LinuX-Lab 2018-05-11

Procedura startu systemu

Sprzętowo

Procesory `IA-32` i `x86_64`

Pierwszą czynnością po "zdjęciu" sygnału `reset` jest skok po adres `0xFFFF0` (16 bajtów poniżej 1MB). Procesor jest w trybie rzeczywistym, 16-bitowym.

W PC w tym miejscu znajduje się albo BIOS albo punkt wejściowy do UEFI.

Procesory `ARM32` w trybie surowym (Cortex-M/Cortex-R)

Wskaźnik stosu ustawiany jest na zawartość słowa z adresu 0, po czym procesor skacze pod adres wskazany pod adresem 4.

Procesory `ARM32/ARM64` w trybie pełnym (Cortex-A)

Wykonanie zaczyna się od bootloadera zaszytego w wewnętrznym ROM-ie/FLASH-u procesora (procesor skacze albo pod `0x0` albo pod `0xffff0000`, gdzie zmapowana jest pamięć bootloadera). Popularnym otwartoźródłowym bootloaderem jest u-Boot - można o tym pomyśleć jako o swoistym LILO/GRUB dla ARM.

Programowo

BIOS

Nie omawiamy tu CD/DVD w standardzie `El Torito`.

- BIOS inicjalizuje podstawowe komponenty płyty głównej (kontrolery pamięci, zegara procesora, DMA, przerwania, itp),
- Wykrywa karty rozszerzeń i jeżeli dana karta go posiada, wywołuje jej BIOS (karta grafiki przejmuje wtedy przerwanie 10h, kontrolery RAID przejmują przerwanie 13h, itd)
- BIOS przegląda urządzenia w kolejności bootowania i dla każdego z nich za pomocą przerwania 13h:
  - Wczytuje **pierwszy sektor** do pamięci, pod adres `0000:7C00`
  - Sprawdza, czy ostatnie dwa bajty sektora zawierają sygnaturę `0xAA55`

        * Jeżeli **tak**, wykonuje skok do pierwszego bajtu załadowanego sektora (od tego momentu kod użytkownika przejmuje całkowitą kontrolę)
        * Jeżeli **nie**, przechodzi do następnego urządzenia z listy
- Jeżeli nie udało się znaleźć sygnatury na żadnym urządzeniu, wyświetla błąd i zawiesza komputer.

Z poziomu programu użytkownika dostępne jest trochę usług, na przykład:

- `int 10h` - interfejs do karty graficznej
- `int 13h` - operacje dostępu do dysków po sektorach
- `int 14h` - operacje na porcie szeregowym
- `int 16h` - operacje na klawiaturze
- `int 17h` - operacje na drukarce
- `int 1Ah` - operacje na zegarze (RTC)
- `int 1Ah` - usługi PCI

UEFI

- UEFI inicjalizuje podstawowe komponenty płyty głównej (kontrolery pamięci, zegara procesora, DMA, przerwania, itp),
- Wykrywa karty rozszerzeń i jeżeli dana karta go posiada, wywołuje jej firmware i podpina go pod dostępną funkcjonalność.
- UEFI przegląda urządzenia w kolejności bootowania w `NVRAM` i dla każdego z nich za pomocą usług UEFI:
    - Sprawdza, czy dysk jest w formacie GPT
    - Szuka partycji ESP (*FAT32*, *bootowalna*, *systemowa* plus parę innych flag, nie musi być pierwsza)
        * Jeżeli nie skonfigurowano pamięci `NVRAM` inaczej, próbuje załadować następujący manager:
            · `\EFI\BOOT\BOOTx64.EFI` - PCty ze zwykłym procesorem 64-bitowym
            · `\EFI\BOOT\BOOTIA32.EFI` - niestandardowe, 32-bitowe UEFI w niektórych tabletach i 2-in-1. Ciekawostka - procesor tam **jest** 64 bitowy, niestandardowe UEFI ma tylko utrudnić instalację systemu innego, niż producenta (Windows, khm, khm).
            · `\EFI\BOOT\BOOTIA64.EFI` - komputery z procesorami Itanium
            · `\EFI\BOOT\BOOTARM.EFI` - komputery z 32-bitowymi procesorami ARM
            · `\EFI\BOOT\BOOTAA64.EFI` - komputery z 64-bitowymi procesorami ARM

    - Jeżeli `skonfigurowano` w pamięci `NVRAM` konkretny manager (ID urządzenia, partycja, ścieżka, plik), to jest on ładowany i uruchamiany. U mnie ten wpis wygląda tak: `HD(2,GPT,ea03e786-18b0-4277-80dd-a4f6e7f50935,0x1000,0x64000)/File(\EFI\DEBIAN\`
  - Jeżeli jest aktywny `SecureBoot`, UEFI weryfikuje podpis cyfrowy managera kluczem publicznym zaszytym **w sobie**. Jeżeli podpis się nie zgadza, albo go brakuje, to odmawia dalszej pracy.
- Jeżeli znaleziono manager i (opcjonalnie) przeszedł weryfikację, to jest uruchamiany.

Manager UEFI to aplikacja w formacie PE (Windowsa, khm, khm), która ma dostęp do szeregu usług dostarczanych przez firmware: grafika, system plików, stos sieciowy (!!!), kryptografia.

Do pisania własnych managerów w C można wykorzystać otwartoźródłowe SDK o nazwie `EDK II`, dostępne ze strony TianoCore.