Конструирование ядра операционных систем (II)

Загрузка и отладка

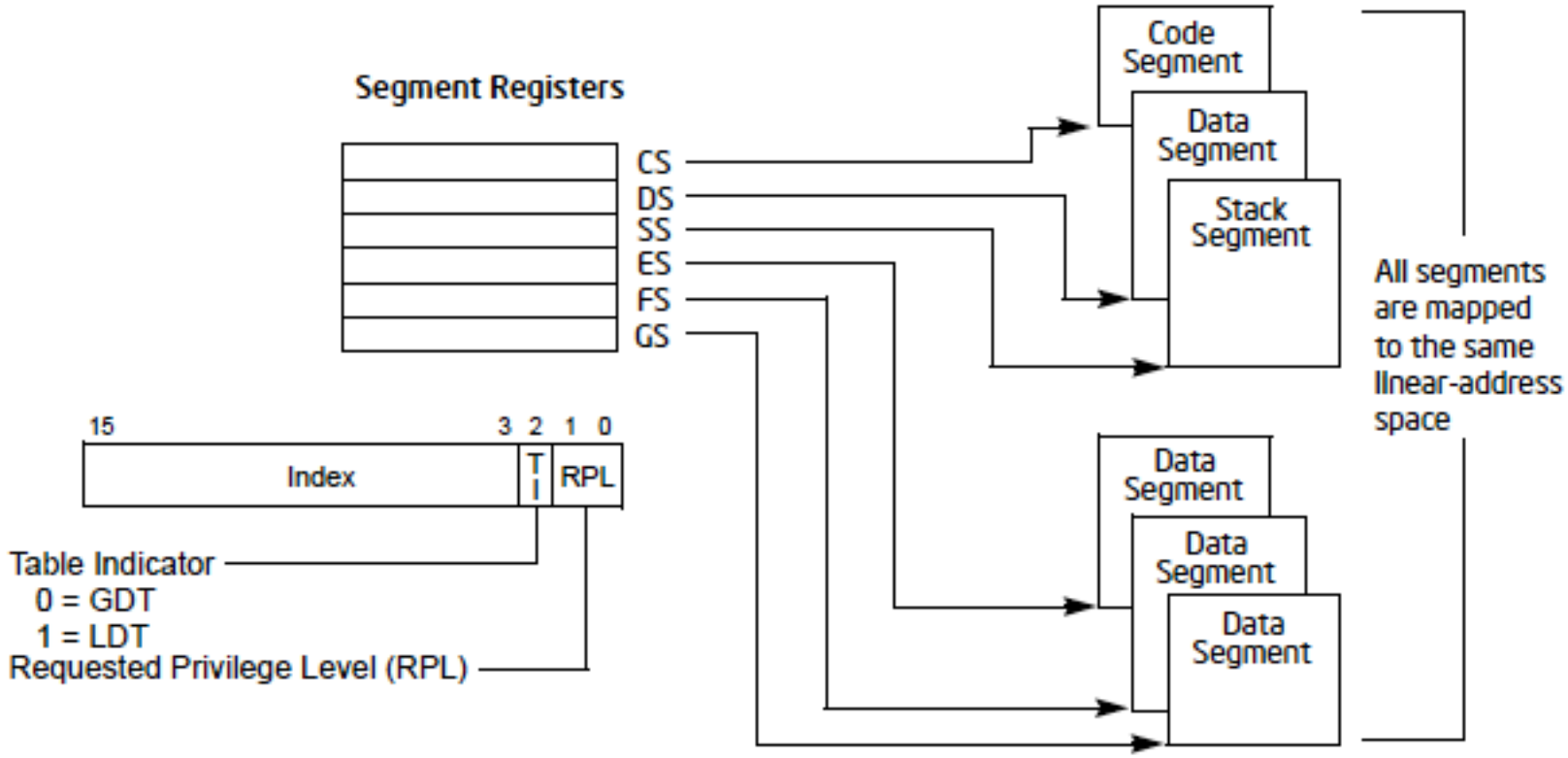
План

- Сегментная адресация x86
- Режимы процессора x86
- Средства отладки при разработке ОС
- Практическая часть лабораторной работы

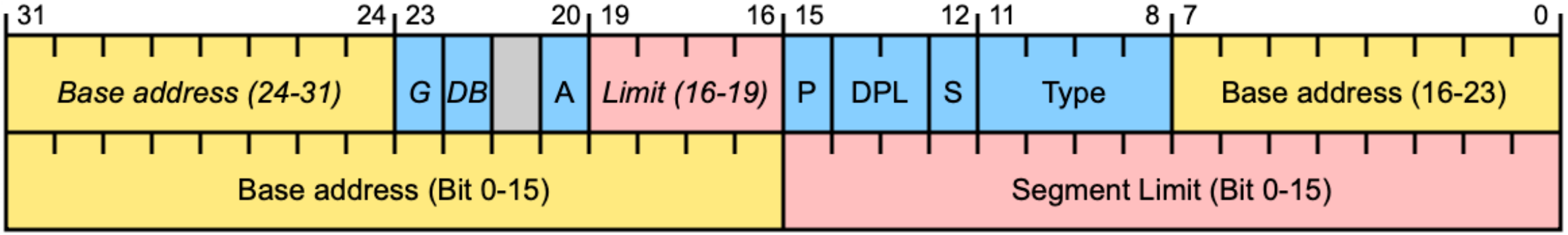
Адресация x86

- Сегментная адресация — классический механизм адресации, доступный начиная с 8086.
  - Контролируется сегментными регистрами (cs, ds, es, fs, gs, ss) и таблицами дескрипторов (GDT, LDT).
  - Выбрать активную область физической памяти до Intel 64.
  - Определяет режим процессора при исполнении кода в сегменте.
  - Определяет права доступа сегмента (по аналогии с MPU).
- Страничная адресация — механизм адресации через виртуальную память, доступный начиная с 80386.
  - Контролируется через таблицу страниц в регистре CR3.
  - Обязательна для 64-битного режима.

Сегментный регистр (селектор)

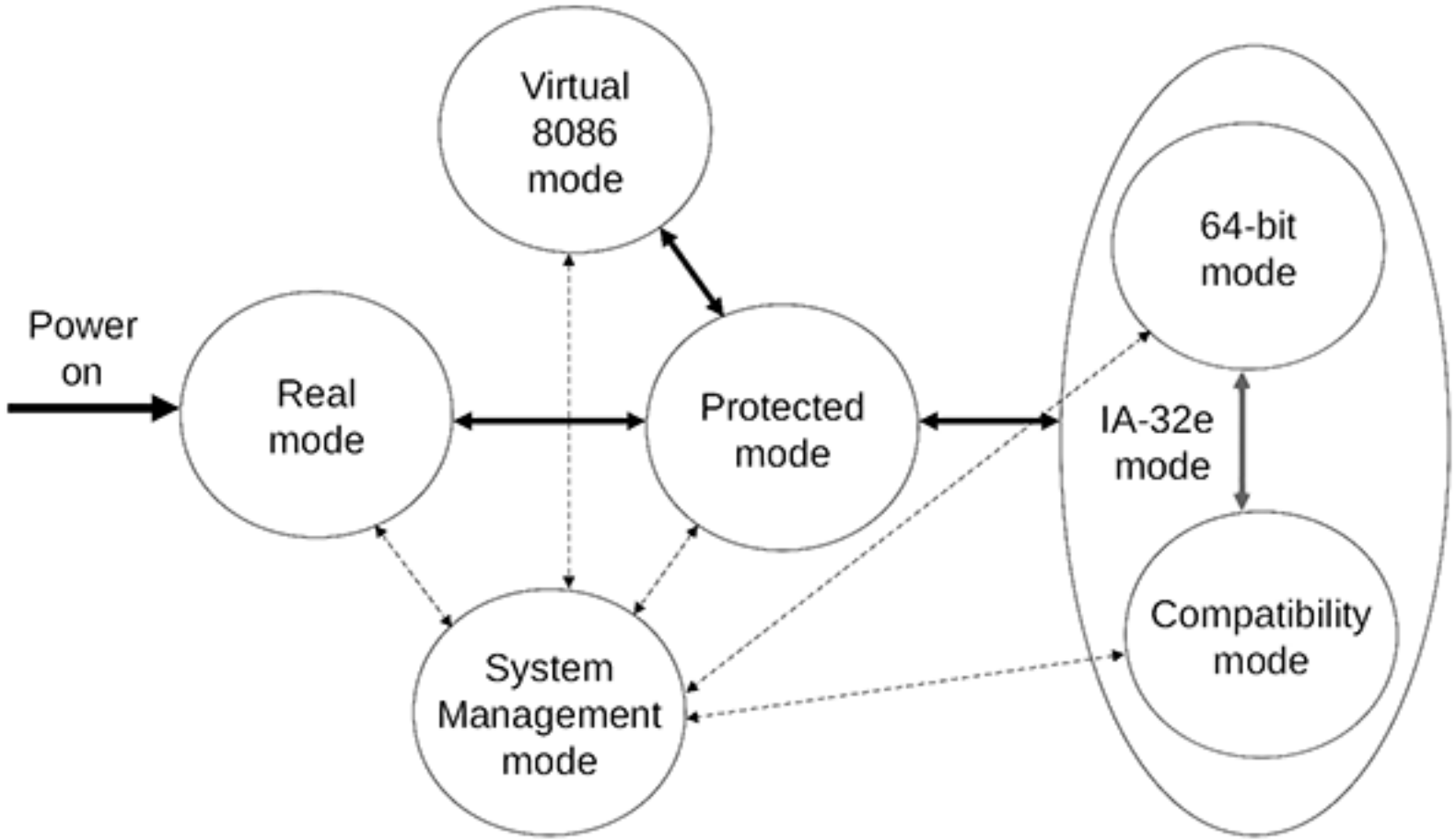


Сегмент (запись в GDT/LDT)



- Base address / Segment Limit — начало/конец сегмента
- G (Granularity) — Segment Limit в страницах (4K)
- D/ B — 32-битный кодовый сегмент / сегмент данных
- L — 64-битный сегмент (D сброшен)
- Type — Code vs Data / P — Present / DPL — Privilege level
- R — Readable / W — Writable / A — Accessed

Режимы x86


Virtual
8086
mode
Power
on
Real
mode
Protected
mode
64-bit
mode
IA-32e
mode
Compatibility
mode
System
Management
mode

Отладка ОС

Для отладки/аварийного управления ОС вне зависимости от состояния часто реализуют дополнительные возможности:

- Вывод информации о возникшей ошибке в ядре (backtrace, значения регистров, информация о драйверах и железе).
- Обработка комбинаций клавиш в ядре для специальных задач (e.g. SysRq, Ctrl+Alt+Del, Ctrl+Opt+Cmd+Shift+Esc).
- Локальные и удалённые отладчики ядра (KDB, KGDB, …).
- Сохранение логов аварийных завершений для анализа (SWAP, файловая система, NVRAM, SMC).

Альтернативы

- Наличия отладчика на вышестоящем программном уровне;
- Отладчик на IP уровне (In-Target Probe): JTAG, SWD, XDP;
- Эмулятор оборудования (In-circuit emulator).

Инструменты в JOS

GDB Stub в QEMU имеет возможности сопоставимые с аппаратным отладчиком и позволяет отлаживать JOS практически в любом состоянии.

Также в JOS имеется реализация монитора, работающая через COM-порт, для взаимодействия с пользователем на уровне команд. *На данный момент это единственный интерфейс пользователя.*

Задачи лабораторной

- Написать реализацию перехода в 64-битный режим из 32-битной прошивки OVMF для запуска JOS.
- Написать реализацию команды backtrace в мониторе JOS для вывода стека вызовов на экран с указанием адресов, файлов, строк, функций.

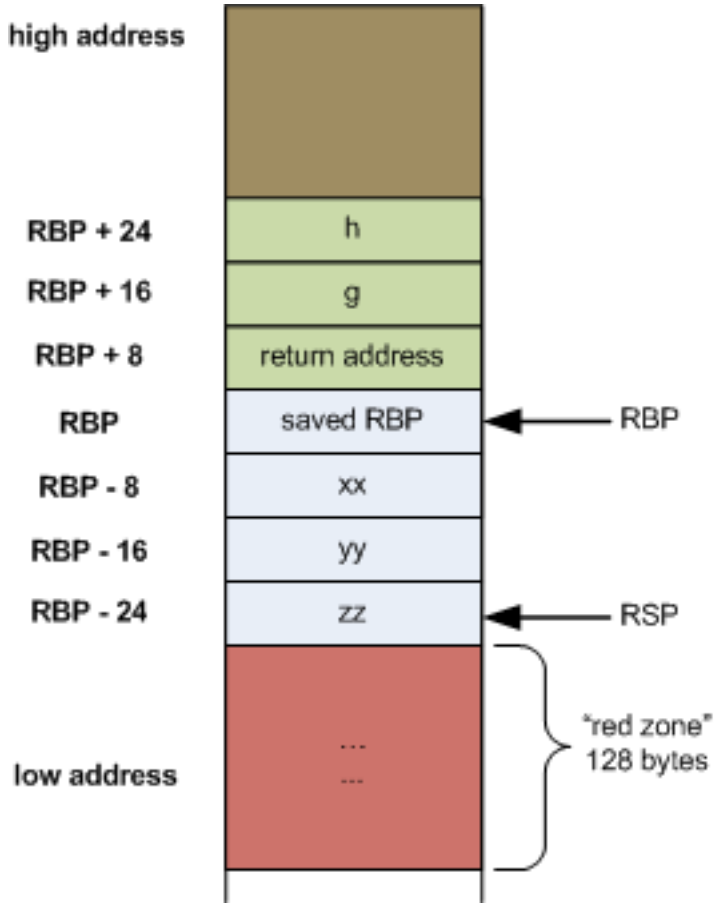
Hint: Для команды backtrace необходимо обратиться к отладочной информации, основная задача которой соотнести сгенерированный машинный код и исходный код. Форматов много: STABS, DWARF, PDB. В JOS используется формат DWARF.

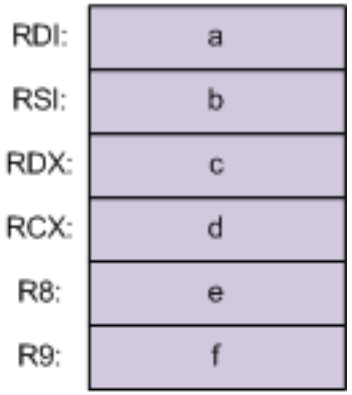
Выравнивание (Alignment)

Наборы команд современных процессоров не всегда могут адресовать (с одинаковой производительностью) области данных различных размеров по различным адресам. Одна из причин — выравнивание адресов.

Alignment — requirement that objects of a particular type be located on storage boundaries with addresses that are particular multiples of a byte address (ISO/IEC 9899 2018, Programming languages — C).

System V ABI x86-64





В UEFI используется Microsoft x86-64 ABI.

Скаляры и указатели передаются через регистры RCX, RDX, R8, R9.

docs.microsoft.com/cpp/build/x64-calling-convention

Полезная литература

- Intel x86-64 ABI: Function Calling Sequence
- Executable and Linking Format (ELF) Specification
- DWARF specification
- Intel® 64 and IA-32 Architectures Software Developer’s Manual Volume 2: Instruction Set Reference, A-Z

Спасибо за внимание!

Вопросы?