

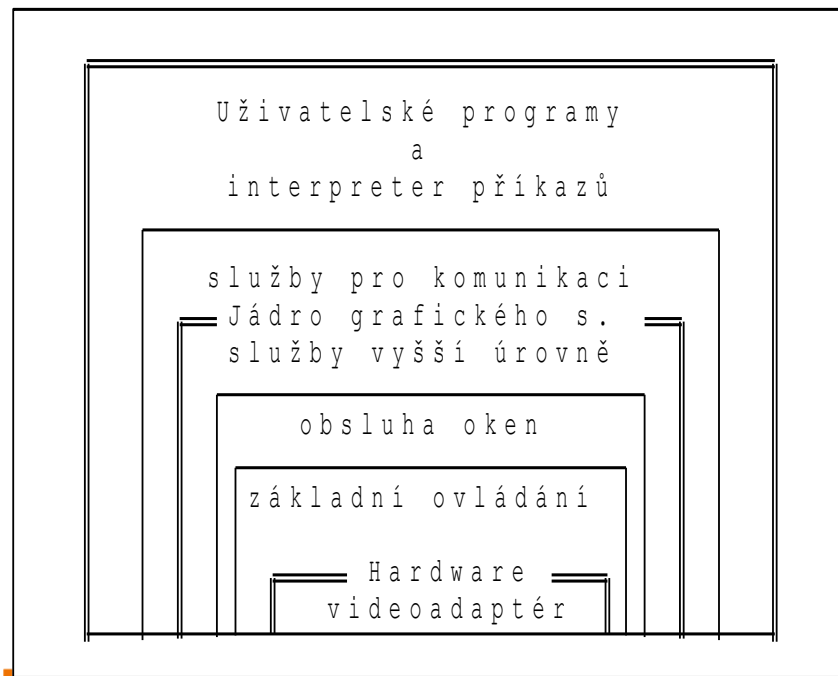


**OSTRAVSKÁ UNIVERZITA**  
PŘÍRODOVĚDECKÁ FAKULTA

# Grafický systém, okna, služby

Ing. Pavel Smolka, Ph.D.

# Vrstvený grafický systém



# Vrstvený grafický systém

- Na nejnižší úrovni - jednoduchý systém základních grafických služeb umožňujících vlastní zápis základních grafických objektů na obrazovku. Je-li počítač osazen kvalitním grafickým procesorem, nemusí být tento systém vůbec zapotřebí.
- V další vrstvě - systém zajišťující práci s obrazovkovými okny a/nebo s virtuálními obrazovkami. V závislosti na konkrétním návrhu grafického systému mohou být tyto dvě nejnižší vrstvy navzájem 'prohozeny'. Na vrstvě obsluhující okna každopádně leží zodpovědnost za korektní spolupráci s interaktivními vstupními zařízeními (jako je klávesnice nebo myš) i se samotnými procesy - ty totiž někdy potřebují vědět, v jakém stavu jsou právě jejich okna.



- Na další úrovni - vrstva umožňující programátorům aplikací pracovat s grafickými objekty (jako je čára, čtverec, kruh, plocha nebo třeba koule), a ne s nějakými obrazovými body, jejichž počet i barva závisí na grafickém adaptéru i na jeho momentálním režimu práce.
- Nejvyšší úrovní - vrstva služeb uživatelského grafického rozhraní. Jedná se o prostředky, které programům usnadní komunikaci s uživatelem prostřednictvím nabídek (menu), dialogových oken a řady dalších, dnes již do značné míry standardizovaných, prvků. Tato vrstva je velmi důležitá nejen pro usnadnění práce aplikačním programátorům, ale především proto, aby bylo ovládání všech aplikací podobné a konzistentní.
- A poslední - je jí grafický interpret příkazů uživatele, který v grafických systémech stojí na místě 'shellu' systémů orientovaných textově.



# Počty barev

Počet barev	Počet bitů	
16	4	
256	8	
65536	16	Hi color
16,7 mil	24	True color
4 294,9 mil	32	

Pozn. Ve skutečnosti většina grafických adaptérů zobrazuje  
max. 16,7 mil barev



# Velikosti paměti

Rozlišení	256 barev	64 tis.barev	16 mil.barev	40294,9 mil barev
640x480	512KB	1MB	1MB	4MB
800x600	512KB	1MB	2MB	8MB
1024x768	1MB	2MB	4MB	8MB
1280x1024	2MB	4MB	4MB	16MB
1600x1200	2MB	4MB	6MB	32MB

# Uživatelské rozhraní

- Okna
- Ikony
- Nabídky
- Dialogová okna
- Výstražné dialogy
- Kurzory
- Indikátory



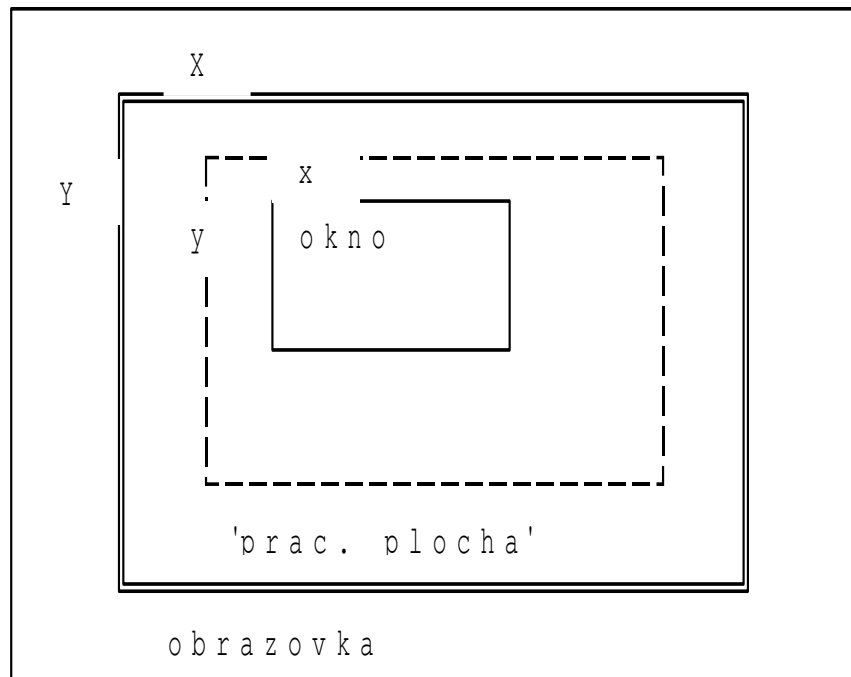
# Okna

- Okna jsou mechanismem pro virtualizaci obrazovky
- Okno je samostatné grafické výstupní zařízení.
- Z hlediska programu je okno obvykle pravoúhelníkem určitých rozměrů (s některými dalšími atributy), do kterého lze zapisovat libovolné grafické informace. Uživatel pak víceméně nezávisle na programu, který s oknem pracuje - určí postavení okna na obrazovce, jeho rozměry a viditelnost; operační systém sám se postará o správné zobrazení obsahu okna (určeného programem) na obrazovce.

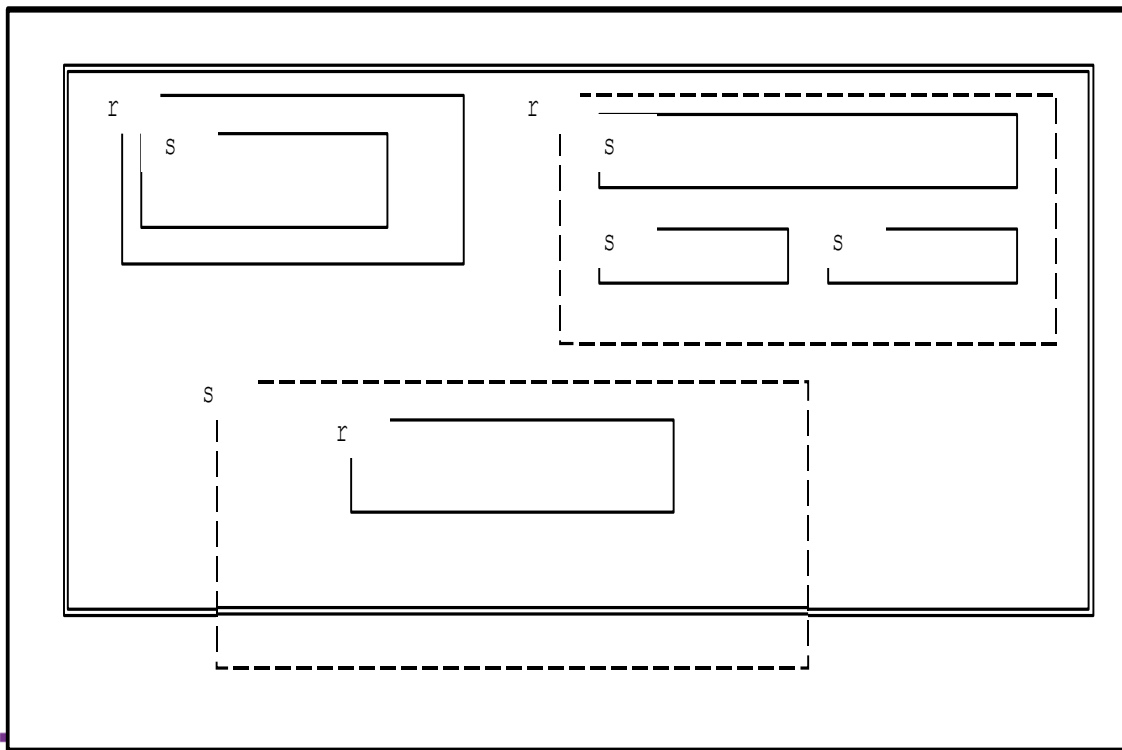




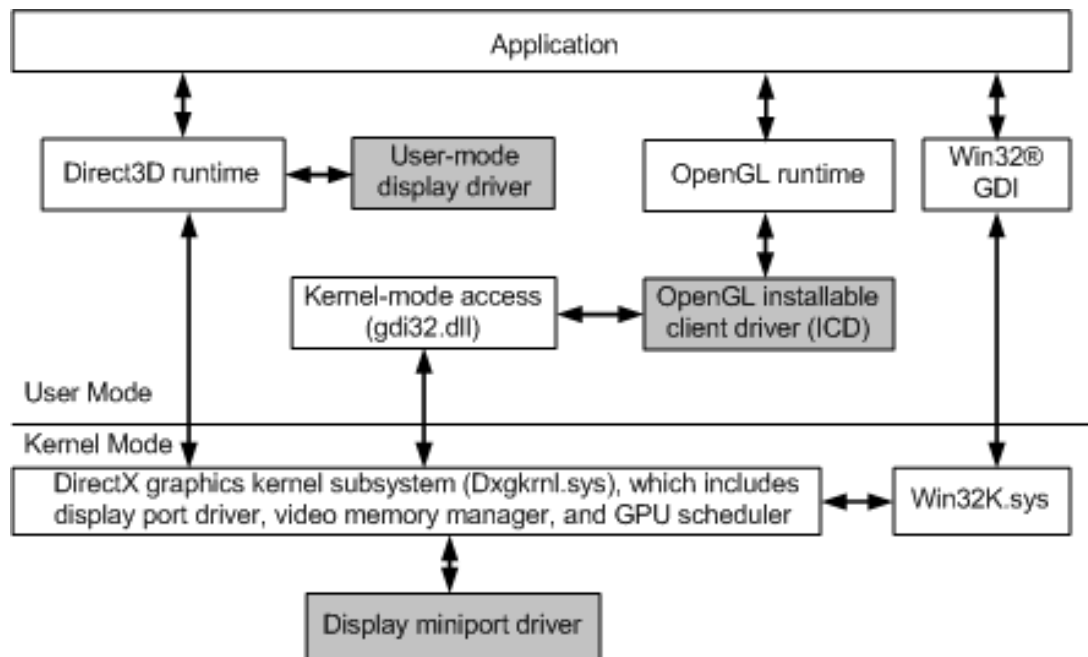
# Okno na obrazovce



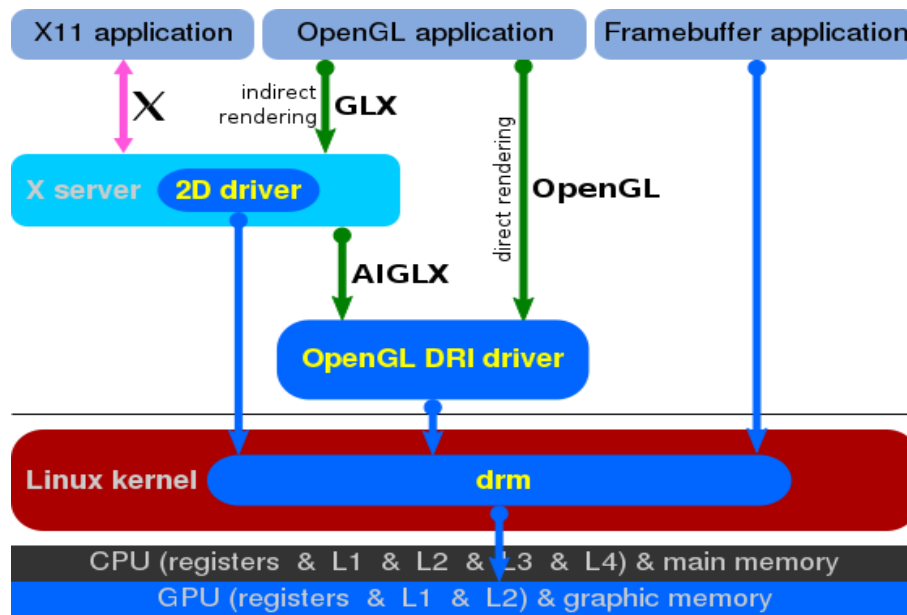
# Hierarchie oken



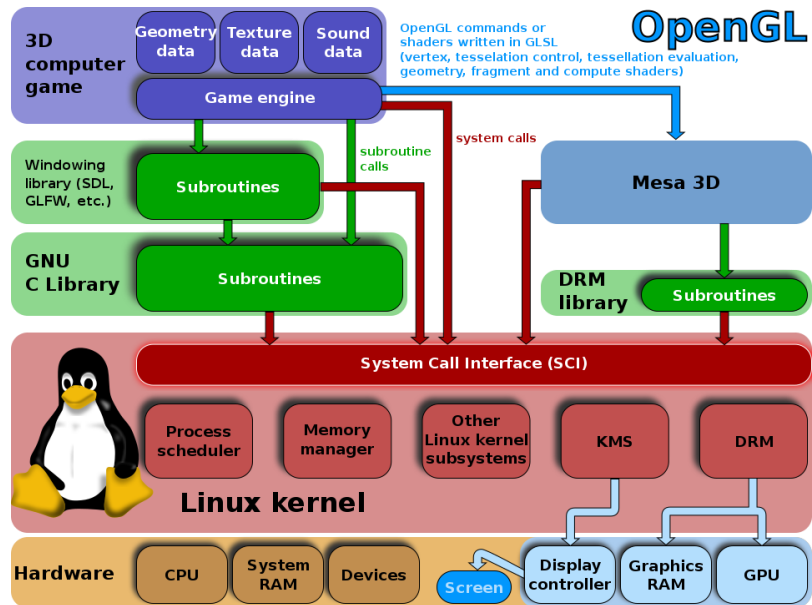
## Windows Display Driver Model (WDDM) Architecture

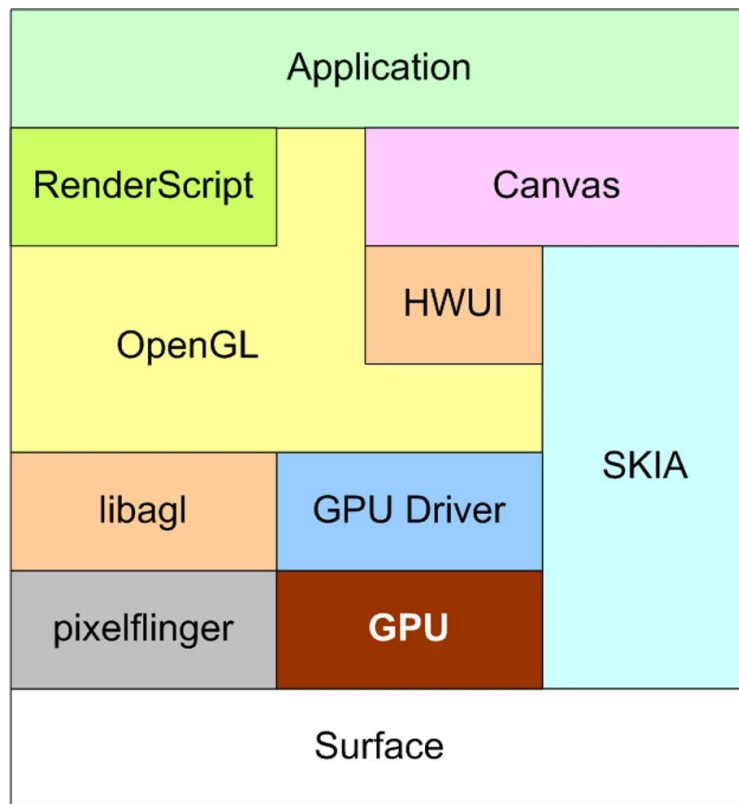


# Linux - Direct Rendering Infrastructure (DRI)



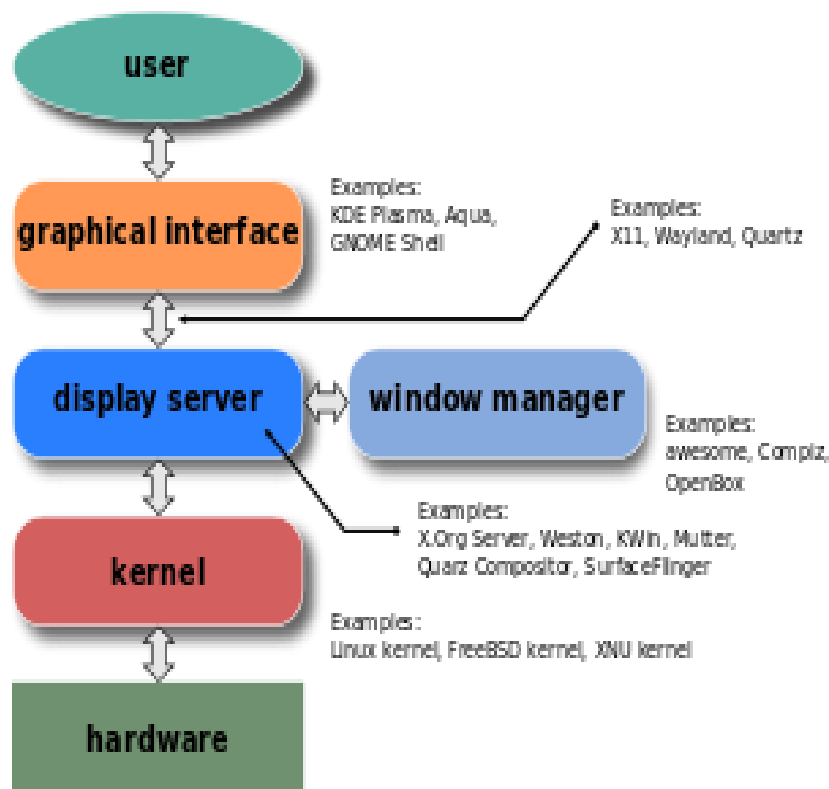
# Mesa - Linux





©2012 MIPS Technologies

Grafický subsystém androidu



# GUI - historie

- PARC (WIMP)
  - Xerox 8010 Star Information systém (1981)
  - Apple Lisa (1983)
  - OS/2 (1988)
  - Windows 1.0 (1985) – nástavba MSDOS
  - WinNT 3.1 (1993) – nosné prostředí GUI
  - X-Windows (1984)
- Post – WIMP – mobilní zařízení





# Paradigma WIMP

- Windows, Ikony, Menu, Ukazovací zařízení
- pro všechna GUI koncem 80. let - podobné principy ovládání a zobrazování
- pojem virtuálního „pracovního stolu“ (desktop)
- okno (window) sdružuje související objekty (patřící jedné aplikaci)
- zobrazují se různé ovládací prvky (controls, widgets)
  - tlačítka (button)
  - menu (pop up menu, drop-down menu)
  - vypínače (checkbox)
  - vícepolohové přepínače (radio button)
  - seznamy (list, drop-down list)
  - textová pole (edit control)
  - kombinované prvky (drop down list, combo box)
- s ovládacími prvky se manipuluje obvykle myší



# Post-WIMP GUI

- pokročilejší, resp. odlišné ovládání než WIMP
- hodí se na specifické účely
- počítačové hry
- virtuální realita
- ovládání pomocí řeči
- fyzicky založené ovládání (TUI - Tangible User Interface)
  - tangible = hmatatelný
  - letové simulátory atd..
- ovládání interagující s více smysly (multisensory user interface)



# Program řízený událostmi 1

- event driven programming
- program odpovídá na události (events), resp. Zprávy (messages) vykonáním příslušné činnosti
- v GUI události generovány (v abstraktním pojetí) jednotlivými ovládacími prvky
  - stisk tlačítka
  - výběr položky z menu
  - změna velikosti okna
  - požadavek na ukončení programu
- reakce na událost je obvykle implementována pomocí funkce typu callback (event handler)



# Program řízený událostmi 2

- vytvoření funkčního GUI spočívá v:
  - návrhu rozložení ovládacích prvků
  - implementaci reakcí na události v podobě funkcí – event handlerů
  - přiřazení (binding, registration) handlerů k událostem
- „hlavní program“ (funkce `main()`, resp. `WinMain()`) obsahuje:
  - inicializaci (vytvoření datových struktur objektů, registraci event handlerů)
  - vstup do hlavní smyčky



# Program řízený událostmi 3

- **hlavní smyčka** (main loop, event loop, event dispatcher)
  - přijímá vstup uživatele, resp. další zprávy, od operačního systému
  - implementuje generování událostí jednotlivými ovládacími prvky
  - zjistí, kam uživatel kliknul, zda se na dané pozici nachází nějaké tlačítko atd...
  - zavolání handler registrovaný pro danou událost
- **Win32 API:** funcke WinMain()
  - volá ve smyčce GetMessage()
  - GetMessage() čeká dokud nenastane událost, vrátí ji v podobě zprávy
  - DispatchMessage() reaguje na zprávu (volá callback)
- **X Window:** hlavní smyčka obsažena v knihovně Xlib



# Objektově orientovaný přístup 1

- **API** (Application Programming Interface, rozhraní pro programování aplikací) - využívají vlastností objektově orientovaných jazyků
- **hierarchie objektů** s využitím **dědičnosti** (inheritance) a **polymorfismu** (virtuální metody)
  - ovládací prvky mají společnou rodičovskou třídu (base class)
  - virtuální metody: nakresli mě, aktivuj mě, vrať seznam mých událostí atd.
- Každé okno **implementováno** jako třída
  - atributy jsou ovládací prvky
  - konstruktor okna registruje event handlers, resp. i samotné ovládací prvky
  - virtuální metody: nakresli okno, změň velikost, zavři okno atd.



# Objektově orientovaný přístup 2

- objektově orientovaná rozhraní k API operačního systému
- na různých úrovních:
  - pouze knihovna (např. GTK+)
  - programovací jazyk a vývojové prostředí (Microsoft .NET apod.)
  - rozšíření jazyka, resp. možností kompilátoru o specifické funkce



# Desktopové prostředí Windows 1

- GDI (Graphics Device Interface) vykresluje fonty, čáry, křivky, spravuje palety - komunikuje s ovladačem grafické karty
- analogie GDK v X Window/GTK+
- akcelerace 2D grafiky - DirectDraw
- GDI používá manažer oken (user32.dll)
- GDI reprezentuje objekty pomocí prostředků (resources)
- již od Windows XP modernizace - GDI+





# Desktopové prostředí Windows 2

- Od Windows Vista, Windows 7, 8, 10 – náhrada:
  - **DWM** (Desktop Window Manager)
- kompozitní manažer oken (viz dále)
- „staré“ aplikace používající GDI běží pod DWM (GDI již nekomunikuje přímo s ovladačem grafiky a není hardwarově akcelerováno)
- akcelerace 2D grafiky - Direct2D (DirectDraw součástí DirectX, ale již je považován za zastaralý)
- AERO - Autentický, Energický, Reflexní a Otevřený
  - Windows Flip 3D



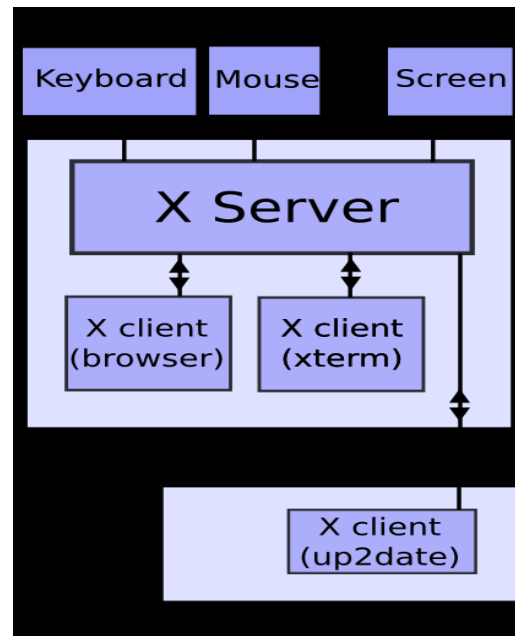
# X Windows systém

- systém zobrazování grafiky a grafického uživatelského rozhraní pro počítače různých architektur a operačních systémů
- začátky v r. 1984 na MIT v rámci projektu ATHENA
  - (projekt distribuovaného počítačového systému, v rámci něho vznikl např. Kerberos, ovlivnil vznik LDAP atd.)
- budován jako síťový protokol (X display protocol) → aplikace může běžet na jednom počítači a zobrazovat se na jiném (klient a server)
  - od r. 1987 verze protokolu X11 - až do současnosti
- datový tok lze tunelovat přes ssh (za účelem šifrování)



# X Windows server, X klient

- X server běží na počítači uživatele
- X klient je aplikace (na stejném nebo jiném počítači), která chce na počítači uživatele (terminálu) zobrazovat
- X server poskytuje
  - vykreslování oken - výřezů obrazovky (okno aplikace, ale i ovládací prvky - tlačítko, seznam, apod.)
  - interakci s periferiemi (klávesnice, myš)
  - vykreslování fontů
- X server přijímá od klienta požadavky (requests) (nakresli to a to) a odesílá mu události (events) (stisk klávesy, změna velikosti okna apod.)
- Okna uspořádána v hierarchii



# Vzdálení plocha

- model klient-server
- RDP (Remote Desktop Protocol)
  - vzdálená plocha Windows
  - umí na vzdálený terminál přesměrovat i zvuk a tisk
- VNC (Virtual Network Computing)
  - používá protokol RFB (Remote FrameBuffer)
  - lze přeposílat dál přes RDP (xRDP server pro UNIX)
- NX - Nomachine NX, X2GO (nekompatibilní)
  - komprimovaný a SSH tunelovaný X protokol
  - velmi svižný i při pomalém připojení
  - bitmapy se přenášeny komprimované jako JPEG
  - umí přesměrovat i tisk
- XDMCP (X Display Manager Control Protocol)
  - (nekomprimovaný) přenos celého X desktopu včetně přihlašovací obrazovky



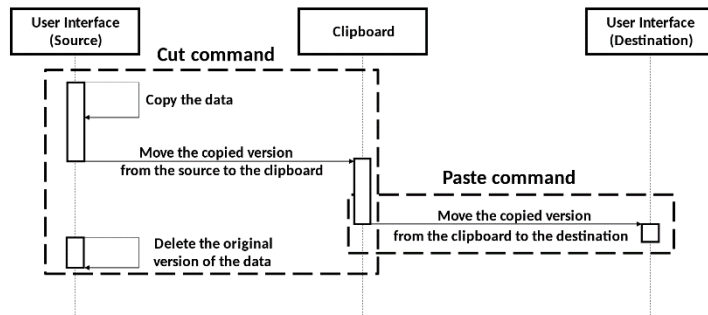
# Služeb OS

- Textové služby
- Národní prostředí
- Databáze
- Komunikace programů
  - Schránka
  - Inteligentní schránka
  - Spojení dat
  - 'Drag and drop'
  - Datové služby



# Schránka - clipboard

- Komponenta OS umožňuje v rámci jednoho programu (schránka aplikace) nebo mezi programy (schránka shellu) pohodlným způsobem přenášet data.
- Schránka shellu je pro čtení i zápis dostupná všem pod danou instancí shellu běžícím programům.



# Schránka Windows

- Schránka v systému Microsoft Windows obsahuje jednu položku ve více dostupných formátech.
- Tři různé typy možných podporovaných formátů jsou:
  - standardní formáty (např. CF\_BITMAP nebo CF\_UNICODETEXT),
  - registrované formáty (např. CF\_HTML),
  - soukromé formáty pro interní použití
- V novějších verzích systému Windows je obsah schránky přístupný prostřednictvím správců schránky.
- Data lze ukládat do schránky Windows jak prostřednictvím GUI, či klávesových příkazů, tak pomocí příkazového řádku.

# CLI (Command Line Interface)

- Interpret příkazů - Shell
  - Příkazy a jejich syntaxe
- Microsoft Windows
  - Command.com (cmd.exe)
  - PowerShell
- Unixové shelly
  - Bourne-Again shell
  - Bourne shell
  - C shell atd.





# PowerShell

- Rozšiřitelný textový (řádkový) shell se skriptovacím jazykem od společnosti Microsoft.
- Produkt je založen na platformě .NET Framework.
- Cmdlet (sloveso-podstatné jméno)
- PowerShell ISE - hostitelská aplikace pro Windows PowerShell.



# Výpočet výkonu disků

- Základem je IOPS - (input / output operations per second). Klíčové pro výpočet jsou následující parametry:
  - Average latency
  - Average seek time - write
  - Average seek time – read

Příklad výpočtu pro čtení:

$$\text{IOPS} = 1 / (\text{Average latency} + \text{Average seek time})$$



# Výpočet výkonu diskového pole

- Uplatňujeme penalizaci dle typu:
  - RAID 1 = 2
  - RAID 10 = 2
  - RAID 5 = 4

IOPS / penalizace



# Výpočty kapacity RAID polí a bezpečnost

- RAID 0
  - $n \times s$  ( $n$  je počet disků,  $s$  je velikost jednoho disku)
  - obnova dat není možná
- RAID 1
  - $n / 2$  ( $n$  je počet disků)
  - vyšší rychlost čtení a zápisu
- RAID 3 a 4
  - velikost je rovna  $n-1$  ( $n$  počet disků, jeden je vyhrazen na paritní informace)
  - rozdíl je ve velikosti datového bloku, který u RAID 4 nemusí být 1 byte



# Výpočty kapacity RAID polí a bezpečnost

- RAID 5
  - minimálně 3 disky
  - $s_{\min} \times (n - 1)$  ( $n$  je počet disků,  $s_{\min}$  je velikost nejmenšího disku)
  - obnova dat není možná
  - dosahuje lepší rychlosti čtení u větších souborů.
  - bezpečnost → máme pole RAID5 o 5 discích, 4 jsou na data, 1 pro paritní informace (kapacitně)

Disk #1: 00101010 (Data)  
Disk #2: 10001110 (Data)  
Disk #3: 11110111 (Data)  
Disk #4: 10110101 (Data)  
Disk #5: ----- (Parita)

Jak už bylo řečeno, parita se vypočítá logickou funkcí XOR. Výsledek proto bude

00101010 XOR 10001110 XOR 11110111 XOR 10110101 = 11100110



# RAID 5

- V případě poškození jednoho disku:

Disk #1: ???????? (Data)  
Disk #2: 10001110 (Data)  
Disk #3: 11110111 (Data)  
Disk #4: 10110101 (Data)  
Disk #5: 11100110 (Parita)

Disk #1 = Disk #2 XOR Disk #3 XOR Disk #4 XOR Disk #5 (Parita)  
Disk #1 = 00101010



# RAID 6

- Minimem jsou čtyři disky.
- RAID 6 je podobný RAIDu 5 s tím, že paritní blok není jeden, ale dva, takže pole, které má kapacitu  $n-2$  disků, unese selhání kterýchkoliv dvou z nich.
- Výkon RAIDu 6 je podobný jako výkon RAIDu 5, náročnost na výpočetní výkon je ovšem o něco vyšší (počítají se dva paritní bloky).



# Děkuji za pozornost

