56

Метод:

- Замеры, определяющие стоимость базовых операций на процессоре
- Статический поиск самого длинного (дорогого) пути для всех методов ABI
 - Циклы конечной длины
 - Ограниченная глубина рекурсии
- Обработка в ИАС Chronos

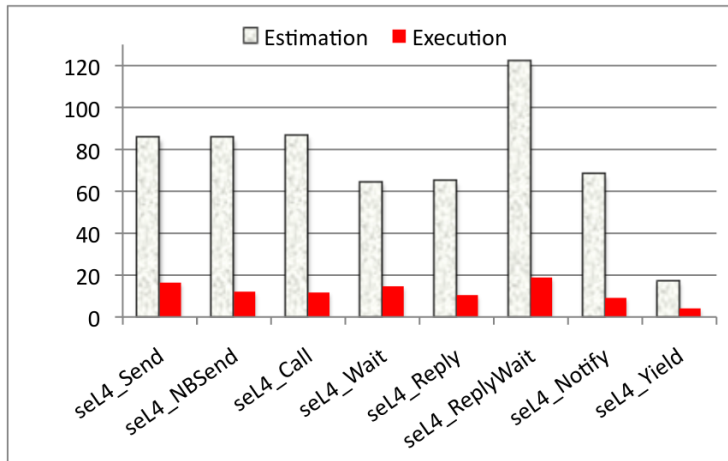






Figure 2. The error between estimation and real execution time for typical invocations of the seL4 system calls, measured in μs .

-  Bernard Blackham и др. “Timing analysis of a protected operating system kernel”. В: *2011 IEEE 32nd Real-Time Systems Symposium*. IEEE. 2011, с. 339—348.
-  Gernot Heiser. “The seL Microkernel An Introduction”. В: (2020).
-  Gerwin Klein и др. “seL4: Formal verification of an OS kernel”. В: *Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles*. 2009, с. 207—220.
-  Thomas Sewell, Felix Kam и Gernot Heiser. “Complete, high-assurance determination of loop bounds and infeasible paths for WCET analysis”. В: *2016 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*. IEEE. 2016, с. 1—11.