8. PCI a USB, funkce a způsob implementace, ovladače PCI a USB zařízení pro OS Windows a Linux. (A4M38KRP)

PCI

- Peripheral Component Interconnect
- typy
    - šířka sběrnice 32 bitů, frekvence hodin 0 - 33 MHz, přenosová rychlost 132 MB/s
    - šířka sběrnice 32/64 bitů, frekvence hodin 66/33 MHz, přenosová rychlost 264 MB/s
    - šířka sběrnice 64 bitů, frekvence hodin 66 MHz, přenosová rychlost 528 MB/s
- napájení 3,3V nebo 5V - běžné PCI karty mají buď jeden nebo dva klíčovací zářezy podle napěťové signalizace - karty vyžadující 3,3 V mají zářez vedle přední strany karty, zatímco ty, vyžadující 5 voltů mají zářez na druhé straně, takzvané univerzální karty mají oba zářezy a mohou přijímat oba typy signálů.
- používá paralelní přenos dat
- od zbytku systému oddělena pomocí PCI mostů, které zprostředkovávají komunikaci s kartami
- možnosti připojení
    - planární zařízení - integrovaný obvod zabudovaný přímo do základní desky
    - rozšiřující karta

Přerušení

- o jedno přerušení se může dělit více karet - ovladače jsou pak volány v sérii - nevýhodou je vyšší latence a režie obsluhy přerušení
- obsahuje 4 linky přerušení
- jsou úrovňově spouštěné

Automatická konfigurace

1. karta je vsunuta do slotu vypnutého počítače
2. po zapnutí počítače je aktivována PnP část BIOSu
3. BIOS postupně vyzve všechna zařízení připojená ke sběrnici k identifikaci
4. zařízení odesílají své identifikátory a požadavky
5. BIOS přidělí níže uvedené systémové prostředky mezi připojená zařízení tak, aby nedošlo ke konfliktům:
6. přerušení
7. I/O porty
8. adresový prostor v paměti RAM (pro paměť na kartě)
9. údaje o konfiguraci jsou umístěna do paměti
10. je spuštěn operační systém
11. podle identifikace zařízení operační systém vyhledá ovladače
12. ovladače si přečtou konfiguraci svých zařízení a začnou je obsluhovat

- Konfigurace jednotlivých zařízení je uložena v registrech PCI sběrnice (ESCD – Extended System Configuration Data), která mohou být použita při dalším startu počítače. Uživatel v některých verzích BIOSů může ručně vynutit novou kompletní inicializaci pomocí vymazání ESCD. BIOSy, které podporují ACPI, ukládají do speciálních tabulek mnohem více informací.
- Celý proces automatické konfigurace funguje spíše jako černá skříňka, takže uživatel má obvykle minimální možnosti, jak jej ovlivnit. Ze stejného důvodu není výsledek automatické konfigurace předvídatelný ani u podobných zařízení.

USB

- umožňuje připojit až 127 zařízení
- automatická konfigurace hned po připojení
- řeší napájení zařízení
- Host => Hubs => USB zařízení
- Host
    - odesílá a přijímá data
    - konvertuje paralelní data na sériová a opačně
    - proces enumerace
    - správa napájení
    - OHCI (Open Host Controller Interface) / UHCI (Universal ...) - Intel
- Hub
    - distribuuje data
    - spravuje napájení svých portů
    - reportuje stav svých portů
    - detekce full/low rychlosti
    - mohou mít vlastní napájení nebo napájení přes sběrnici
    - vždy jsou alespoň full speed
- USB zařízení
    - připojené do HUBu
    - low speed 1,5 Mbit/s - full speed 12 Mbit/s (USB 1.1) - high speed 480 Mbit/s (USB 2.0)
    - vlastní napájení / napájení ze sběrnice - low powered I_{max} = 100 mA, hign powered I_{max} = 500 mA
    - podpora plug & play
    - kabel pro low speed - ne/stíněná ne/kroucená dvoulinka, maximálně 3m
    - kabel pro high speed - stíněná a kroucená dvoulinka maximálně 5 m
- USB zařízení může mít 16 endpointů - EP0 je vždy obousměrný, EP1 až EP15 může být vstupní nebo výstupní - maximálně tedy 31 rour
- Typy přenosů
    - Control Data - konfigurace a řízení - vždy přes EP0
    - Interrupt Data - pro malá data, myš, klávesnice, ...
    - Isochronous Data - audio a video přenosy, pouze pro full speed a výše
    - Bulk Data - velká data, tiskárny, skenery, disky..., pouze full speed a výše
- Proces enumerace
    1. Hub informuje hostitelský počítač o tom, že bylo připojeno nové zařízení.
    2. Hostitelský počítač se dotáže hubu, na který port bylo zařízení připojeno.

3. Po doručení odpovědi vydá počítač příkaz tento port zapnout a provést vynulování (reset) sběrnice.
4. Hub vyrobí nulovací signál (reset) o délce 10 ms. Uvolní pro zařízení napájecí proud 100 mA. Zařízení je nyní připraveno a odpovídá na implicitní (default) adrese.
5. Než zařízení USB obdrží svou vlastní adresu sběrnice, je možno se na ně obracet přes implicitní adresu 0. Hostitel si přečte první bajty popisovači zařízení, aby stanovil, jakou délku mohou mít datové pakety.
6. Hostitel přiřadí zařízení jeho adresu na sběrnici.
7. Hostitel si ze zařízení pod novou sběrnicovou adresou načte všechny konfigurační informace.
8. Hostitel přiřadí zařízení jednu z možných konfigurací. Zařízení nyní může odebírat tolik proudu, kolik je uvedeno v jeho popisovači. Tím je připraveno k použití. Hostitel přiřadí zařízení jeho adresu na sběrnici.

Ovladače

Windows

Typy ovladačů:

- Virtual device drivers (VDD) - user mode ovladač, virtualizace HW pro DOS aplikace
- Kernel-mode drivers
    - File system drivers
    - PnP drivers
        - WDM drivers
            - Class drivers - pro zařízení s podobnou funkcionalitou (např. HID), podporuje společné rysy
            - Mini-drivers - podporuje specifické vlastnosti nepodporované class ovladačem
            - Monolithic function drivers - standardní ovladač konkrétního zařízení
            - Filter drivers - umožňuje modifikovat funkce standardního ovladače
    - Legacy drivers - pro zařízení nepodporující PnP, typicky se nahrává při startu systému

Driver Entry:

- WDM Driver inicializuje Driver Object a končí
- Ne WDM ovladač navíc
    - detekuje HW
    - vytváří device object
    - dokončuje konfiguraci
- Inicializace WDM Driver objektu
    - adresa Driver Unload
    - adresa AddDevice
    - adresy pro zpracování PnP, Power a SystemControl žádostí

Driver Unload:

- Úklid po Driver Entry

- např. uvolnění paměti apod.

Linux

- module_init(ourmouse_init); module_exit(cleanup_module);

```
struct file_operations our_mouse_fops = {
  owner: THIS_MODULE, /* Automatic usage management */
  read: read_mouse, /* You can read a mouse */
  write: write_mouse, /* This won't do a lot */
  poll: poll_mouse, /* Poll */
  open: open_mouse, /* Called on open */
  release: close_mouse, /* Called on close */
};
```