Конструирование ядра операционных систем (I)

Введение

План

- Элементы архитектуры x86
- Базовое программное обеспечение (BIOS/UEFI)
- Процесс загрузки операционной системы
- Операционная система JOS
- Практическая часть лабораторной работы

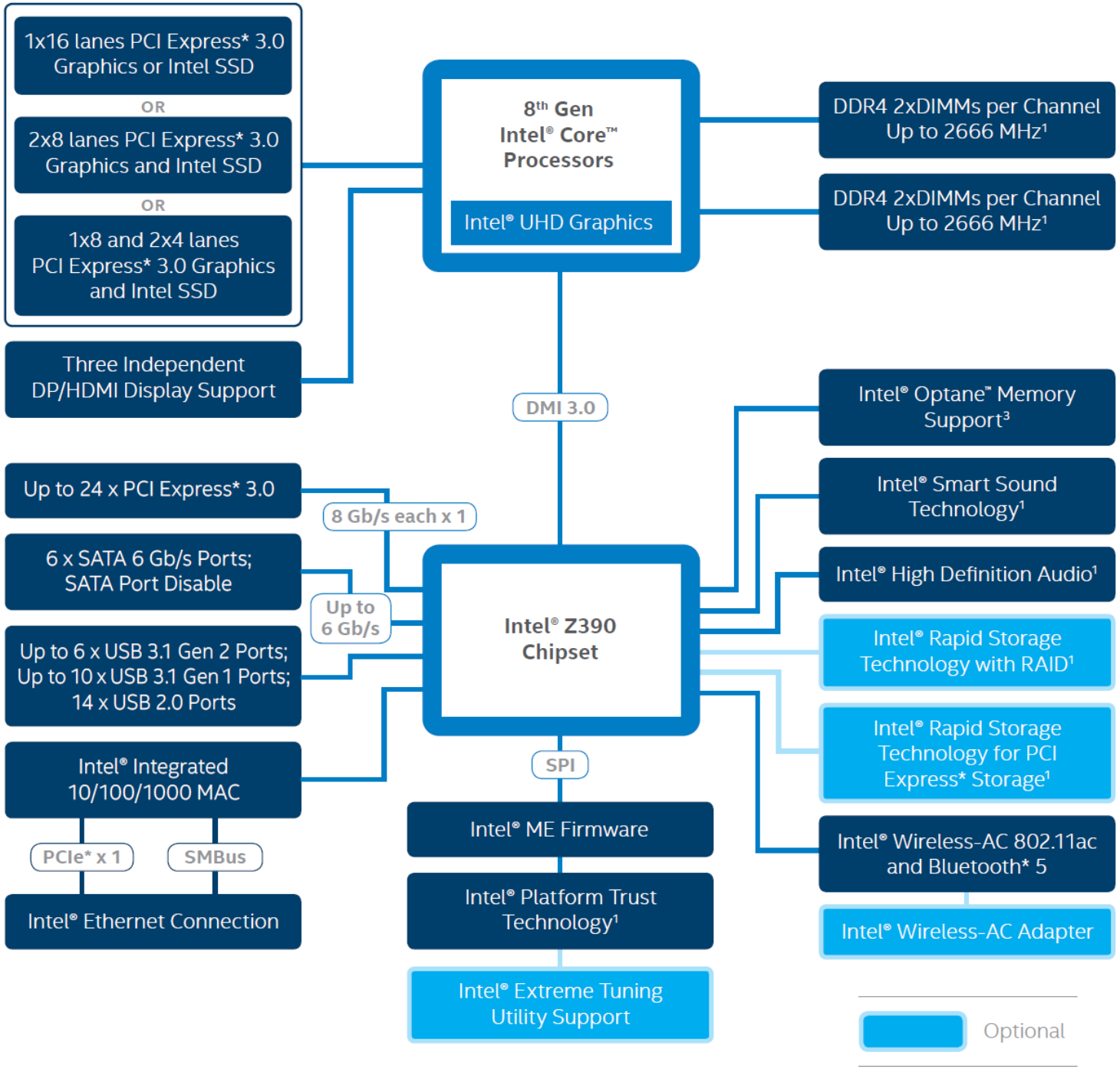
Чипсет

Intel Quark (x86, 32-bit)
- BootGuard
- ME firmware (MINIX 3)
- PAVP/TXT …

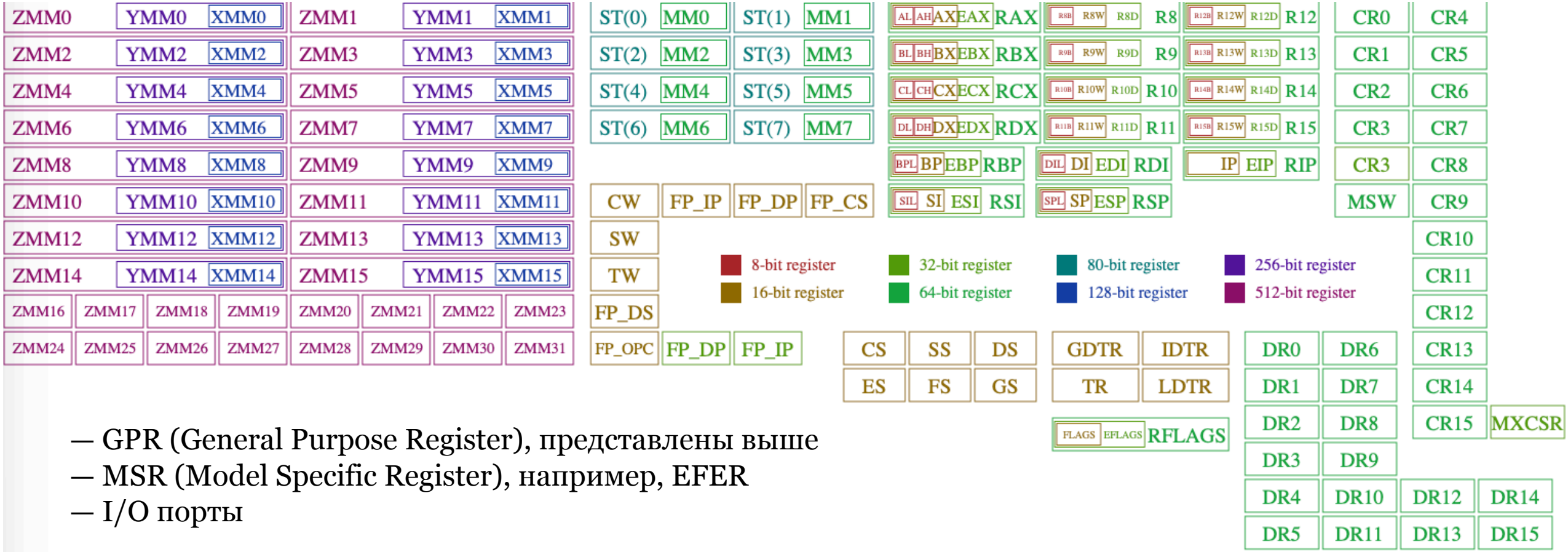
Intel Core (x86, 64-bit)
- Microcode
- UEFI firmware
- GPU Option ROM
- GBE Option ROM
- RAID Option ROM …

Other
- Wireless firmware
- SSD/HDD firmware
- TPM firmware …

Ref: Lab1-BH19-CSME.pdf



Прикладной уровень



— GPR (General Purpose Register), представлены выше
— MSR (Model Specific Register), например, EFER
— I/O порты

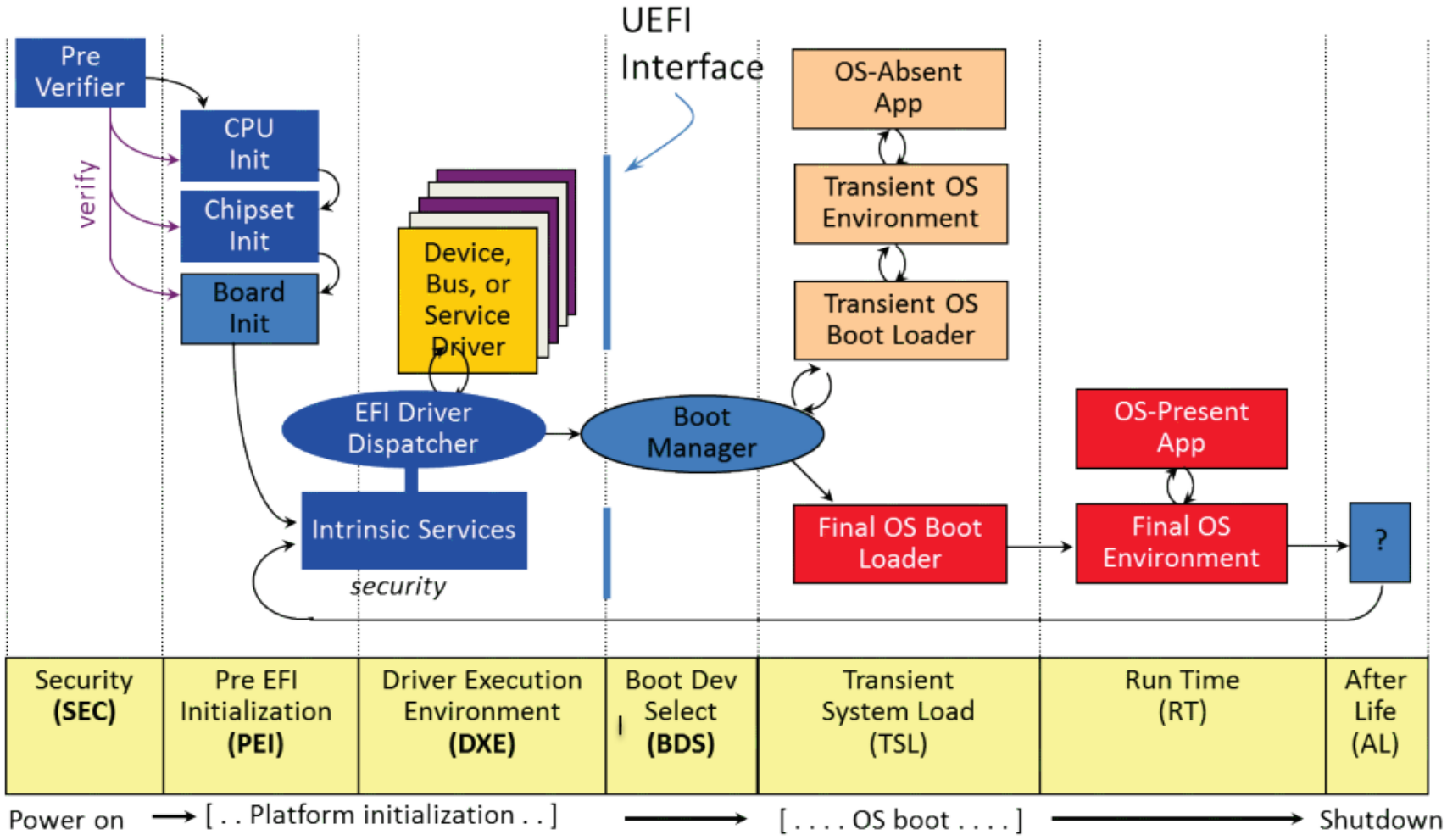
Процесс загрузки прошивки

Bootstrap:
— eSPI/SPI ROM
— BootGuard

SEC:
— Reset Vector
— Microcode update
— Cache as RAM (CAR)
— Root of Trust

PEI:
— CPU init (MSR, PM)
— Platform init (MCH, ICH)
— RAM init
— S3 resume (опцион.)

Ref: Lab1-AAPL-T2.pdf
Ref: Lab1-CAR.pdf



Режимы адресации процессора

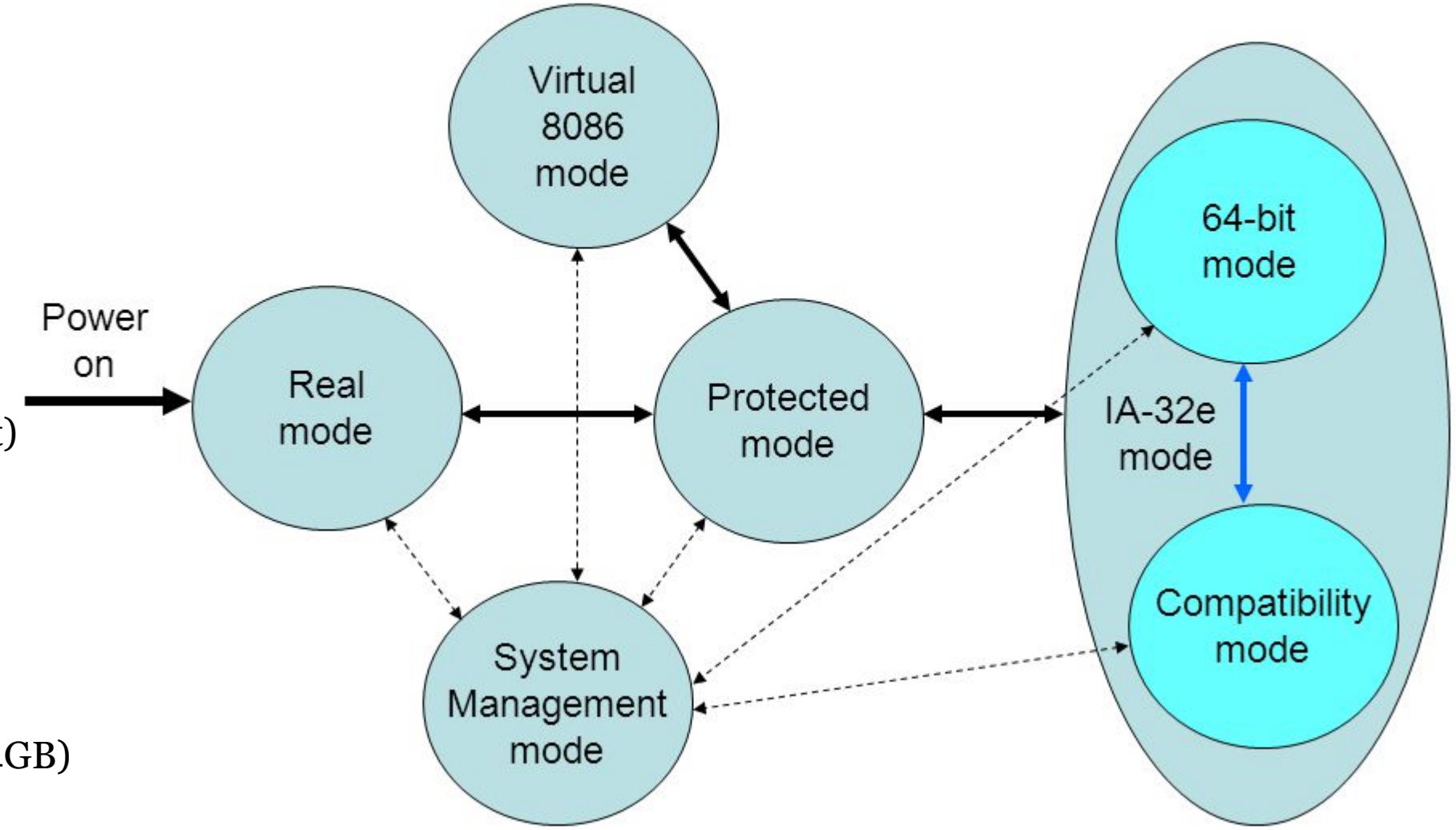
SEC:
— Real Mode (16-bit)
— Protected Mode (32-bit)

PEI:
— Protected Mode (32-bit)

DXE:
— Long Mode (32 or 64-bit)
— SMM (32 or 64-bit)

OS:
— ???
— SMM (32 or 64-bit)

— “Unreal” Mode (16-bit, 4GB)
— Protected Mode (16-bit)
— Paging + PAE

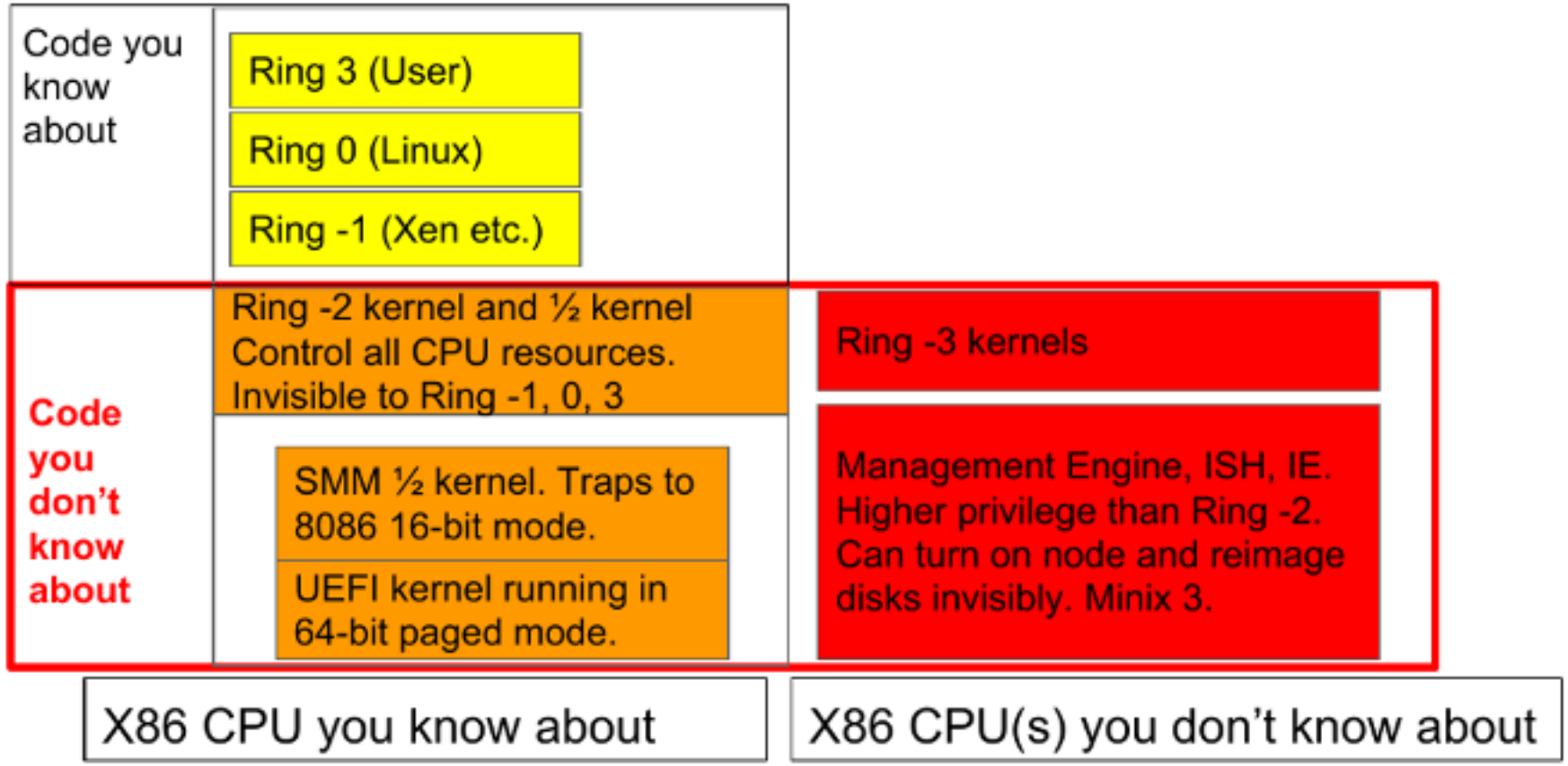


Режимы привилегий процессора

- Классические CPL уровни (3-0)
- Гипервизор (-1)
- UEFI Runtime Services, UEFI SMM, Microcode (-2)
- Management Engine, BMC вроде Apple T2 (-3)

Бонус:
- Nested virtualization ☺
- Third-party peripherals ☺

Ref: Intel SDM EPT Tables, Intel VT-d, Intel VT-x



BIOS vs UEFI

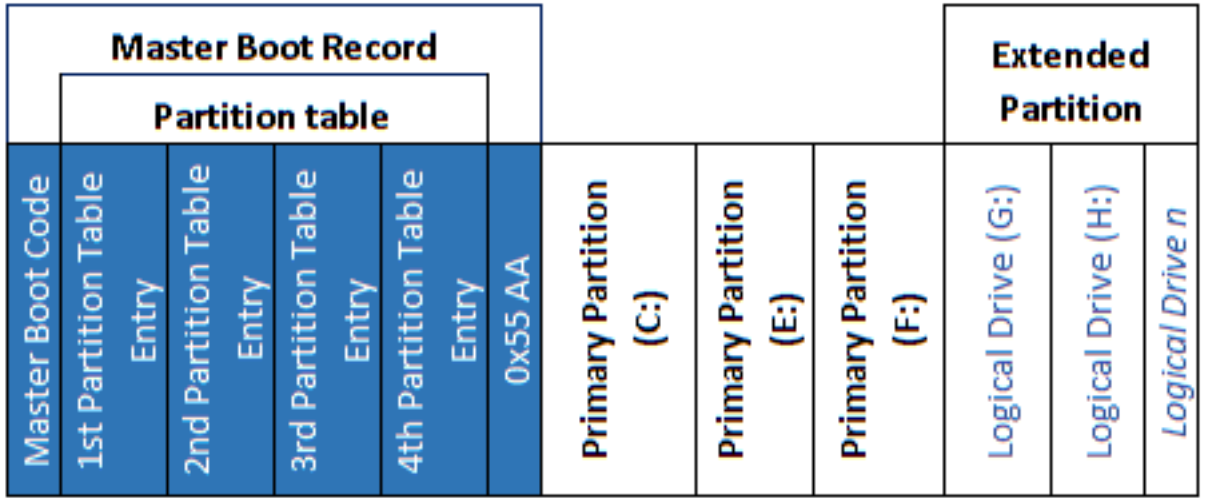
BIOS (x86)	**UEFI (ARM, x86, RISC-V) + PowerPC**
ASM/C; mostly 32-bit	ASM/C; mostly 64-bit
16-bit ASM interface Hardcoded PMIO & MMIO ACPI & COM & SSIO & SMBIOS	32-bit or 64-bit C interface Drivers & Protocols & Hobs ACPI & COM & SSIO & SMBIOS
PMIO Video Output PS/2 Support (optional PnP)	GOP/UGA Video Output (optional CSM) USB PnP (optional CSM)
Fixed memory model	UEFI Memory Map
Interrupt hooking for extension	Option ROMs
MBR boot & 16-bit handoff	GPT boot & 32-bit or 64-bit handoff (optional MBR/El Torito)
CMOS aka RTC “NVRAM” (128-256 bytes) Internal Flash NVRAM	CMOS aka RTC “NVRAM” (256 bytes) UEFI Variable Flash NVRAM

Ref: ACPI Specification, UEFI specification, UEFI PI specification на uefi.org/specifications

MBR vs GPT

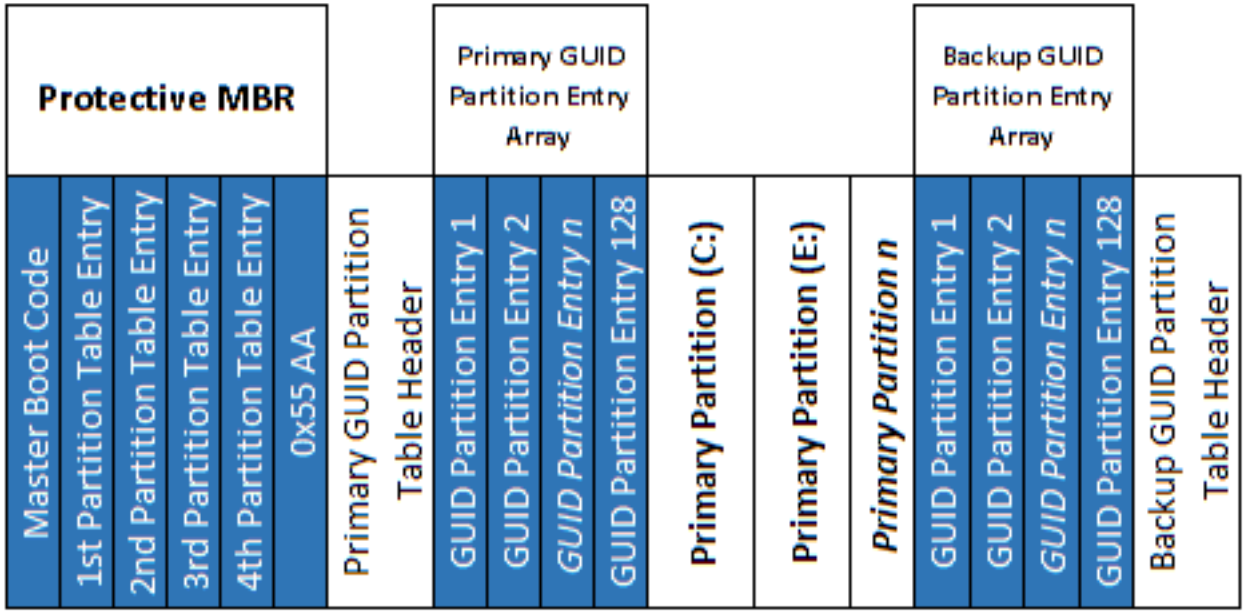
→ Стадия BDS или BIOS
→ MBR загрузчик по физ. сектору диска
→ Основной загрузчик ОС
→ Ядро ОС

MBR

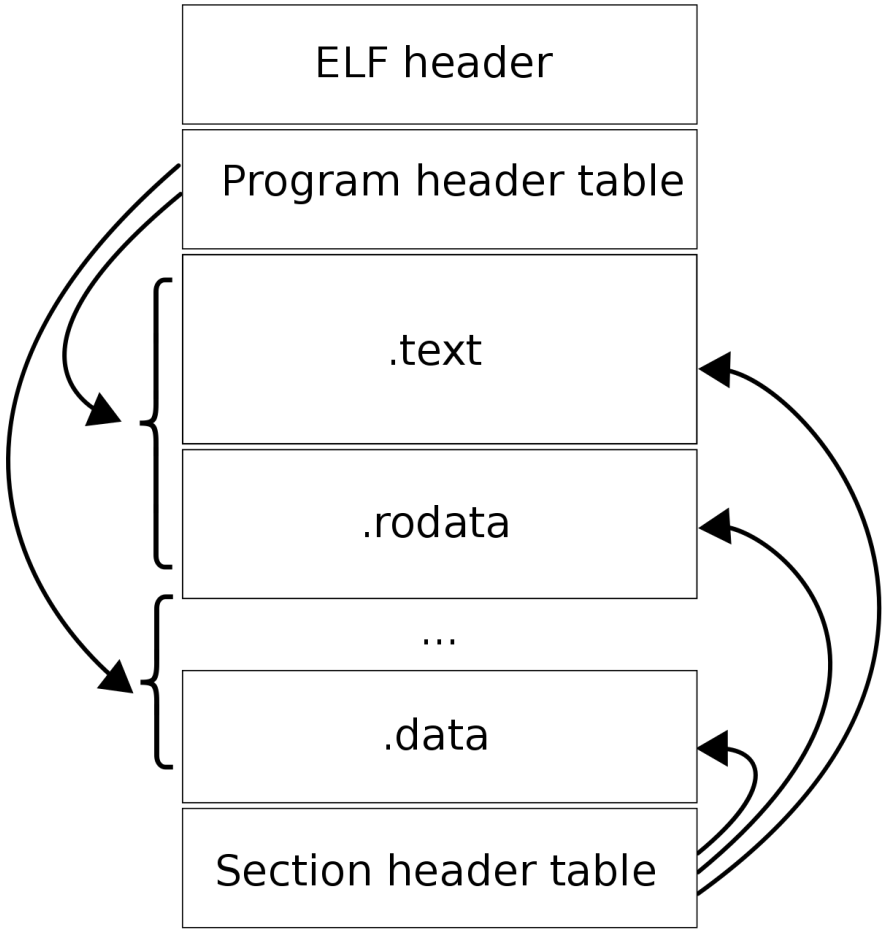


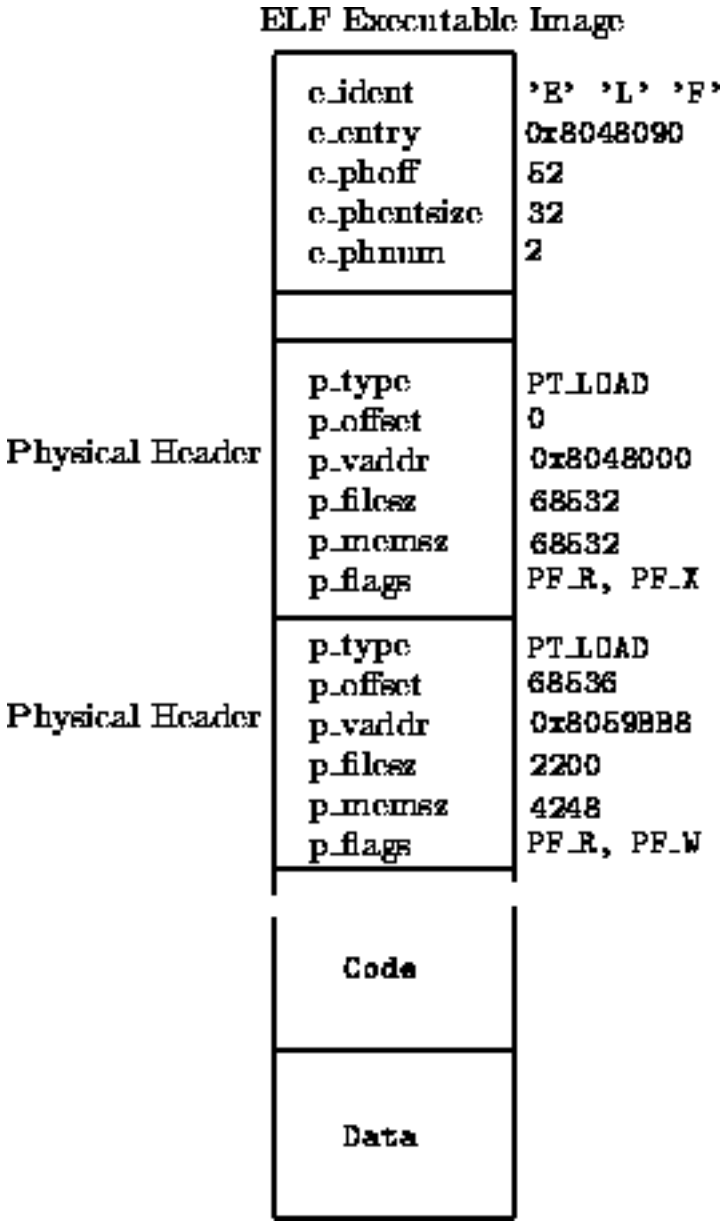
→ Стадия BDS
→ Системный раздел ESP (FAT32)
→ Основной загрузчик ОС (PE файл)
→ Ядро ОС

GPT



Формат ELF


ELF header
Program header table
.text
.rodata
...
.data
Section header table


ELF Executable Image
e_ident 'E' 'L' 'F'
e_entry 0x8048090
e_phoff 52
e_phentsize 32
e_phnum 2
Physical Header
p_type PT_LOAD
p_offset 0
p_vaddr 0x8048000
p_filesz 68532
p_memsz 68532
p_flags PF_R, PF_X
Physical Header
p_type PT_LOAD
p_offset 68536
p_vaddr 0x8059BB8
p_filesz 2200
p_memsz 4248
p_flags PF_R, PF_W
Code
Data

Нормативная литература

- ISO/IEC 9899 C Programming Language (2018)
- https://uefi.org/specifications
  UEFI Specification 2.8, UEFI PI Specification 1.7
- https://software.intel.com/en-us/articles/intel-sdm
  Intel® 64 and IA-32 Architectures Software Developer's Manual Combined Volume Set 1~4
- http://refspecs.linuxbase.org/elf/elf.pdf
  Executable and Linking Format (ELF) Specification 1.2
- Файлы с префиксом Lab1 на forge.ispras.ru
- Инструкции GIT, GDB, GNU Make, BASH.

Спасибо за внимание!

Вопросы?