



— Загружаем Linux в RISC-V —

МФТИ
Осень 2024

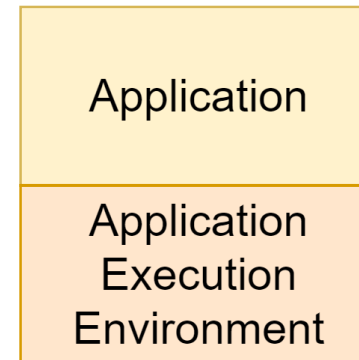
В начале было приложение...



Application

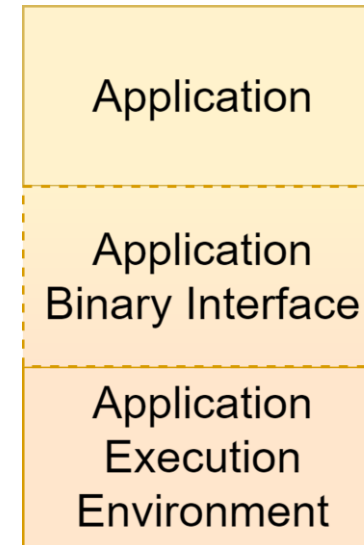
Практические аспекты

- Приложение исполняется в какой-то среде исполнения
- Среды исполнения могут быть разнообразными:
 - Baremetal
 - OS
 - Симулятор
- Возможно ли спроектировать их взаимодействие так, чтобы был возможен запуск приложения в разных АЕЕ без изменения его исходников?



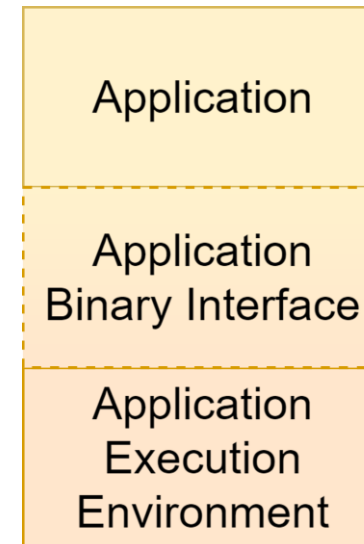
Абстрагируем среду исполнения

- Приложение взаимодействует со средой исполнения через ABI
- ABI включает в себя
 - ISA
 - Набор вызов для взаимодействия со средой исполнения
- ABI абстрагирует среду исполнения от приложения



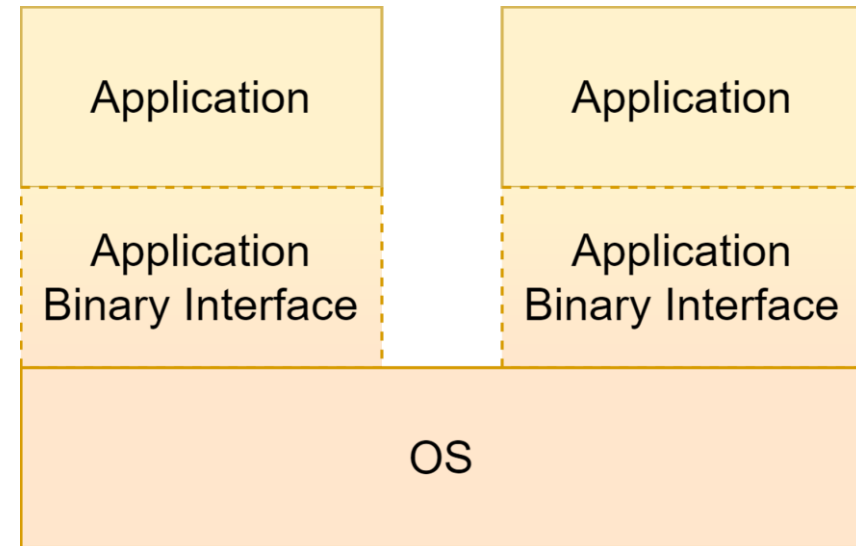
Baremetal AEE

- Baremetal среда исполнения неудобна:
 - Необходимо перекомпилировать приложение при каждом изменении среды исполнения
 - Для запуска другого приложения необходимо перепрошивать вычислительную систему



OS AEE

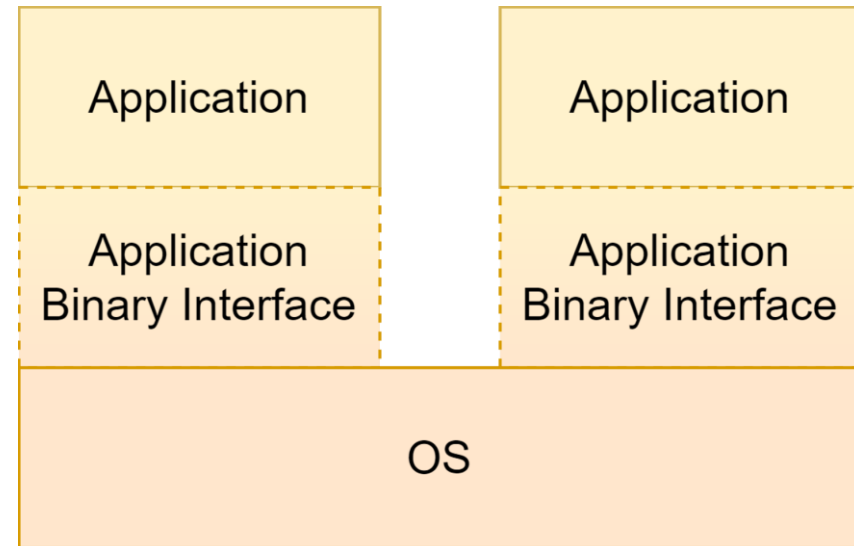
- OS среда исполнения предоставляет намного больше возможностей:
 - Одновременный запуск нескольких приложений
 - Более богатый набор абстракций
 - Больше число драйверов из коробки



Проблемы наивной реализации OS AEE

Проблема 1:

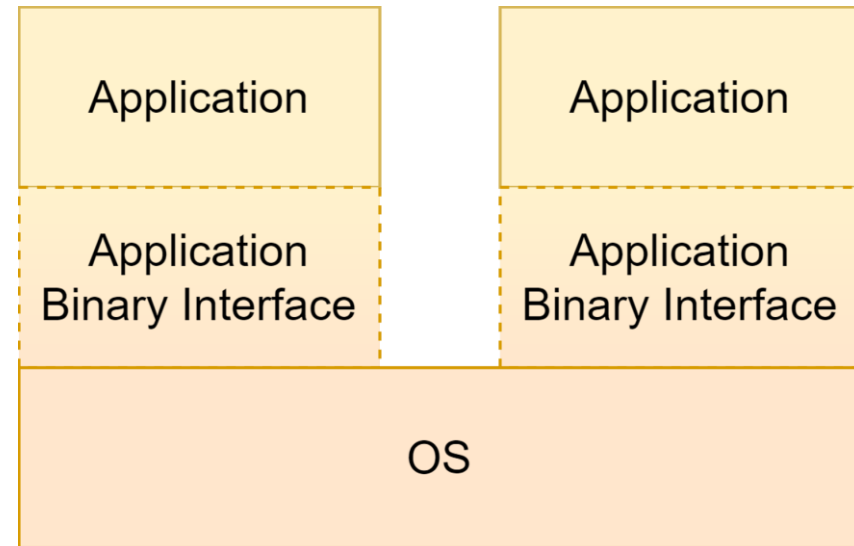
- Операционная система настроила прерывания для обработки ввода с клавиатуры
- Приложение изменяет настройку прерываний
- ???
- BOOM!!!



Проблемы наивной реализации OS AEE

Проблема 2:

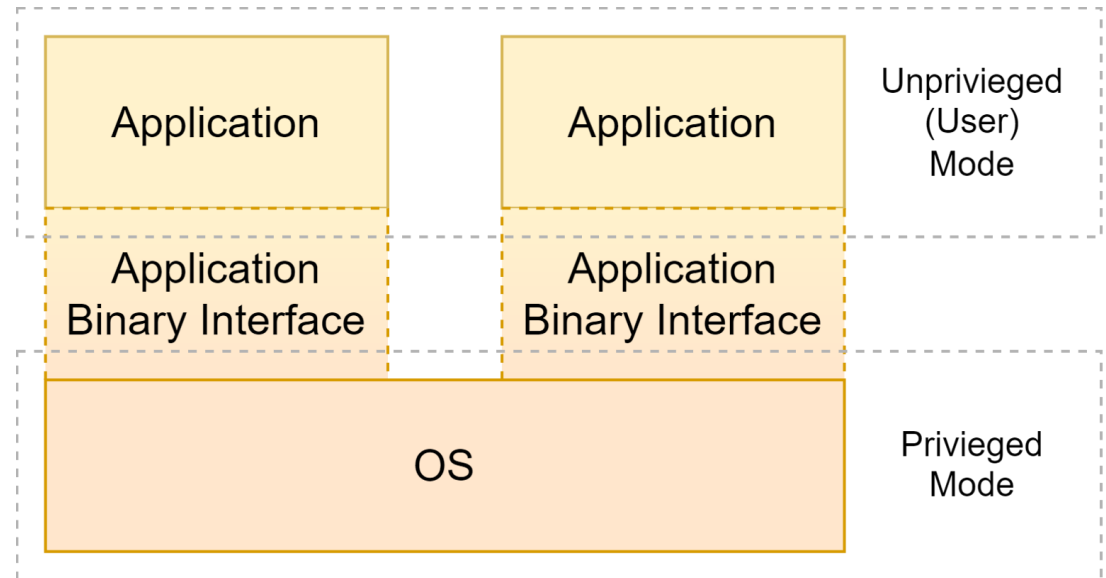
- Операционная система настроила контроллер памяти для аппаратной трансляции адресов
- Один процесс изменяет эти настройки для себя и портит память, принадлежащую другому процессу
- ???
- BOOM!!!



Решение проблемы с доступом

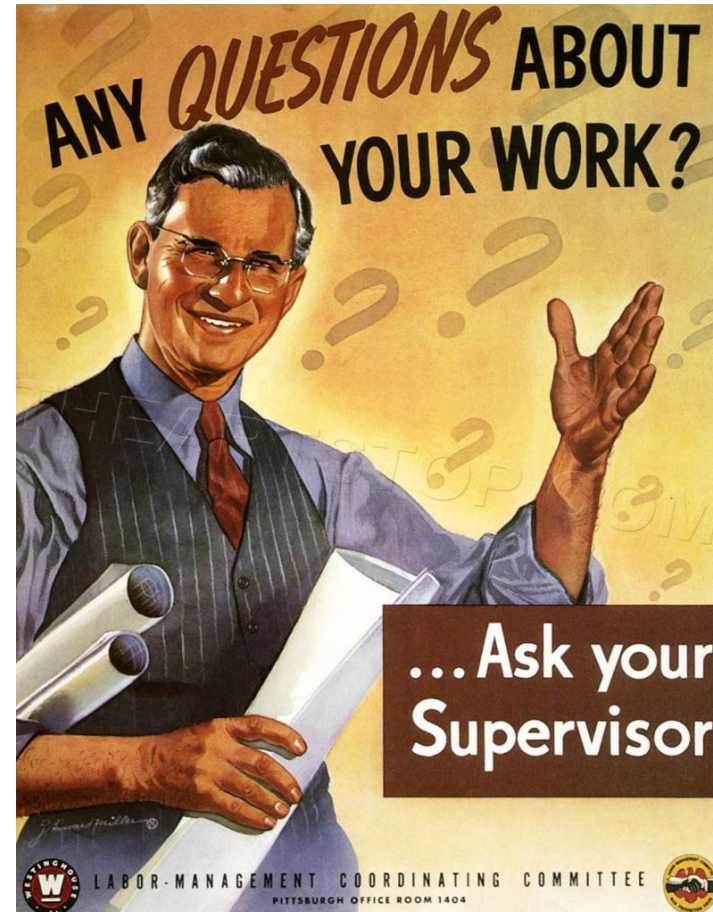
Вводим режимы исполнения:

- Привилегированный
 - Доступны все архитектурные возможности
- Непривилегированный
 - Доступно ограниченное количество архитектурных возможностей
 - В RISC-V называют User модом (U-mode)



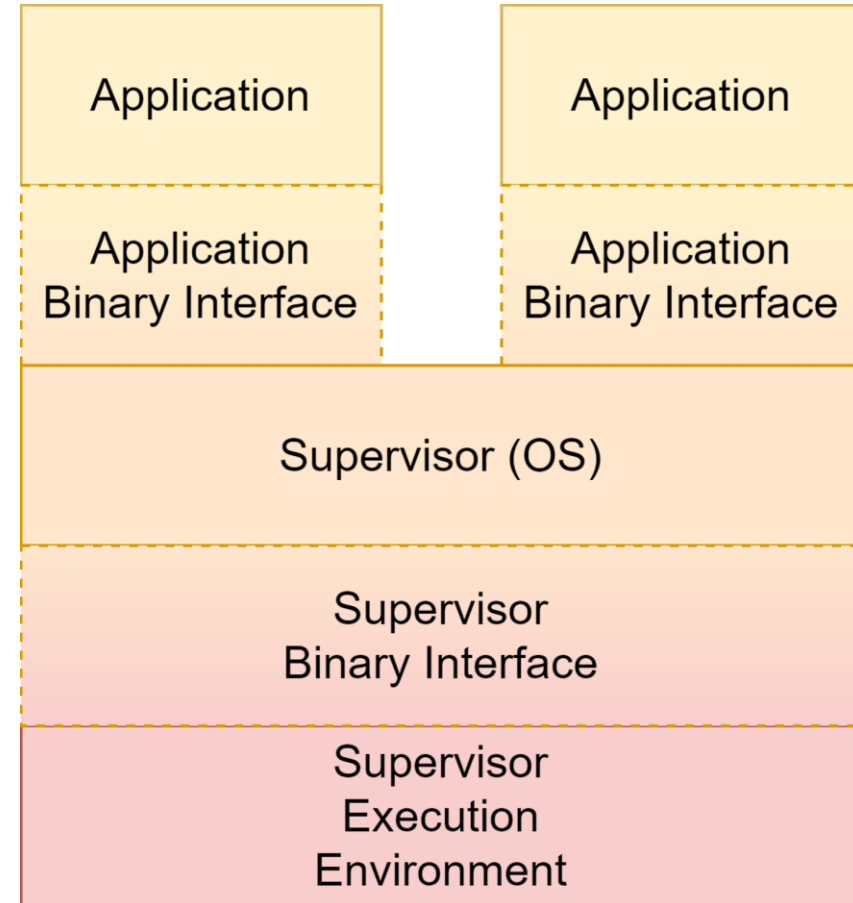
Абстракция супервизора

- Супервизор – программа, управляющая исполнением других программ, их планированием, операциями ввода-вывода и обработкой ошибок
- Впервые этот термин ввел IBM, называя так линейку операционных систем OS/360
- Сегодня супервизором в ОС называют ядро



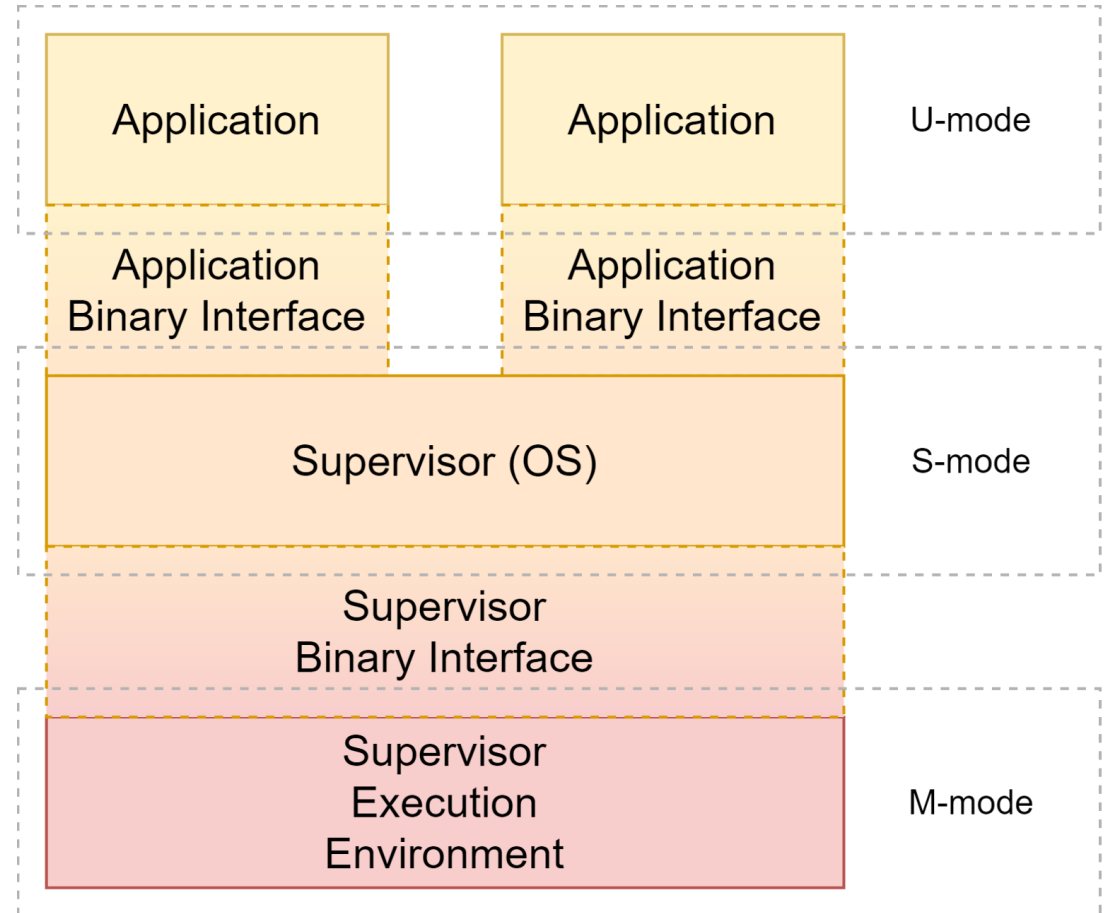
Абстракция супервизора

- По аналогии с обычными приложениями, ядро выполняется в особой среде исполнения
- В RISC-V уровень ОС называют **Supervisor**
- Ее среду исполнения – **Supervisor Execution Environment (SEE)**
- Интерфейс предоставляющий SEE – **Supervisor Binary Interface (SBI)**



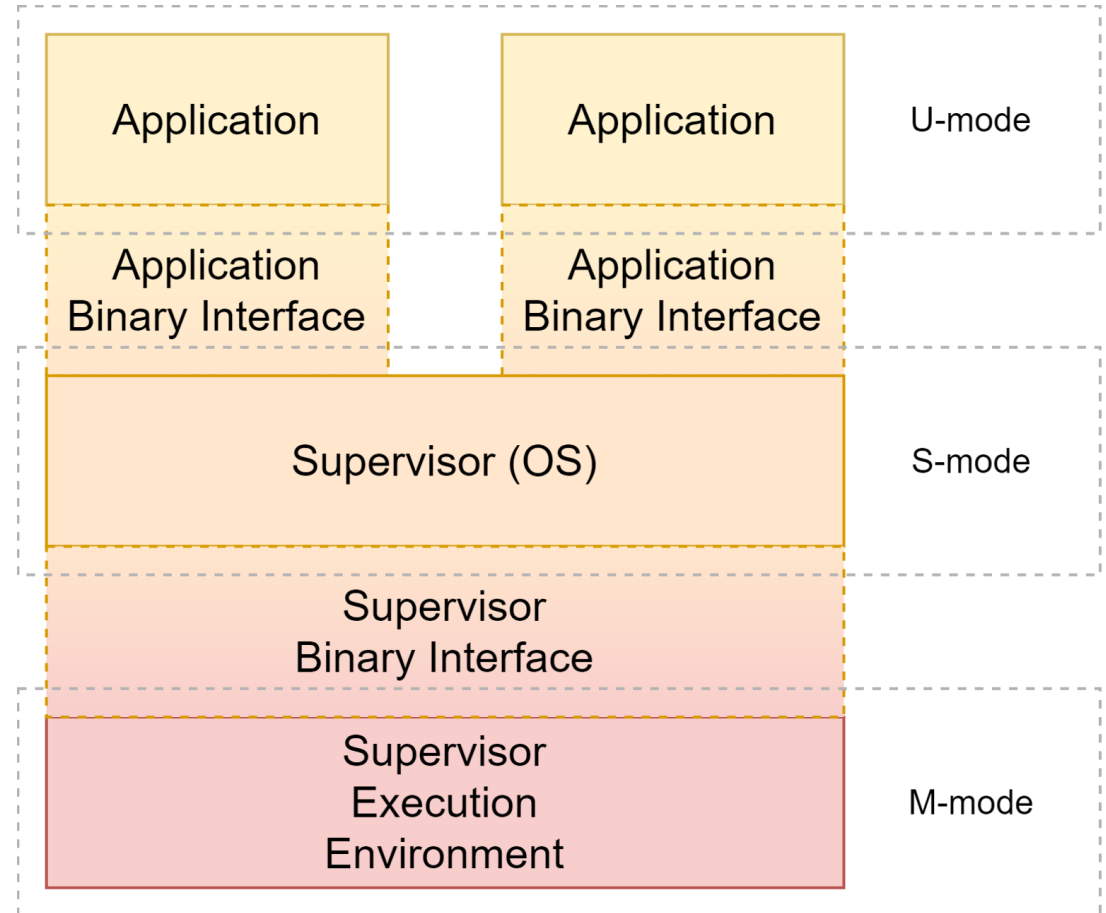
+1 привилегированный режим

- По аналогии с разделением привилегированности исполнения между *приложением* и *супервизором*, можно создать разделение между *супервизором* и *его средой исполнения*
- В RISC-V режим исполнения супервизора называется S-mode (Supervisor mode)
- Наиболее привилегированный режим называется M-mode (Machine mode)



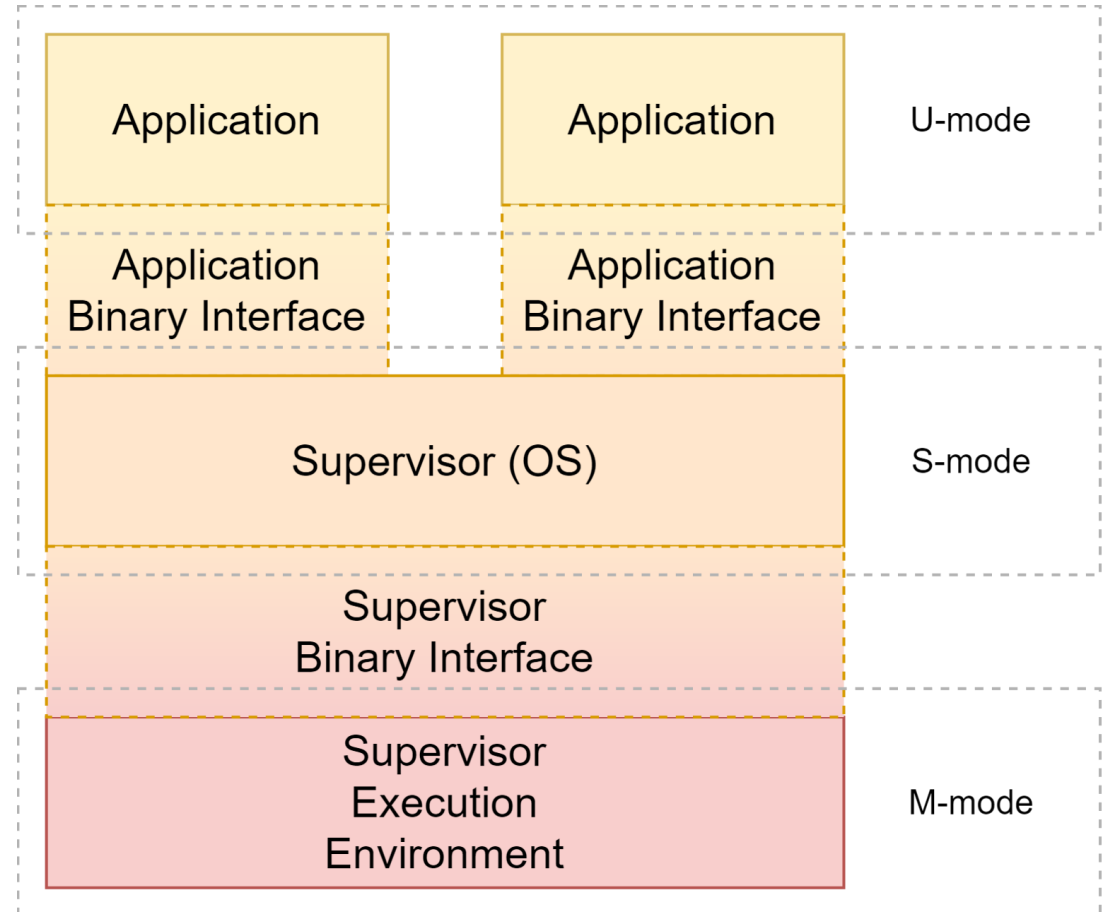
Supervisor Execution Environment

- Среду исполнения для супервизора могут предоставить:
 - Загрузчик
 - Симулятор
 - Гипервизор



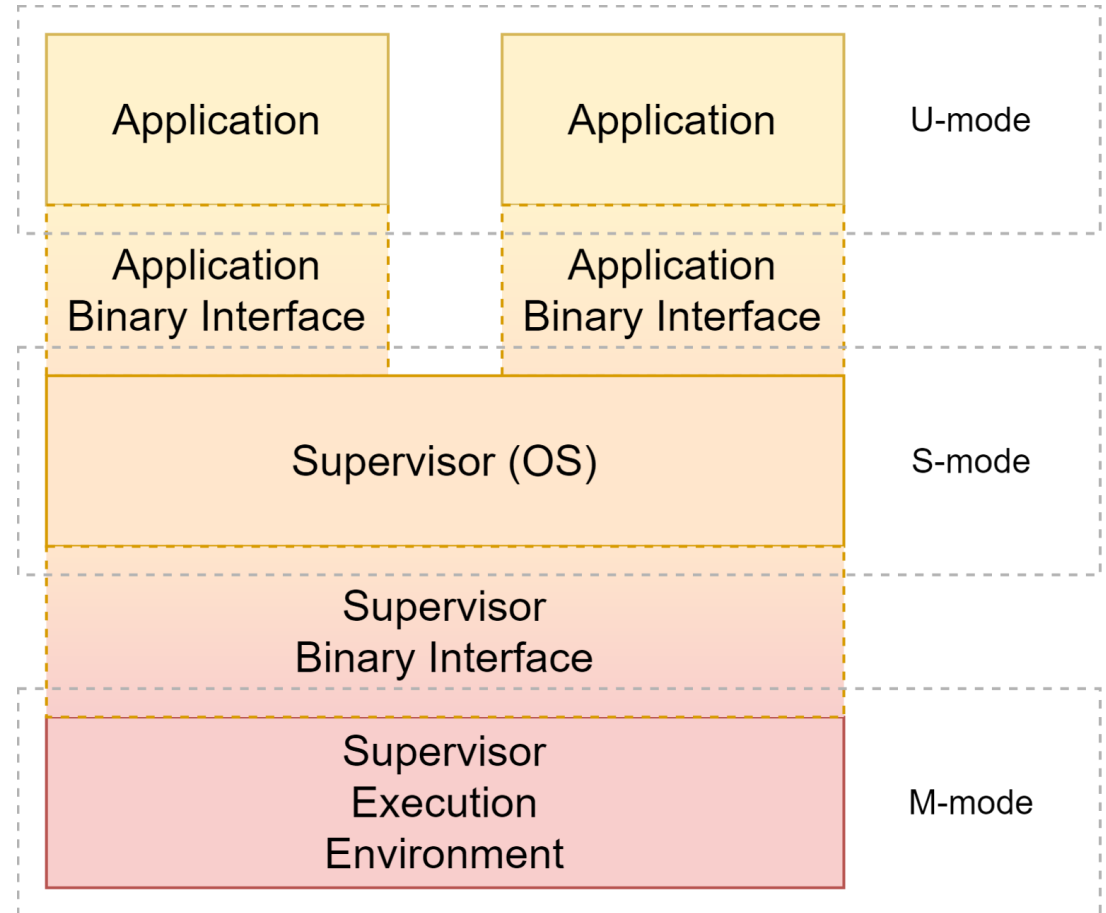
Bootloader-based SEE

- Загрузчик линкуется с реализацией SBI
- SBI в M-mode производит минимальную настройку процессора
- После этого SBI передает управление загрузчику, переключая выполнение в S-mode
- Загрузчик загружает ядро в память и передает управление ядру



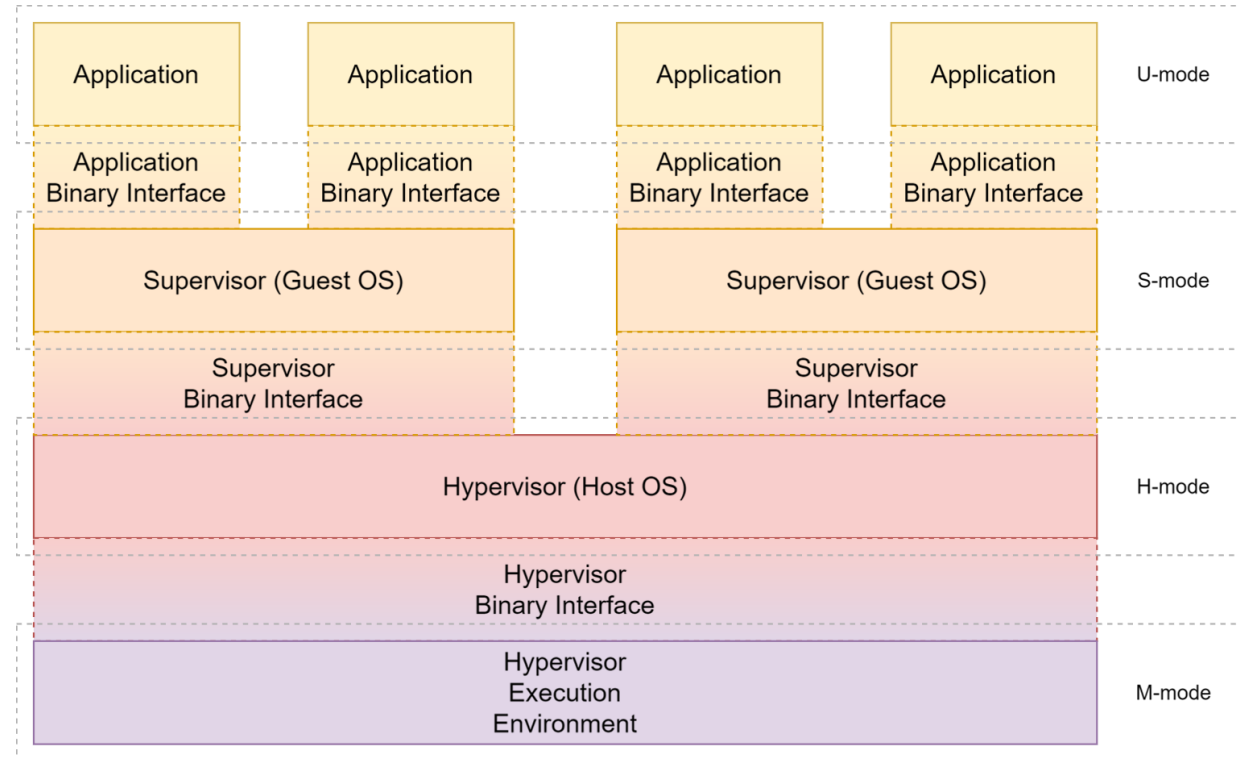
Simulator-based SEE

- Честная реализация M-mode может быть весьма сложной
- Помимо этого, полная честная реализация M-mode требует точной реализации периферии (uart, spi, timers и др.)
- Поэтому некоторые симуляторы реализуют M-mode частично, делегируя недостающую функциональность через Semi-Hosting



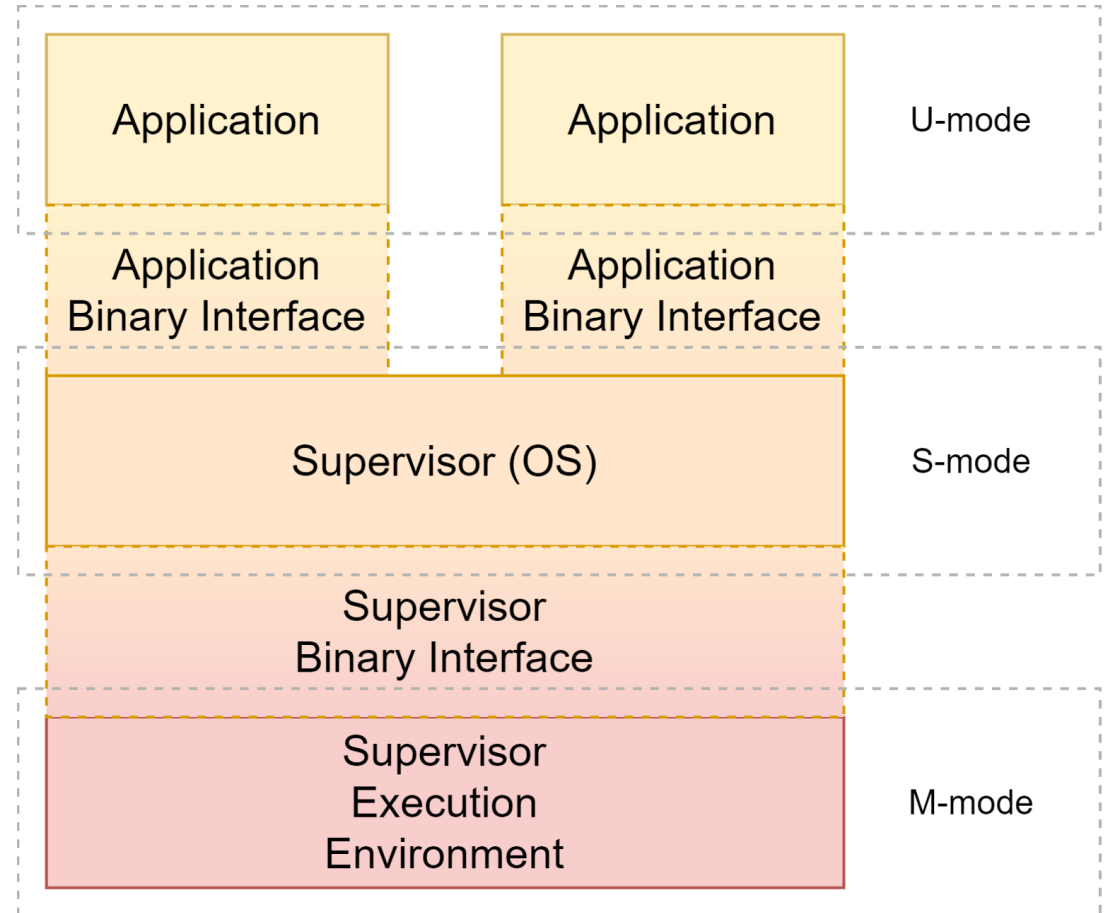
Hypervisor-based SEE

- Гипервизор – подсистема вычислительной системы, обеспечивающая возможность запуска сразу нескольких ОС
- Виртуализирует окружение исполнения операционных систем
- Самый распространенный вариант реализации – хозяйская ОС, предоставляющая механизмы виртуализации гостевым ОС



Типичный стек в RISC-V

- Реализация SBI
 - OpenSBI
 - M-mode
- Загрузчик uboot
 - S-mode
- Linux
 - S-mode
- Приложения
 - U-mode



Минимальные требования к платформе для запуска Linux

- Unprivileged:
 - I-extension
 - M-extension
 - A-extension
 - Zicsr-extension
- Privileged:
 - U-mode
 - Несколько CSR

M-extension

- Реализует аппаратную поддержку умножения и деления
- Произведением двух чисел с M разрядов в каждом является число с $2M$ разрядов
- То есть результат умножения двух регистров поместится только в два регистра
- MUL – умножить два числа, записать нижние 32 знака
- MULH[[S]U] – умножить два числа, записать верхние 32 знака

Muls	Divs
MUL	DIV
MULH	DIVU
MULHU	REM
MULHSU	REMU
MULW	DIVW
	DIVUW
	REMW
	REMUW

Зеленым обозначены
RV64M-only инструкции

A-extension

- Реализует аппаратную поддержку некоторых атомарных операций
- Эти инструкции используются для реализации примитивов многопоточной синхронизации
- Атомарный ADD может выполняться в 500-600 раз медленнее, потому что выполняется, синхронизируя данные в памяти во всех ядрах
- В одноядерных системах от обычных операций отличаются наложенными ограничениями на переупорядочивание таких операций

CAS
LR.W
SC.W
LR.D
SC.D

AMO	
AMOSWAP.W	AMOSWAP.D
AMOADD.W	AMOADD.D
AMOAND.W	AMOAND.D
AMOODR.W	AMOODR.D
AMOXOR.W	AMOXOR.D
AMOMAX[U].W	AMOMAX[U].D
AMOMIN[U].W	AMOMIN[U].D

U-mode и CSR'ы

Для минимальной сборки линукса потребуются следующие CSR:

- mstatus
- cyclel
- cycleh
- timerl
- timerh
- timermatchl
- timermatchh
- mscratch
- mtvec
- mie
- mip
- mepc
- mtval
- mcause

А так же набор флагов:

- U/M-mode
- Резервация LR/SC
- WFI (wait for interrupt)

Zicsr
CSRRW
CSRRS
CSRRC
CSRRWI
CSRRSI
CSRRCI

Задание: пишем симулятор

- Реализуйте в вашем симуляторе M и A расширения
- Реализуйте в вашем симуляторе Zicsr расширение с указанными CSR
- Реализуйте в вашем симуляторе U-mode
- Попробуйте загрузить линукс =)

To be continued ...

На следующем занятии

- Узнаем какие могут возникнуть сложности при портировании приложений на новые архитектуры

Список литературы

- The RISC-V Instruction Set Manual Volume I Unprivileged Architecture Version 20240411 // Chapter 13. "M" Extension for Integer Multiplication and Division, Version 2.0
- The RISC-V Instruction Set Manual Volume I Unprivileged Architecture Version 20240411 // Chapter 14. "A" Extension for Atomic Instructions, Version 2.1
- The RISC-V Instruction Set Manual Volume I Unprivileged Architecture Version 20240411 // Chapter 7. "Zicsr", Extension for Control and Status Register (CSR) Instructions, Version 2.0
- The RISC-V Instruction Set Manual: Volume II Privileged Architecture Version 20240411 // Chapter 2. Control and Status Registers (CSRs)
- The RISC-V Instruction Set Manual: Volume II Privileged Architecture Version 20240411 // Chapter 3. Machine-Level ISA, Version 1.13
- The RISC-V Instruction Set Manual Volume I Unprivileged Architecture Version 20240411 // 1.6. Exceptions, Traps, and Interrupts