

ZÁKLADY POČÍTAČOVÝCH SYSTÉMŮ



LETNÝ SEMESTER

2019/2020

Obsah

§ 3 Úvod do sietí	1
§ 3.1 Spôsoby pripojenia.....	1
§ 3.2 Kapacita siete	1
§ 3.3 Server.....	1
§ 3.4 Paket	1
§ 3.5 Megabajt vs Megabit	1
§ 3.6 Rozdelenie siete podľa veľkosti.....	2
§ 3.7 Model OSI	2
§ 3.8 Internet Protocol (IP)	2
§ 3.9 Počítanie IPv4 adres	3
§ 3.9.1 Príklad	3
 § 4 Základná doska, zdroj	4
§ 4.1 Základná doska.....	4
§ 4.2 Počítačový zdroj.....	5
 § 5 USB PCL PCe.....	6
§ 5.1 USB 2 (Universal Serial Bus)	6
§ 5.1.1 USB proces.....	6
§ 5.2 USB 3.....	7
§ 5.3 USB C.....	7
§ 5.4 Krútená dvojlinka (Twisted Two Pair).....	7
§ 5.5 Round-Trip Time (RTT)	7
§ 5.6 PCI zbernica (Peripheral Component Interconnect).....	8
§ 5.6.1 PCI-X (Peripheral Component Interconnect eXtended)	8
§ 5.6.2 PCIe (Peripheral Component Interconnect Express)	8
§ 5.7 Accelerated Graphics Port	8

§ 6 Procesor, chladenie	9
§ 6.1 Chladenie procesora	9
§ 6.1.1 Heat pipe.....	9
§ 6.2 CPU (Centra Processing Unit).....	9
§ 6.2.1 Mikroprocesor	10
§ 6.2.2 Dual Core verzus Dual procesorový systém	10
§ 6.3 Kódy na detekciu chýb pri prenose dát.....	11
 § 7 Skriptovanie	 12
 § 8 Počítačová architektúra	 15
§ 8.1 North Bridge a South Bridge	15
§ 8.2 Von Neumannova a Harvardská architektúra počítača	15
§ 8.2.1 Radič (controller)	16
§ 8.2.2 Aritmeticko-logická jednotka (ALU).....	16
§ 8.2.3 Pamäť počítača – Von Neumann.....	16

§ 3 Úvod do sietí

- (1) **Access point**: miesto na ktoré sa pripája (Access point môže byť router napríklad)
- (2) **Smerovač** (switch): zariadenie na smerovanie toku dát (niečo ako predlžovačka akurát pre internet)

§ 3.1 Spôsoby pripojenia

- (1) drôtový kábel
- (2) bezdrôtová WiFi
- (3) satelitná 3G, 4G, ..

§ 3.2 Kapacita siete

- (1) maximálny prenosový potenciál. Udáva sa ako download/upload

§ 3.3 Server

- (1) doslova **počítač** ktorý je stavaný na neustálu **prevádzku na sieti**. Sú na nej uložené stránky, aplikácie a podobne..

§ 3.4 Paket

- (1) po sieti sa data **posielajú** vo forme paketu
- (2) je **výhodnejšie** rozdeliť jeden súbor na viac malých a poselať ho akoby po častiach
- (3) výhodou je, že keby je poslaný paket chybný (čo sa môže stať úplne bežne) nič sa nedeje, paket sa pošle znova a keďže má minimálnu veľkosť, tak si to užívateľ ani nevšimne
- (4) ak by si teda posielal veľký súbor z vnútornej siete, tak by si zabil celú linku, takto sa dátový tok rozdelí na pakety a do vonkajšej siete prebiehajú postupne
- (5) obr. = vnútorné rozdelenie paketu

Header	Sender's IP address Receiver's IP address Protocol Packet number	96 bits
Payload	Data	896 bits
Trailer	Data to show end of packet Error correction	32 bits

§ 3.5 Megabajt vs Megabit

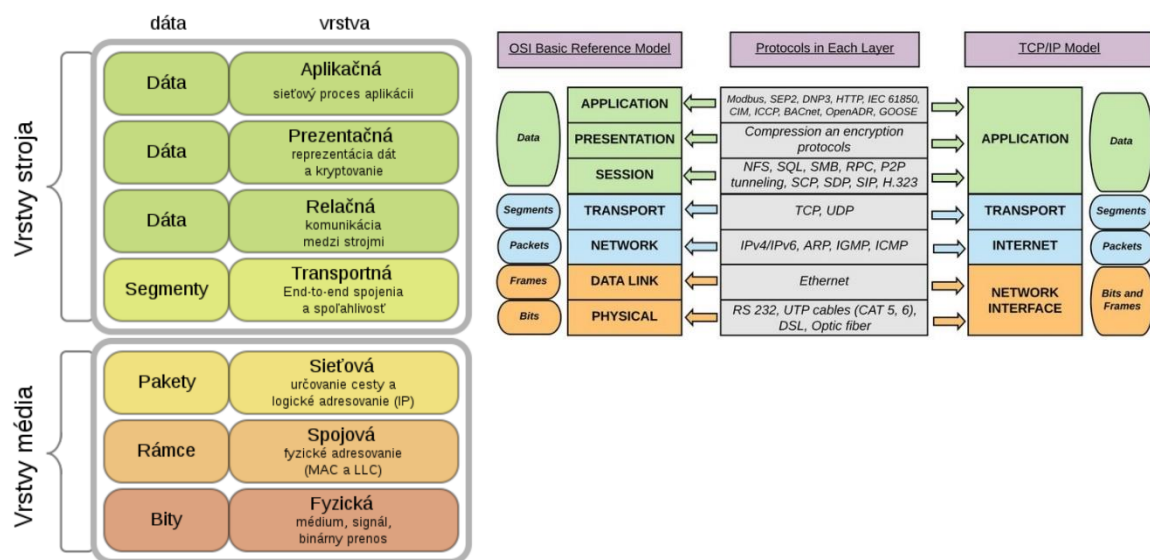
- (1) Megabajt je MB (v tom sa väčšinou udáva veľkosť sťahovania)
- (2) vzťah medzi týmito dvoma jednotkami je veľmi podstatný pre dorozumenie v informatike pretože tento pojem je často zamieňaný
- (3) 1 megabajt (MB) = 8 megabitov (Mb)

§ 3.6 Rozdelenie sieti podľa veľkosti

- (1) **PAN** ⇒ Personálna, keď si pripojíš slúchadla na telefón
- (2) **LAN** ⇒ Lokálna, proste LANka napríklad viac zosietovaných notebookov v jednej miestnosti
- (3) **MAN** ⇒ Metropolitná (prepojenie mesta)
- (4) **WAN** ⇒ Svetová sieť

§ 3.7 Model OSI

- (1) logické rozdelenie rôznych vrstiev sieti
- (2) základný a rozšírený model OSI



§ 3.8 Internet Protocol (IP)

- (1) základom IP protokolu je IP adresa
 - a) **IPv4** - 192.168.131.205, 32-bitové číslo – v podstate existuje viac než 4 miliardy (4 294 967 296) možných adries.
 - b) **IPv6** – rozšírenie IPv4, vzniklo preto, lebo počet možných IPv4 adries už nie je dostačujúci

§ 3.9 Počítanie IPv4 adries

- (1) dekadický tvar IP adresy, je možné jednoducho zapísať v binárnom tvare (1 a 0)
- (2) postup krokov pre určenie druhov IP adries :
- (3) **adresa siete** (pokiaľ nemám zadané)
 - a) zvolím si IP počítača, ktorý je v danej sieti
 - b) zistím Masku: 32b číslo
 - c) spravím binárny súčin týchto dvoch adries
- (4) **maska** je jednoduchá adresa, ktorá sa skladá z jednotiek, ktoré idú z ľava doprava.. ak ich raz preruší nulový bit, viac jednotiek už nenasleduje
- (5) maska sa bežne zapisuje **dekadicky** (v desiatkovej sústave) $\Rightarrow /22, /18, \dots$ atd \Rightarrow toto číslo nám udáva počet bitov, ktoré sú aktívne \Rightarrow stav 1

§ 3.9.1 Príklad

- adresa PC: 192.168.16.33
- jednotlivé oktety :

192: 1100 0000

168: 1010 0000

16: 0001 0000

33: 0010 0001

- maska / 18 :

255. 192. 0

1111 1111 1111 1111 1100 0000 0000 0000

- adresu siete teda vypočítame ako **logický resp. binárny súčin týchto dvoch adries** (PC & Maska):

1100 0000 1010 0000 0001 0000 0010 0001 (Adresa PC v binárnom tvare)

1111 1111 1111 1111 1100 0000 0000 0000 (Maska v binárnom tvare)

1100 0000 1010 0000 0000 0000 0000 0000

192. 168. 0. 0 /18

\Rightarrow je finálna adresa siete s maskou /18

§ 4 Základná doska, zdroj

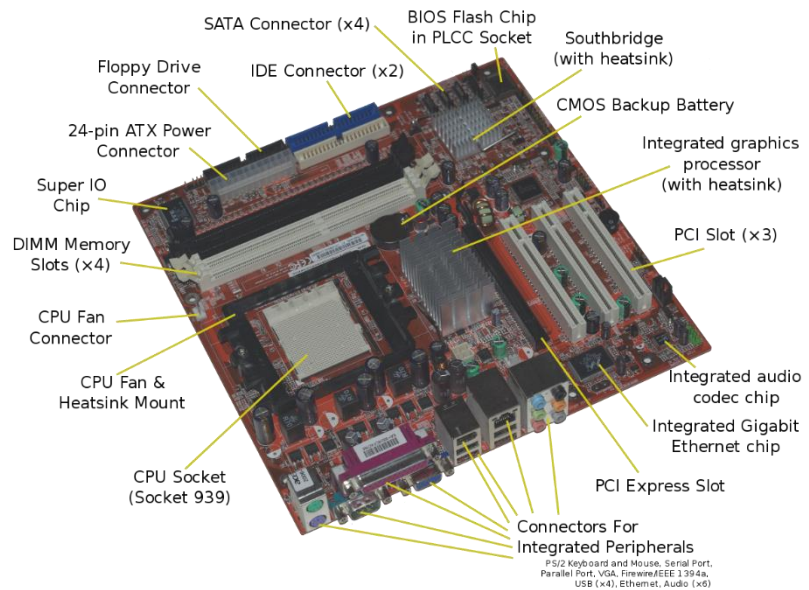
§ 4.1 Základná doska

(1) má určitý **formát**

- a) ATX
- b) Micro-ATX
- c) Mini-ITX

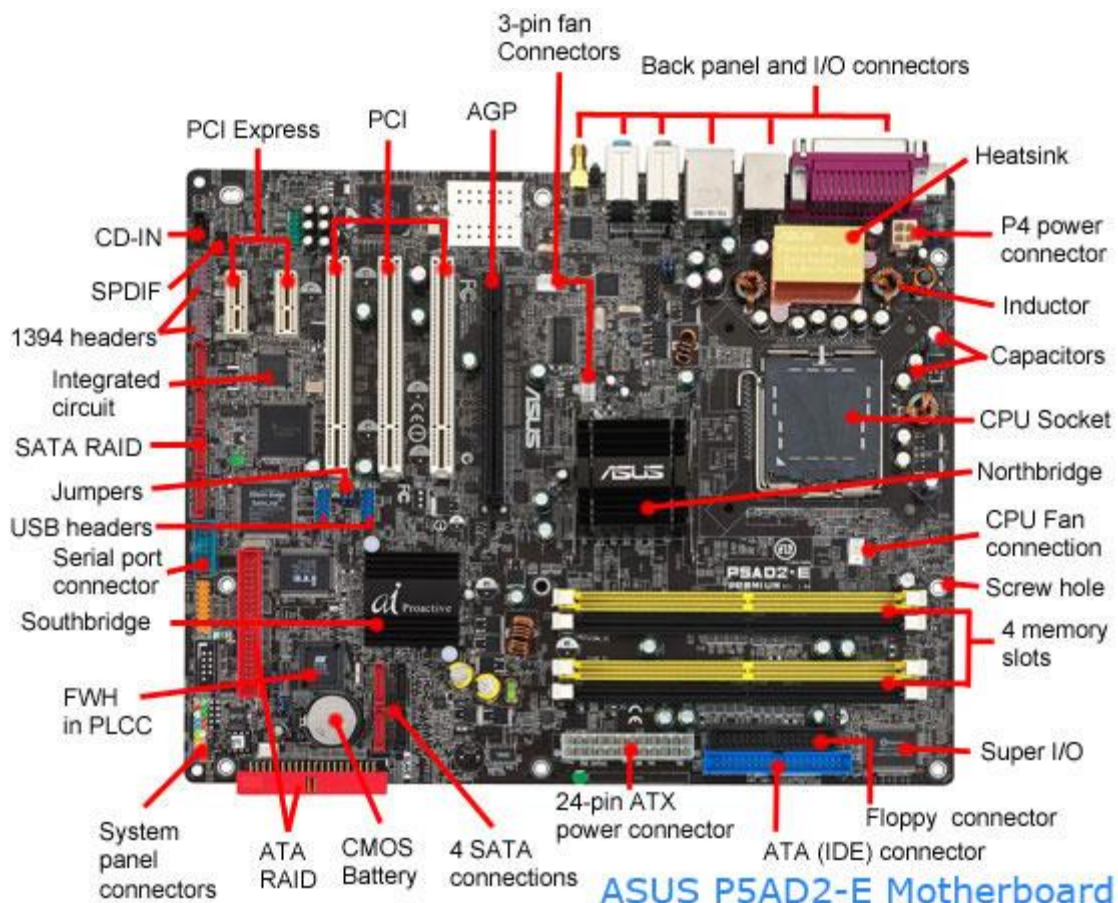
(2) **definuje**

- a) výkonové možnosti počítača
- b) možnosti rozšíriteľnosti
- c) konektivitu počítača (typ procesora)



(3) skladá sa z rôznych **častí**

- a) napájací konektor
- b) socket procesora
- c) kondenzátory (také valčeky) a cievky, sú súčasťou obvodov ktoré umožňujú zmenu napájacieho napätia pre procesor -> pretaktovanie
- d) chladenie
- e) doplnkový napájací port
- f) konektor na ventilátor procesora (4 piny)
- g) back-panelové konektory (USB, audio, internetová sieť...)
- h) sloty pre pripojenie RAM pamäte
- i) AGP slot
- j) PATA konektory na pripojenie starých harddiskov (disketová mechanika..)
- k) SATA konektory na pripojenie novších harddiskov
- l) PCI a PCIe sloty na pripojenie rôznych kariet (grafická, sieťová, zvuková..)
- m) piny na pripojenie LED diód a tlačidiel
- n) baterka na udržanie nastavení BIOSu
- o) south bridge a north bridge, slúžia ako interface smerom k procesoru



§ 4.2 Počítačový zdroj

- (1) **konvertor** striedavého napätia na **jednosmerné** (3.3, 5 a 12V)
- (2) notebookový zdroj dodáva len jeden druh napätia, napr. 19V
- (3) na zadnom paneli má ventilátor, vypínač a konektor na napájací kábel – tieto časti vidno
- (4) z druhej strany doňho vedú zväzky káblov
 - a. **žltý** - 12V
 - b. **čierny** - zem
 - c. **oranžový** - 3.3V
 - d. **červený** - 5V
- (5) konektor sa vkladá do základnej dosky, dá sa vložiť len jedným smerom
- (6) doska môže mať 20 alebo 24 pinový **napájací konektor**, a tak isto zdroj môže mať tiež 20 alebo 24 pinový konektor – všetky možnosti zapojenia spolu fungujú
- (7) **4 pinový káblík** zo zdroja s dvoma žltými a dvoma čiernymi káblíkmi k procesoru vie priniesť dodatočných 24V (pre náročnejšie aplikácie)
- (8) napájací **konektor pre ventilátor** – farby káblov a napätia sa nezhodujú, má 4 piny: napájanie, zem, tachometrický signál (informácie o otáčkach ventilátora) a kontrolný pin (zmena otáčok)

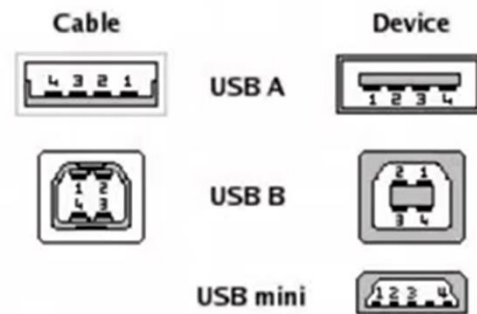


§ 5 USB PCL PCe

§ 5.1 USB 2 (Universal Serial Bus)

(1) 4 konektory (4 piny)

- a) pin 1: **červený**, 5V
- b) pin 2: **biely**, data –
- c) pin 3: **zelený**, data +
- d) pin 4: **čierny**, uzemnenie



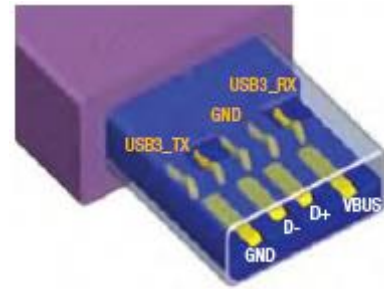
- (2) napr. typy USB A, USB B (pripojenie tlačiarne), USB 2 (obojsmerné, na vysokorýchlostné prepojenie dvoch počítačov, má v strede konvertor)
- (3) univerzálna zbernica, USB štandard podporuje až 127 zariadení (až v 7 úrovniach USB hub-ov)
- (4) USB hub (s napájaním / bez)

§ 5.1.1 USB proces

- (1) **štart** USB hostiteľa (host)
- (2) **zistenie** pripojených USB zariadení
- (3) pripojeným zariadeniam host priradí **adresy** (enumeration)
- (4) zistenie hostom aké **typy data transferov** tieto zariadenia vyžadujú
 - a) **control** – konfigurácia zariadenia po jeho pripojení
 - b) **interrupt** – typ prenosu, kedy sú prenášané malé údaje s dôrazom na skoré doručenie (myška, klávesnica..)
 - c) **bulk** – prenos čo najväčšieho množstva dát čo najrýchlejšie (typický prenos, napr. tlačiareň)
 - d) **isochronous** – prenos dát citlivých na zachovanie časovania (realtime video z kamery)
- (5) zisťovanie **požiadaviek** na šírku prenosového pásma
- (6) pri požiadavkách vyšších než je 90% pásma celého USB prenosu, host **odmietne** obsluhu ďalších zariadení

§ 5.2 USB 3

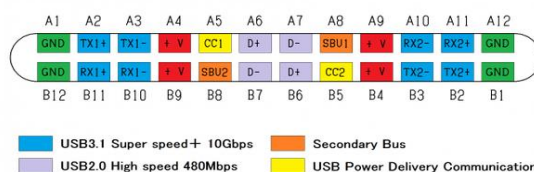
- (1) má akúsi nadstavbu nad 4 pinmi (oproti USB 2) – **5 pinov**
- (2) má (preto) oveľa vyššiu rýchlosť prenosu
- (3) **väčšia šírka pásma**, používa dve jednosmerné cesty, ktoré fungujú zvlášť (dva piny pre prijímanie a dva pre odosielanie)
- (4) USB 2 ide zapojiť do USB 3 konektoru (spätná kompatibilita)
- (5) tok 8 bitov zakódovaných v 10 b (zabezpečenie bezchybnosti prenosu – na detekciu chýb sa využívajú rôzne kódy, napr. **paritný kód** alebo **check-sum**)



§ 5.3 USB C

- (1) obojstranný konektor
- (2) piny sú ako keby duálne
- (3) obojsmerné napájanie

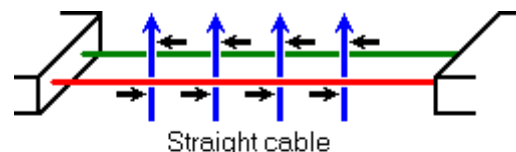
USB Type-C Connector Pin Assign



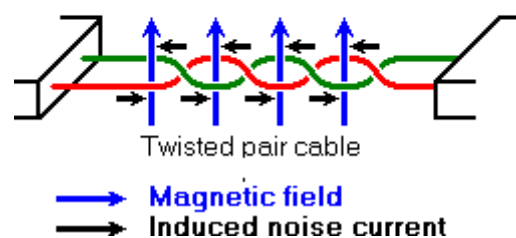
§ 5.4 Krútená dvojlinka (Twisted Two Pair)

- (1) ekonomicky náročnejšie, ale menej náchylný na rušenie z vonka

- (2) keď pôsobí nejaké elektrické rušenie na **paralelný pár**, na jeden drôt pôsobí viac a na dlhej vzdialenosti sa tam naindukujú veľký rozdiel medzi nimi – môže to pôsobiť ako rušivá informácia pri čítaní

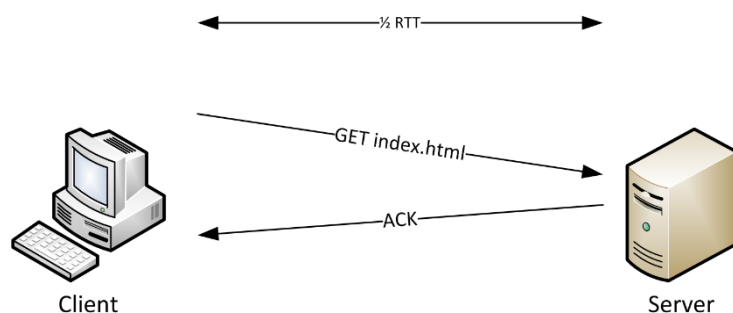


- (3) oproti tomu na **krútenej dvojlinke** sa striedajú, a teda sa na nich indukuje striedavo a na konci medzi nimi nie je rozdiel a nenachádza sa tu (taká) rušivá informácia



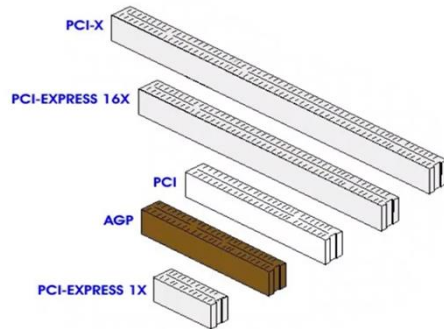
§ 5.5 Round-Trip Time (RTT)

- (1) Máme dva uzly, A a B – uzol **A** vyšle informáciu do uzla B, a trvá to nejaký čas. Potom v uzle **B** prechádza nejaké **spracovanie**, vytvorí odpoveď, a tú potom pošle uzlu A. Čas, za ktorý prebehne celý tento prenos sa nazýva **round-trip time**.
- (2) pre výpočet sa využíva vzorec $s = v * t$ (dráha= rýchlosť * čas)



§ 5.6 PCI zbernica (Peripheral Component Interconnect)

- (1) starší typ zbernice, ktorý sa už dnes nepoužíva
- (2) **zdieľaný typ** zbernice
- (3) šírka 32 bitov (32 vodičov nad sebou prepája procesor s ostatnými komponentami – nie jedným, lebo je zdieľaná)
- (4) prístup na zbernicu riadi **master** – určuje, kto bude kedy vysielat dáta
 - a) k zbernici pristupuje iba jednosmerne – **first party DMA** – masterom sa stáva niektoré zo zariadení (procesor, grafická karta...)
 - b) existuje aj **third party DMA** – prístup riadi špecializovaný DMA controller
- (5) **frekvencia** sa prispôsobuje najpomalšiemu zariadeniu, ktoré je do zbernice zapojené (aby spolu vedeli komunikovať)
- (6) nezávisí, aký typ **procesora** je pripojený (AMD, Intel..) – podporuje štandardizovaný formát



§ 5.6.1 PCI-X (Peripheral Component Interconnect eXtended)

- (1) serverové aplikácia, aplikácia náročné na šírku pásma
- (2) väčšia dĺžka oproti PCI a 4x rýchlejšia

§ 5.6.2 PCIe (Peripheral Component Interconnect Express)

- (1) nie je kompatibilná s PCI zbernicou
- (2) nie je zdieľaná – podporuje full-duplex komunikáciu medzi dvomi koncovými bodmi bez závislosti na zariadeniach pripojených do iných PCIe slotov
 - a) **simplex** (napr. autorádio) vysielateľ vysielá smerom k prijímaču signál, je to jednosmerný kanál
 - b) **duplex** je obojsmerný prenos informácie (dva počítače v sieti)
 - c) **full-duplex** - zariadenia môžu naraz odosielať aj prijímať dáta
 - d) **half-duplex** – zariadenia sú prepojené, ale jedno vždy počúva a druhé prijíma, neodosielajú odrazu
- (3) vodiče teda nie sú paralelné, ale sériové
- (4) 4 piny pre jednu signálovú linku
- (5) point-to-point spojenie paketového charakteru – jedno zariadenie je pripojené práve na jednu linku
- (6) nahrádza PCI, PCI-X aj AGP, má vyššiu dátovú priepustnosť
- (7) viacero generácií (Gen 2, Gen 3..) a fyzických prevedení (x1, x4, x8, x16..)

§ 5.7 Accelerated Graphics Port

- (1) ďalší typ zbernice
- (2) slúžil výhradne na pripojenie grafickej karty ku základnej doske
- (3) vznikol pre potreby 3D grafiky – potreba prenosu vysokých dátových tokov
- (4) nezdieľa pripojenie procesoru s ďalšími zariadeniami
- (5) adresovacia a dátová časť zbernice sú oddelené



§ 6 Procesor, chladenie

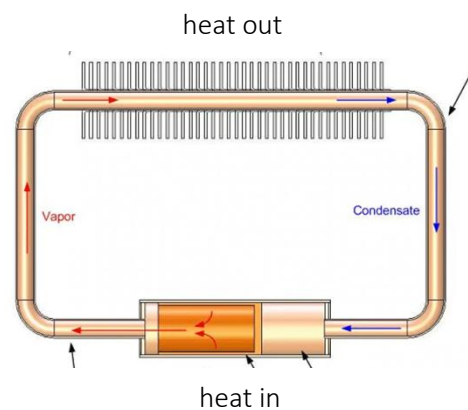
§ 6.1 Chladenie procesora

- (1) **aktívna** súčasť – chladič na vrchu, ktorý sa prikladá na procesor, pripína sa k základnej doske **sponou**
- (2) chladiť sa dá aj inak, napríklad tekutým dusíkom, olejom (nie priamo vodou – jedine v rúrkach)
- (3) dôležité je pripevnenie procesora k základnej doske (správny spôsob – nie napr. tavnou pištolou)



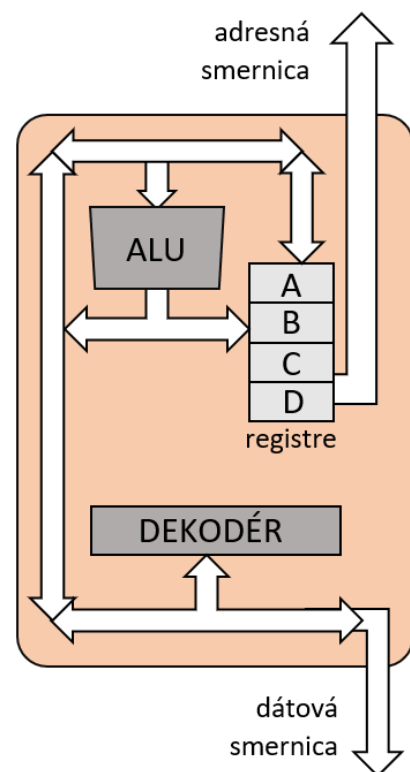
§ 6.1.1 Heat pipe

- (1) **medené** rúrky, ktoré odvádzajú teplo
- (2) kvapalina na spodnej časti zbiera teplo, kým sa nedostane do stavu že sa vyparuje – premieňa sa na **paru**, cestuje do chladiča, teplo sa uvoľňuje a para kondenzuje na vodu a vracia sa do nádoby, kde zbiera teplo = okruh
- (3) v trúbkách môže byť ako médium aj napr. amoniak, alkohol alebo etanol



§ 6.2 CPU (Centra Processing Unit)

- (1) **procesor** = mozog počítača
- (2) výpočtový stroj, ktorý na základe inštrukcii (a operandov) vykonáva operácie
- (3) inštrukcie sú vo forme **numerických strojových kódov**
- (4) stavba a funkcionality
 - a) **blackbox** pripojený na adresnú **smernicu** a dátovú smernicu
 - b) po **adresnej smernici** behajú adresy, po **dátovej smernici** dáta
 - c) dáta vstúpia do procesora (vstúpia dve operandy a inštrukcie čo sa s nimi má stať)
 - d) inštrukčný **dekodér** ich dekoduje a dáva informáciu ALU, čo má s operandami spraviť
 - e) **registre** sú také mikro pamäte a napríklad načítajú operandy, zapisujú sa do nich výsledky
 - f) tie potom idú na dátovú smernicu a zapisujú sa do pamäte
- (5) procesor vo všeobecnosti je pripojený na **adresnú zbernici**, **dátovú zbernici** a nachádza sa tu aj **kontrolná zbernica** – takto komunikuje s pamäťou, vstupnými a výstupnými obvodmi.. = jednoduchý model počítača (**dôležité!**)



§ 6.2.1 Mikroprocesor

- (1) zminiaturizované CPU (CPU môže byť aj veľmi veľké)
- (2) vznikol spojením ALU a radiča do jedného čipu
- (3) skladá sa z viacerých funkčných blokov (control unit, arithmetic logic unit, registre)
- (4) existujú 4 základné architektúry
 - a) **CISC** (Complex Instruction Set Computer) – má **komplexný** set inštrukcií s **mnohými** módmi adresovania, používa **separovanú** mikroprogramovaciu jednotku s kontrolnou pamäťou na implementovanie týchto komplexných inštrukcií, je to **jednoduchý** kompilovací dizajn, výpočty sú **pomalšie** ale **presné**, dekódovanie inštrukcií je **náročné** a exekučný čas je **dlhý**
 - b) **RISC** (Reduced Instruction Set Computer) – **neobsahuje** všetky inštrukcie (špecializovaný na niečo konkrétne) a má menej módov adresovania, je to **komplexný** kompilovací dizajn, výpočty sú **kratšie** a **presné**, dekódovanie je jednoduché a exekučný čas je **krátky**
 - c) **MISC** (Minimum Instruction Set Computer) – snaha o používanie inštrukcií bez operandov (s minimom), má nižšie nároky na rýchlosť pamäte
 - d) **VLIW** (Very Long Instruction Word) – práva naopak pracujú s veľmi dlhými inštrukciami a majú vysoké nároky na pamäť

§ 6.2.2 Dual Core verzus Dual procesorový systém

- (1) Dual procesorový systém má **dva samostatné CPU** čipy (procesory)
- (2) Dual Core procesor je procesor, v ktorom sú **dve výpočtové jadrá** na **jednom CPU** čipe
- (3) tak isto to funguje s **Quad** procesorovým systémom a Quad Core procesorom
- (4) nie všetky programy vedia využívať dvoj a viac procesorové architektúry
- (5) **orientáciu procesora** v päťici je možné určiť značkou na procesore, výrezom, chýbajúcimi nožičkami...
- (6) dakedy sa procesor do dosky vtláčal, potom vznikli také, ktoré sa zaťahovali páčkami a tie je možné odobrať (ľahšie)

§ 6.3 Kódy na detekciu chýb pri prenose dát

(1) **paritný kód** (1001 1100 -> 1001 1100 0)

- a) je to jednoduchý kontrolný súčet určený pre ochranu integrity jedného dátového slova (obvykle 8-bitov, t. j. 1 byte)
- b) parita je obvykle jeden bit, ktorý sa pripája k dátovému slovu, a vyjadruje, či je počet logických jednotiek v dátovom slove párný alebo nepárny
- c) **môže byť teda párna alebo nepárna parita** – k užitočnému kódu pridáme jednu paritu tak, aby celkový počet jednotiek bol párný
- d) **spočítame počet jednotiek**, ak je párný pridáme **nulu**, v nepárnej parite pridáme **jednotku**
- e) chyby nedokáže opraviť, ale najprimitívnejší možný kód vie **detekovať jednu chybu** (párny / nepárny počet)
- f) napríklad vysielateľ aj prijímač vedie, že sa **používa nepárna parita**, čiže keď príde slovo s **párnym** počtom jednotiek vieme, že sa niekde stala **chyba** – vyžiadanie informácia nanovo

(2) ďalší typ kódu používa **10 bitové slová**, a tento vie opraviť až dve chyby

- a) systém vie, aké používa slová: 00000 00000, 00000 11111, 11111 00000, 11111 11111
- b) keď mu príde pokazené slovo, napr. 11000 00000, vie určiť, na ktoré správne sa najviac podobá

(3) **chceck-sum**

- a) prepočet podľa hodnôt ASCII kódu (ako hash)

§ 7 Skriptovanie

1. výpis používateľov v systéme

- rozdelenie výpisu podľa -d **delimitera** (nejaké znamienko)
- určenie správnej hodnoty (f1 : f2 : f3...)
- | spájanie príkazov

```
> cat /etc/passwd | cut -d":" -f1
```

2. výpis zoznamu používateľských skupín

```
> cat/etc/group | cut -d":"-f1
```

3. výpis process ID procesov zadaného používateľa

- výpis len riadkov, ktoré obsahujú slovo debian (napr.)

```
> ps -aux | grep "debian"
```

- zlúčenie medzier dokopy aby ich delimiter správne chápal

```
> ps -aux | grep "debian" | tr -s " "
```

- vyselektovanie 1 a 2 stĺpca podľa medzery

```
> ps -aux | grep "debian" | tr -s " " | cut -d" " -f1,2
```

- keď to vypisuje ešte niečo čo by nemalo (napr. root)

```
> ps -aux | grep "debian" | tr -s " " | cut -d" " -f1,2 | grep -v "root"
```

4. výpis názvov súborov v adresári s nejakými určitými oprávneniami

- práva = read, write, execute
- _____ (-rwerwerwe)
- _ (typ: d = adresár, pomlčka = súbor, l = odkaz) ___ (vlastník) ___ (skupina) ___ (vš. ostatní)
- ak chceme vyselektovať w práva skupiny w bude na 6 pozícií => "^.....w" (bodky = hocikajký znak)

```
> ls -al | grep "^.....w"
```

- zjednotíme medzery aby sme mohli použiť cut

```
> ls -al | grep "^.....w" | tr -s " "
```

- vyberieme správny stĺpec

```
> ls -al | grep "^.....w" | tr -s " " | cut -d" " -f9
```


5. výpis textu odkazu nejakej stránky TO CO CHCEME

- stiahneme html z webu a dáme ho vypísať, vyberieme <a href=" "

```
> wget -O- stranka.sk | grep "a href"
```

- odfiltrujeme html (TO CO CHCEME)

```
> wget -O- stranka.sk | grep "a href" | cut -d">" -f2
```

- zbavíme sa poslednej zátvorky čo zostala (slovo)

```
> wget -O- stranka.sk | grep "a href" | cut -d">" -f2 | cut -d"<" -f1
```

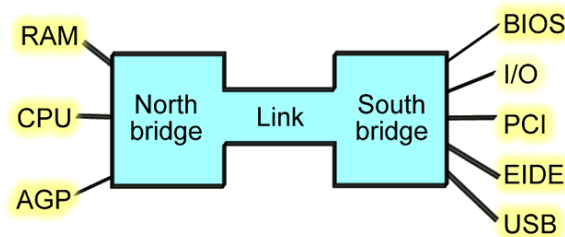
- alebo najlepšia metóda s regulárnym výrazom, .* reprezentuje hocikolko hocijakých znakov, teda nájdeme presne len linky

```
> wget -O- stranka.sk | grep -Eo "a href=.*>.*</a>" | cut -d">" -f2 | cut -d"<" -f1
```

	zápis	názov	popis	príklad		
C	cat	výpis obsahu súboru		cat /etc/passwd		
	cd	change directory				
	cut	cut		výpis len určitých znakov	cut -d">" -f2 cut -d"<" -f1	
		-d" "	-d "delimiter" -f(cislo)			
G	grep	hľadanie stringu, napríklad pre výpis		grep "^.znak"		
		-w	úplná zhoda (pomaranc != pomarance)			
		-i	ignorovanie rozdielov veľké/malé písmená, atď..			
		-v	na nezahrnutie slova (grep "slovo1" grep -v "slovo2")			
		-Eo	regulárny výraz (?)			
	groupadd	pridaj skupinu		grep -Eo "a href=.*>.*		
	groupdel	-f	force (?)			
	groups	výpis skupín v systéme				
CH	chmod	zmena oprávnení súboru		chmod g+rw		
	chown	change ownership	zmeniť vlastníctvo	sudo chown (povodny vlastnik): (novy) (sabor)		
I	ls	list directory content		výpis obsahu adresáru	ls /(priecinok)	
		-la	zoznam vš. súborov, ktoré sa nachádzajú v danom adresári (kombinácia -l + -a)			
		-lh	zrozumiteľné čítanie			
		-a	aj skryté súbory			
		-l	súbory, priečinky, veľkosť, práva, majiteľ			
M	man	manual	manuál			
	mkdir	make directory	vytvorenie priečinku			
N	nano	textový editor v linuxe				
P	passwd	zmeň heslo		sudo passwd (uzivatel) ----> spyta sa na heslo		
	ps	report current processes		vypíše momentálne spustené procesy		
		-e	všetky procesy štandardným syntaxom			
		-aux	aj používateľov a command			
	pwd	vypíše priečinok, v ktorom sa človek nachádza, aj s obsahom				
R	rm	-rf	vynútné vymazanie priečinka aj s obsahom			
S	stat	štatistika o súbore (počet slov..)				
	su	zmena používateľa		su -(uzivatel)		
T	touch	vytvorenie súboru		touch (sabor.pripona)		
	tr	translate or delete chars		zlúčenie/vymazanie znakov (napr pre delimiter)	tr -s "znak"	
		-s" "	nahradenie vela uvedených znakov len jedným takým (vvvvvv -> v)			
		-d	delete			
U	useradd	pridaj usera		sudo useradd (username)		
	userdel	-f	force (?)	sudo userdel -f (meno)		
	usermod	-g	pridanie užívateľov do skupiny	usermod -g (skupina) (uzivatel)		
V	vim	editor ako nano				
W	wc	word count				
	wget	non interactive net downloader		stiahnutie html súboru		
-O-		výpis na obrazovku				
				wget -O- stranka.sk grep "a href"		

§ 8 Počítačová architektúra

§ 8.1 North Bridge a South Bridge



(1) **North Bridge** (memory controller hub)

- a) zabezpečuje **vysokorýchlostné** prepojenie komponentov
- b) využíva **FSB** (Front-Side Bus) základná zbernica v počítači
- c) pripojené **CPU, AGP, PCI-e, RAM, Ethernet** (v serveroch)
- d) do AGP alebo PCI expres sa ďalej cez grafickú kartu zapája monitor

(2) **South Bridge** (I/O controller hub)

- a) **nižšie** dátové rýchlosti
- b) pripojené **BIOS, I/O, PCI, USB, IDE** (Integrated Drive Electronics), **SATA** (Serial Advanced Technology Attachment), **audio, EIDE, ISA**
- c) do IDE alebo SATA sa pripája harddisk, cez USB a I/O vstupno-výstupné zariadenia

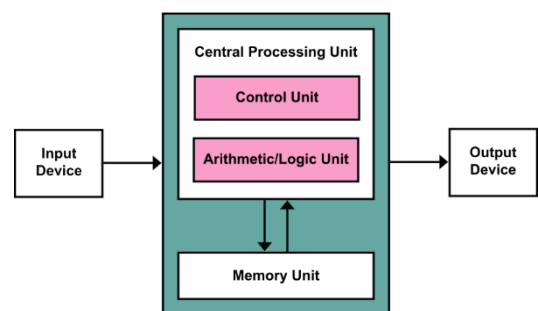
§ 8.2 Von Neumannova a Harvardská architektúra počítača

(1) najnákladnejší rozdiel je **rozdelenie pamäte**

- a) vo **Von Neumannovej** koncepcii je **RAM jednotná** pre program a pre dáta
- b) v **Harvardskej** je pamäť pre dáta **oddelená** od pamäte pre program

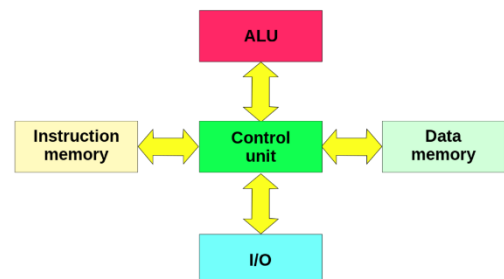
(2) vo **Von Neumannovom** počítači je aj program reprezentovaný pomocou dát

- a) je teda programovateľný, ale v jednom čase pristupuje buď k dátam alebo k programu (nie oboje)
- b) je teda **pomalší ako Harvard**



(3) **Harvardská** má oddelenú pamäť pre dáta od pamäti pre program

- a) teda práca celého počítača je **zrýchlená** – je možné čítať inštrukcie a dáta **odrazu**
- b) typická pre **mikroradiče**
- c) pamäť pre dáta a pre program môžu byť vyrobené odlišnou technológiou



§ 8.2.1 Radič (controller)

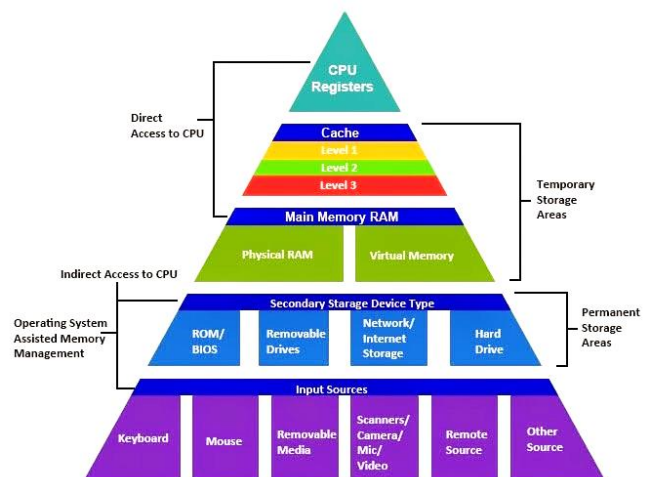
- (1) **centrálny prvok** architektúry (oboch)
- (2) **riadi** všetky časti počítača
- (3) riadi na základe **programu** uloženého v
 - a) **operačnej** pamäti (Von Neumann)
 - b) **inom** type pamäti (Harvard)
- (4) spracováva jednu inštrukciu za druhou, a podľa nich riadi ostatné časti počítača – dekoduje inštrukcie a zabezpečuje, **aby do ALU prichádzali správne operandy**
 - a) **ktoré dáta** sa majú prečítať z pamäti dát
 - b) **aký údaj** sa pošle na spracovanie do ALU
 - c) **akú operáciu** má ALU vykonať
 - d) určuje kam ALU **zapisuje výsledky** výpočtov

§ 8.2.2 Aritmeticko-logická jednotka (ALU)

- (1) slúži na vykonávanie základných výpočtov s operandami, ktoré sú načítané z operačnej pamäte (alebo pamäte v Harv.)
 - a) súčet, rozdiel, bitový súčet, bitový súčin, bitový posun, bitové rotácie
 - b) niektoré ALU robia aj zložitejšie operácie
- (2) vlastnosti ALU určujú aj vlastnosti počítača (64bit / 32bit)

§ 8.2.3 Pamäť počítača – Von Neumann

- (1) slúži na uchovávanie údajov, s ktorými prebiehajú výpočty
 - a) dočasné = **volatilná** pamäť (napr. RAM)
 - b) trvalé = **nevolatilná** pamäť (napr. harddisk) – po **odpojení napájania** sa údaje nestratia
- (2) uchováva aj **programy**, inštrukcie, operačné kódy načítavané do radiča
- (3) **hierarchicky** sa radí podľa veľkosti a rýchlosti
- (4) **druhy:**
 - a) **akumulátor a registre** – nízko kapacitné, ale veľmi rýchle
 - b) **vyrovnávacia pamäť** (cache) – na procesore
 - c) **operačná pamäť** (RAM)
 - d) **pevný disk, páskové pamäte**



Registre

- (1) **malé bloky pamäte** priamo na CPU
- (2) veľkosť = zopár bajtov
- (3) obsahujú dáta, ku ktorým CPU (alebo ALU) vie priamo pristupovať
- (4) napr. Intel Core i7 procesor má 8 (32bit) alebo 16 (64bit) registrov

RAM pamäť

- (1) **dočasné** uchovávanie údajov
- (2) **rýchlosť** je daná
 - a) **frekvenciou** zbernice
 - b) **šírkou** zbernice (násobia sa spolu)
- (3) obsahuje OS a **aplikácie** počas ich behu
- (4) **typy** – delia sa podľa toho akým spôsobom a koľko súčiastok je použitých na uloženie 1bitu
 - a) **statické** (SRAM)
 - nepotrebujú obnovovanie informácie v bunke
 -
 - b) **dynamické** (DRAM)
 - najpoužívanejšie
 - **cyklické obnovovanie** informácie v bunkách pamäte (nevýhoda, pri obnovovaní informáciu nemožno prepisovať...)
 - sú **jednoduché na výrobu** (1 tranzistor a 1 kondenzátor)
 - delia sa na **asynchrónne** (reaguje okamžite na vstup) a **synchrónne** (synch. so zbernicou PC, spoločný hodinový signál)

Virtuálna pamäť (časť RAM)

- (1) keď má RAM **málo miesta**, údaje sa ukladajú na HDD (cacheovanie) ale rýchlosť HDD je oveľa nižšia ako RAM -> vykonávanie programu je **pomalšie**
- (2) vyhradená časť HDD...
- (3) vo Windowse uložená ako samostatný súbor **pagefile.sys**, v Linuxe na **samostatnej partícii**
- (4) niektoré bloky vykonávaného programu nie sú prítomné v RAM, ale na pevnom disku

Vyrovňavacia pamäť (cache)

- (1) je **rýchlejšia** ako RAM, ale má oveľa menšie kapacity
- (2) relatívne **malá** pamäť (> 1% RAM) určená pre informácie, ktoré bude CPU pravdepodobne potrebovať
- (3) **úrovne**
 - a) **Level 1** – čo najrýchlejší prístup k často používaným dátam, skladá sa z **dátovej** a **inštrukčnej** vyrovnávacej pamäte
 - b) **Level 2** – vyššia kapacita ako L1 ale **nižšia rýchlosť**, tiež môže byť rovnako rozdelená
 - c) **Level 3**
- (4) **typy** – podľa (ne)kopírovania dát v L1 do L2..
 - a) **exkluzívny**
 - b) **inkluzívny**
- (5) spôsoby zdieľania cache sú rôzne