

# Grafická uživatelská rozhraní

Pavel Strachota

FJFI ČVUT v Praze

6. prosince 2016

# Obsah

- 1 Historie GUI
  - Xerox Alto
  - PERQ
  - Xerox Star
  - Apple Lisa
  - VisiOn
  - GEM
  - OS/2
  - NeXTstep
  - InView
  - Windows
  - Mac OS
  - X-Window
- 2 Koncepte GUI
- 3 Nástroje pro tvorbu aplikací s GUI
- 4 Desktopové prostředí
- 5 Vzdálená plocha

# Obsah

- 1 Historie GUI
  - Xerox Alto
  - PERQ
  - Xerox Star
  - Apple Lisa
  - VisiOn
  - GEM
  - OS/2
  - NeXTstep
  - InView
  - Windows
  - Mac OS
  - X-Window
- 2 Koncepce GUI
- 3 Nástroje pro tvorbu aplikací s GUI
- 4 Desktopové prostředí
- 5 Vzdálená plocha

# Grafické uživatelské rozhraní

- **GUI** - *Graphical User Interface*
- umožňuje snadno a pohodlně ovládat počítač a dobře se orientovat v informacích
- původně pouze **CLI** (*Command Line Interface*) - ovládání pomocí příkazové řádky
- začátky GUI již v 70. letech
- nejprve náhled do historie ...



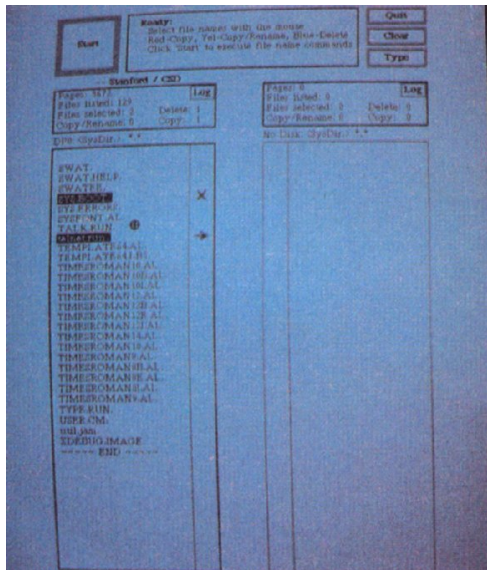


## 1973: Xerox Alto



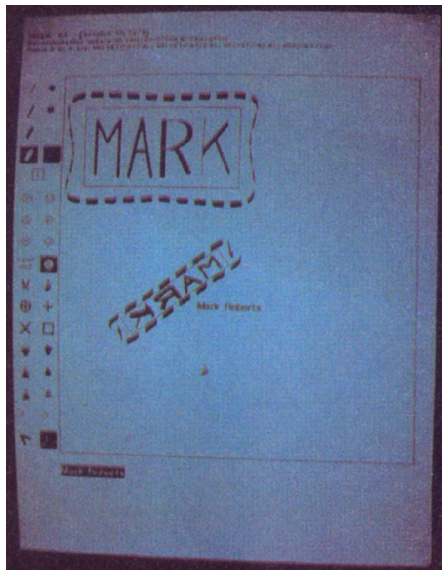
- dvoubarevný (černo-bílý) monitor orientován na výšku
- rozlišení  $606 \times 808$  pixelů
- 3-tlačítková myš
- Systém „Alto Executive“
- SmallTalk - objektově orientovaný jazyk + vlastní grafické prostředí
- nikdy nedošlo ke komerčnímu prodeji
- na univerzitách, ve výzk. centru Palo Alto

# 1973: Xerox Alto



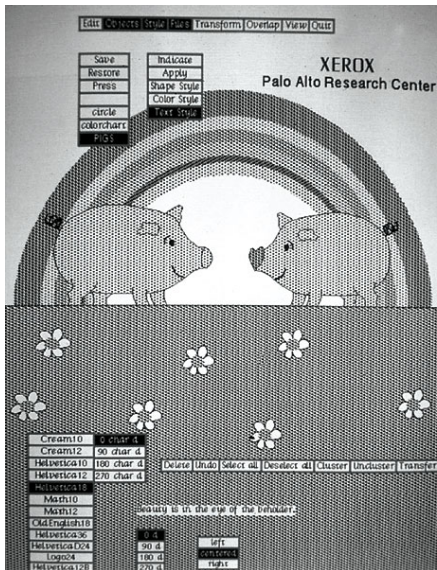
- dvoubarevný (černo-bílý) monitor orientován na výšku
- rozlišení 606 × 808 pixelů
- 3-tlačítková myš
- Systém „Alto Executive“
- SmallTalk - objektově orientovaný jazyk + vlastní grafické prostředí
- nikdy nedošlo ke komerčnímu prodeji
- na univerzitách, ve výzk. centru Palo Alto

## 1973: Xerox Alto



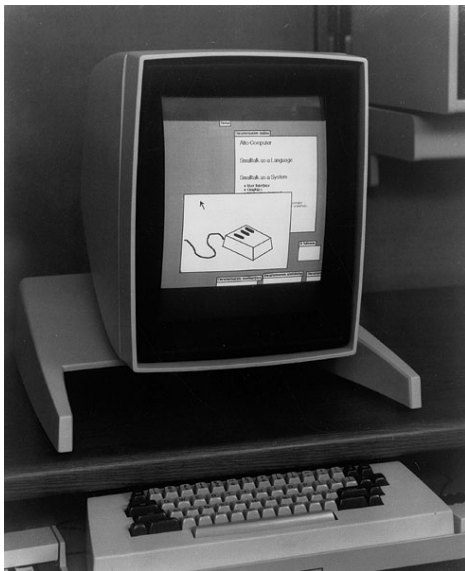
- dvoubarevný (černo-bílý) monitor orientován na výšku
- rozlišení 606 × 808 pixelů
- 3-tlačítková myš
- Systém „Alto Executive“
- SmallTalk - objektově orientovaný jazyk + vlastní grafické prostředí
- nikdy nedošlo ke komerčnímu prodeji
- na univerzitách, ve výzk. centru Palo Alto

# 1973: Xerox Alto



- dvoubarevný (černo-bílý) monitor orientován na výšku
- rozlišení 606 × 808 pixelů
- 3-tlačítková myš
- Systém „Alto Executive“
- SmallTalk - objektově orientovaný jazyk + vlastní grafické prostředí
- nikdy nedošlo ke komerčnímu prodeji
- na univerzitách, ve výzk. centru Palo Alto

## 1973: Xerox Alto



- dvoubarevný (černo-bílý) monitor orientován na výšku
- rozlišení 606 × 808 pixelů
- 3-tlačítková myš
- Systém „Alto Executive“
- SmallTalk - objektově orientovaný jazyk + vlastní grafické prostředí
- nikdy nedošlo ke komerčnímu prodeji
- na univerzitách, ve výzk. centru Palo Alto

# 1980: PERQ

Distributed Computing Carried To Its Ultimate

## PERQ—The Processor Per Person Machine

Why settle for a piece of a computer when Three Rivers can add a PERQ™ to your benefits package. Imagine having a display that shows you data the way you really want it



— crisp, proportionally spaced, black on white text in a variety of fonts, and high resolution graphics. We think our picture is the best in the business, and wait till you see how fast we can update it.

For interacting with the screen, a cursor positioning tablet is standard equipment. You point—



PERQ performs. Check out the keyboard. It's compact, detachable, reliable. It's your key to a powerful software package that includes a user friendly, easy to use operating system, a super pascal compiler, a what-you-see-is-what-you-get text editor, and other software attractions too numerous to mention.



And—your very own processor. It's fast, has a 32 bit virtual address system to handle your big programs, and it's

microprogrammable with our optional writable central store. PERQ comes complete with 256K—1M memory, 12-24M of hard disk storage, RS-232 & IEEE-488 interfaces. Plus—PERQ is on the bus, with our compatible 10 M, 6.17, / SEC Ethernet™ local network option.

So—why share? You deserve another PERQ.

\*PERQ is a trademark of Three Rivers Computer Corporation.  
™Ethernet is a trademark of Xerox Corporation.



**Three Rivers Computer**  
Three Rivers Computer Corporation  
720 Grass Street  
Pittsburgh, Pennsylvania 15224  
412/621-6250

CIRCLE 19 ON READER CARD

- výrobce *Three Rivers Computer*
- založen na systému Alto
- poprvé komerční prodej, ale malý úspěch kvůli vysoké ceně

# 1980: PERQ

## PERQ - A Landmark Computer System

### A Time Sharing Alternative

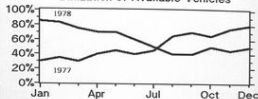
PERQ provides all the benefits of a time shared mainframe without the drawbacks. For about the same cost per user, each user gets a very powerful CPU, a minimum of a quarter megabyte of memory with a big virtual address space, a high resolution video display with full graphics capabilities, a keyboard, and a large capacity rigid disk—integrated into a complete system right at his desk. PERQ has consistent, rapid response time, unaffected by other users' load. System reliability is increased because a failure in one workstation does not affect other users.

A high speed network accesses shared resources such as printers, tape drives, and distributed file systems as if they were local. And because PERQ workstations provide computing resources on a per person basis, installations can be expanded incrementally.

### An Intelligent Terminal

PERQ as an intelligent terminal can stretch your existing system's capabilities. Many of the cycles on your mainframe are used in editors, command processors, and debuggers, with very little time spent on the large number crunching problems your computer was designed for. By off-loading many common functions into PERQ, which has been specifically designed to excel at these interactive tasks, you can extend the viability of your current installation.

Utilization of Available Vehicles



PERQ S

Pascal Compiler

Multi-Process Operating System

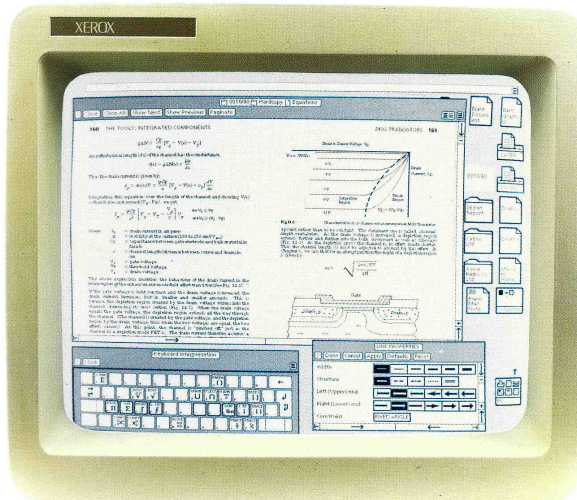
Display Window Manager

Formatting Text Editor

```
PERQ console: 01/11/80 11:22:22
loading stack segment 20
loading segment 30 (MUPHIN.SEG)
loading segment 25 (SLEEP.SEG) -- main program
loading segment 34 (SIGUTIL.SEG)
loading stack segment 20
```

- výrobce *Three Rivers Computer*
- založen na systému Alto
- poprvé komerční prodej, ale malý úspěch kvůli vysoké ceně

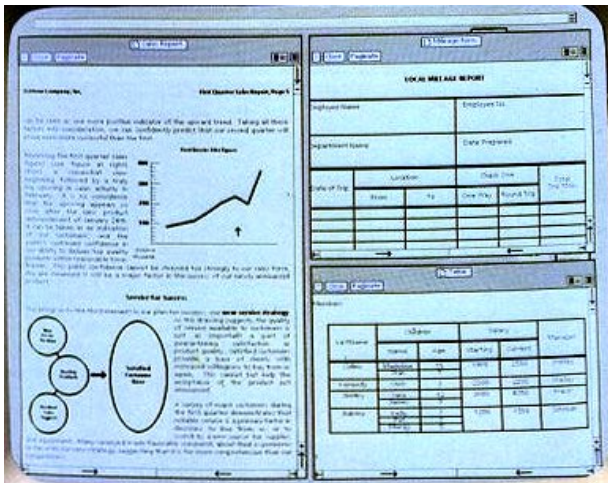
# 1981: Xerox Star



- koncept založen na Xerox Alto
- operační systém zcela přepsán
- ČB displej o úhlopříčce 17"
- rozlišení 1024 × 808 pixelů
- podpora mnoha jazyků (kódování podobné Unicode)
- SmallTalk



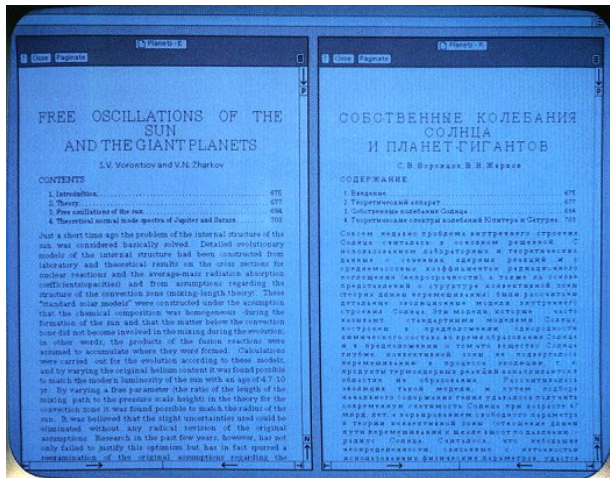
# 1981: Xerox Star



- koncept založen na Xerox Alto
- operační systém zcela přepsán
- ČB displej o úhlopříčce 17"
- rozlišení 1024 × 808 pixelů
- podpora mnoha jazyků (kódování podobné Unicode)
- SmallTalk

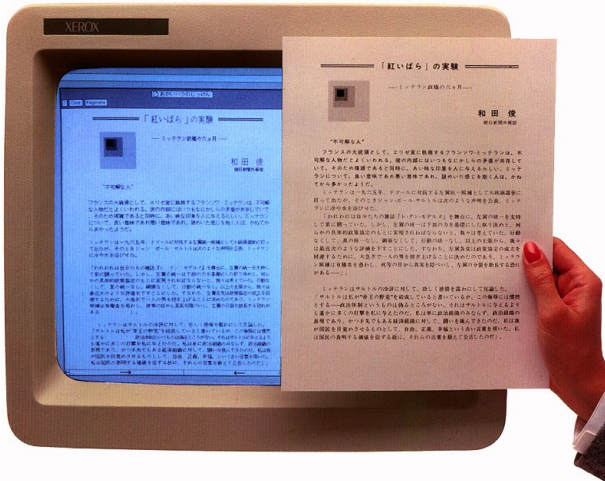


# 1981: Xerox Star



- koncept založen na Xerox Alto
- operační systém zcela přepsán
- ČB displej o úhlopříčce 17"
- rozlišení 1024 × 808 pixelů
- podpora mnoha jazyků (kódování podobné Unicode)
- SmallTalk

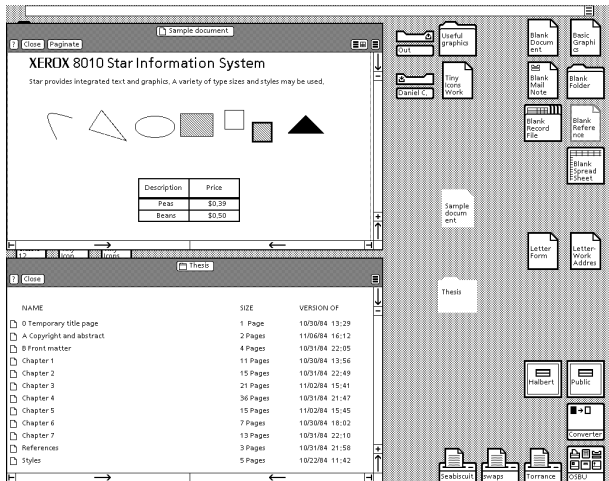
# 1981: Xerox Star



- koncept založen na Xerox Alto
- operační systém zcela přepsán
- ČB displej o úhlopříčce 17"
- rozlišení 1024 × 808 pixelů
- podpora mnoha jazyků (kódování podobné Unicode)
- SmallTalk



# 1981: Xerox Star



- koncept založen na Xerox Alto
- operační systém zcela přepsán
- ČB displej o úhlopříčce 17"
- rozlišení 1024 × 808 pixelů
- podpora mnoha jazyků (kódování podobné Unicode)
- SmallTalk

## 1981: Xerox Star



- koncept založen na Xerox Alto
- operační systém zcela přepsán
- ČB displej o úhlopříčce 17"
- rozlišení 1024 × 808 pixelů
- podpora mnoha jazyků (kódování podobné Unicode)
- SmallTalk

## 1983: Apple Lisa



- první Apple s GUI (Apple II byl již z r. 1978)
- současně probíhal vývoj Apple Macintosh
- původně spolupráce s Xerox PARC (výrobce Alta a Star)

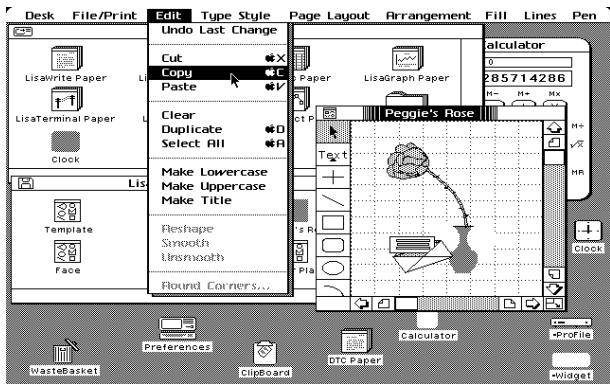
# 1983: Apple Lisa



- první Apple s GUI (Apple II byl již z r. 1978)
- současně probíhal vývoj Apple Macintosh
- původně spolupráce s Xerox PARC (výrobce Alta a Star)



# 1983: Apple Lisa



- první Apple s GUI (Apple II byl již z r. 1978)
- současně probíhal vývoj Apple Macintosh
- původně spolupráce s Xerox PARC (výrobce Alta a Star)

## 1983: VisiOn



Visi ON™

APPLICATIONS  
MANAGER



COPYRIGHT 1983 VISICORP  
ALL RIGHTS RESERVED  
VERSION 1.0

Calibrate the mouse. See the Setup Guide for detailed instructions.

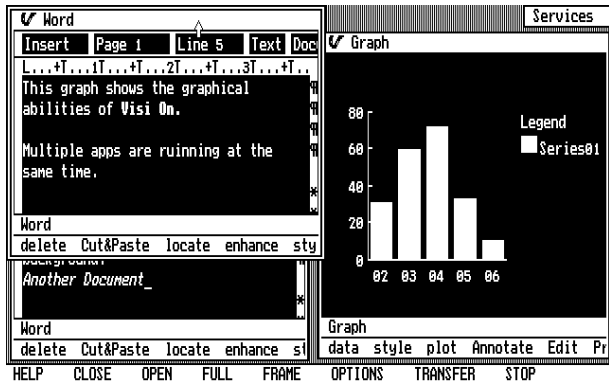
- GUI od VisiCorp pro PC s MS-DOS
- VisiOn Application Manager \$495
- VisiOn Graph \$250
- VisiOn Calc \$395
- VisiOn Word \$375
- myš \$295
- celkem **\$1810**
- scrollování pohybem myši + pravé tlačítko dole

# 1983: VisiOn



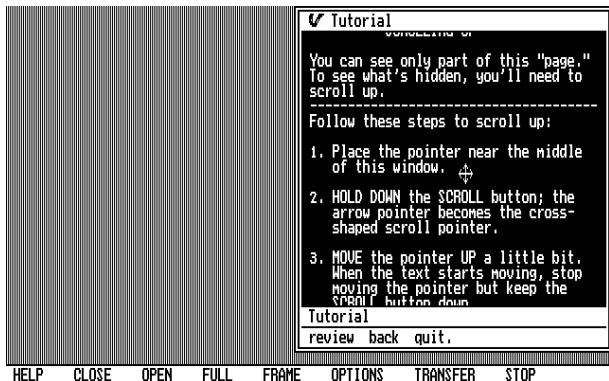
- GUI od VisiCorp pro PC s MS-DOS
- VisiOn Application Manager \$495
- VisiOn Graph \$250
- VisiOn Calc \$395
- VisiOn Word \$375
- myš \$295
- celkem **\$1810**
- scrollování pohybem myši + pravé tlačítko dole

# 1983: VisiOn



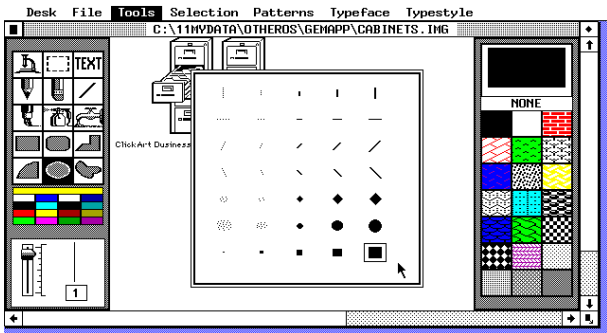
- GUI od VisiCorp pro PC s MS-DOS
- VisiOn Application Manager \$495
- VisiOn Graph \$250
- VisiOn Calc \$395
- VisiOn Word \$375
- myš \$295
- celkem **\$1810**
- scrollování pohybem myši + pravé tlačítko dole

# 1983: VisiOn



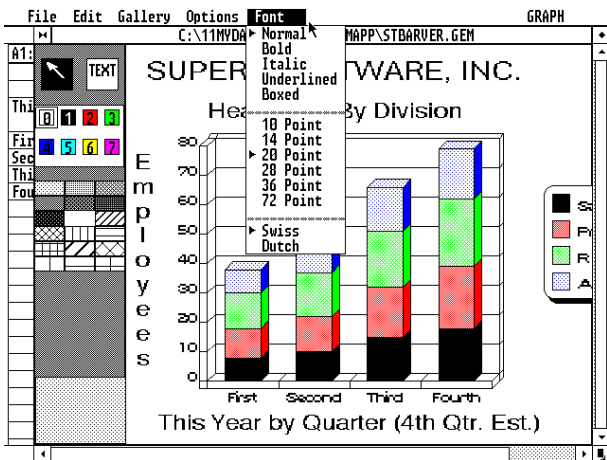
- GUI od VisiCorp pro PC s MS-DOS
- VisiOn Application Manager \$495
- VisiOn Graph \$250
- VisiOn Calc \$395
- VisiOn Word \$375
- myš \$295
- celkem **\$1810**
- scrollování pohybem myši + pravé tlačítko dole

# 1985: GEM



- vyvinuto Digital Research
- pro systém MS-DOS
- později funkčnost záměrně omezena kvůli sporům s Apple
- k dispozici i na Atari TOS

## 1985: GEM



- vyvinuto Digital Research
- pro systém MS-DOS
- později funkčnost záměrně omezena kvůli sporům s Apple
- k dispozici i na Atari TOS

# 1985: GEM

The screenshot displays the GEM desktop environment. At the top, a menu bar includes 'File', 'Edit', 'Block', 'Layout', 'Style', 'Spelling', 'Graphics', and 'Help'. A window titled '1st Word Plus...' is open, showing a text document with the following content:

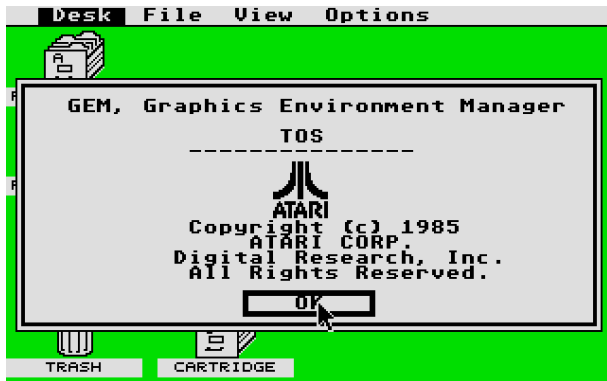
We are pleased to inform you that GEM 1st Mail is now a  
 for use with GEM 1st Word Plus. We believe many of the  
 of company will like the advanced features offered by GEM 1st  
 Mail, including the automatic reformatting of paragraphs with  
 insertions in them. For further details, contact your dealer.

Below the text editor, a 'GEM character set' table is visible, listing various characters and their corresponding GEM codes. To the right of this table is a vertical menu with options: SPACE, RETURN, DELETE <-, DELETE >-, TAB, and FIXED SPACE. At the bottom of the window, a row of function keys (F1-F10) is shown with their respective actions: F1 BOLD, F2 UNDER, F3 ITALIC, F4 LIGHT, F5 SUPER, F6 SUB, F7 INSERT, F8 DEL LINE, F9 INDENT, F10 REFORMAT, and CAPS LOCK.

- vyvinuto Digital Research
- pro systém MS-DOS
- později funkčnost záměrně omezena kvůli sporům s Apple
- k dispozici i na Atari TOS

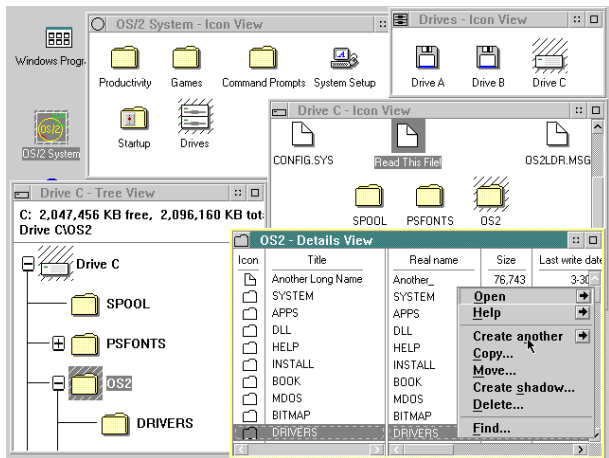


# 1985: GEM



- vyvinuto Digital Research
- pro systém MS-DOS
- později funkčnost záměrně omezena kvůli sporům s Apple
- k dispozici i na Atari TOS

# 1988: OS/2



- zpočátku společný projekt IBM a Microsoftu
- začátek 1987, GUI od 1988
- OS/2 2.0
- OS/2 3.0 Warp (1994)
- OS/2 4.0 Warp (1996)
- Warp uměl spouštět aplikace pro Windows 3.11

# 1988: OS/2

OS/2 Basics - 12 of 17

Introduction

**OS/2 Basics**

Applications

Objects

Customizing

About Help

Multimedia and Games

Index

Practice

Do it!

OS/2 Think of the LaunchPad as a file cabinet where you can organize frequently used objects. The LaunchPad objects have drawers that slide out and contain additional objects.

Windows

Expert

Templates

Lockup Find Shut down Window list

Using the LaunchPad

1. Click on a LaunchPad object to open it.
2. Click on the drawer button to view other objects.
3. To protect your information from view, click on the Lockup button.
4. To locate an object, click on the Find button.
5. To shut down your computer, click on the Shut down button.
6. To see the list of open windows, click on the Window List button.

Lockup Find Shut down Window list

Lockup Find Shut down Window list

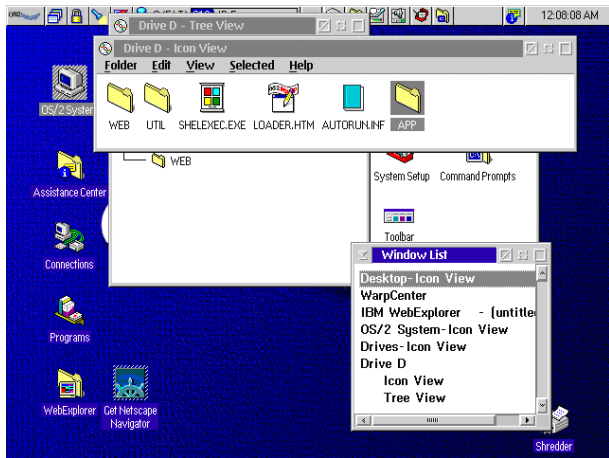
Lockup Find Shut down Window list

Return

Quit

- zpočátku společný projekt IBM a Microsoftu
- začátek 1987, GUI od 1988
- OS/2 2.0
- OS/2 3.0 Warp (1994)
- OS/2 4.0 Warp (1996)
- Warp uměl spouštět aplikace pro Windows 3.11

# 1988: OS/2



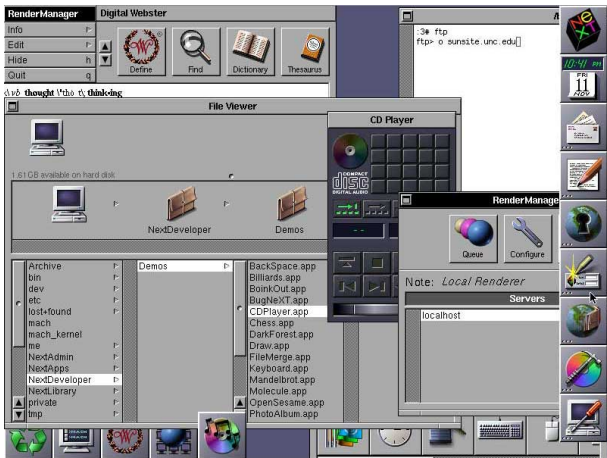
- zpočátku společný projekt IBM a Microsoftu
- začátek 1987, GUI od 1988
- OS/2 2.0
- OS/2 3.0 Warp (1994)
- OS/2 4.0 Warp (1996)
- Warp uměl spouštět aplikace pro Windows 3.11

# 1989-1995: NeXTstep



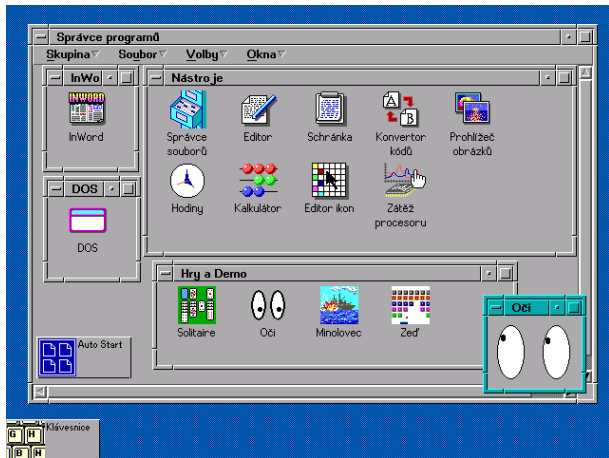
- pracovní stanice NeXT
- UNIXový OS založený na jádru Mach
- propracované GUI ovlivnilo např. Mac OS X
- zobrazování založeno na DPS (Display PostScript), nikoliv na X-Window

# 1989-1995: NeXTstep



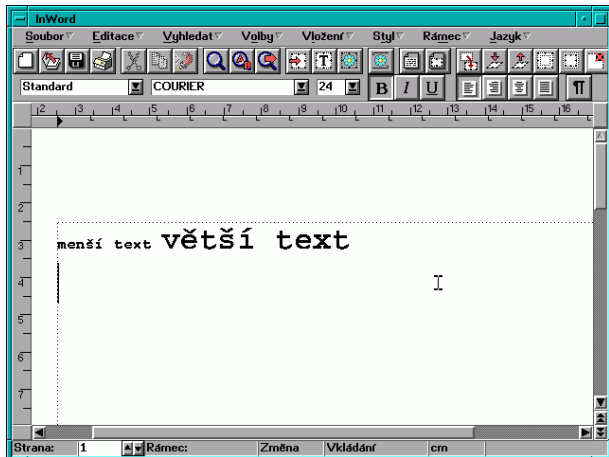
- pracovní stanice NeXT
- UNIXový OS založený na jádru Mach
- propracované GUI ovlivnilo např. Mac OS X
- zobrazování založeno na DPS (Display PostScript), nikoliv na X-Window

## cca 1995: InView



- grafická nadstavba DOSu od zlínské firmy Alcor
- žádný další software kromě programků od Alcor
- editor InWord - pouze vlastní formát .inw
- program brzy upadl v zapomnění - po právu

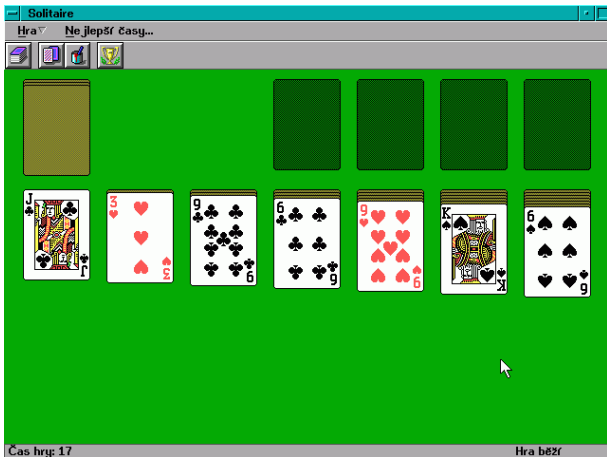
## cca 1995: InView



- grafická nastavení DOSu od zlínské firmy Alcor
- žádný další software kromě programů od Alcor
- editor InWord - pouze vlastní formát .inw
- program brzy upadl v zapomnění - po právu

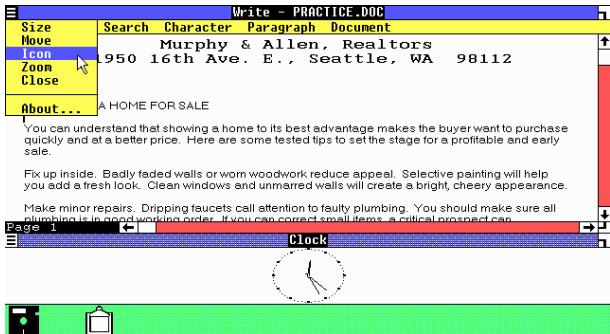


## cca 1995: InView



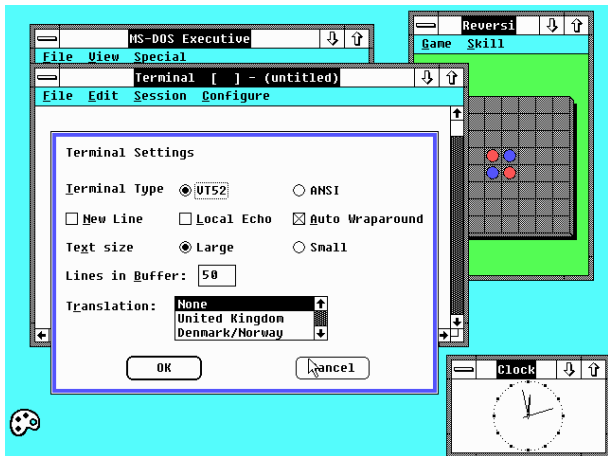
- grafická nadstavba DOSu od zlínské firmy Alcor
- žádný další software kromě programků od Alcor
- editor InWord - pouze vlastní formát .inw
- program brzy upadl v zapomnění - po právu

# Historie MS Windows 1/2



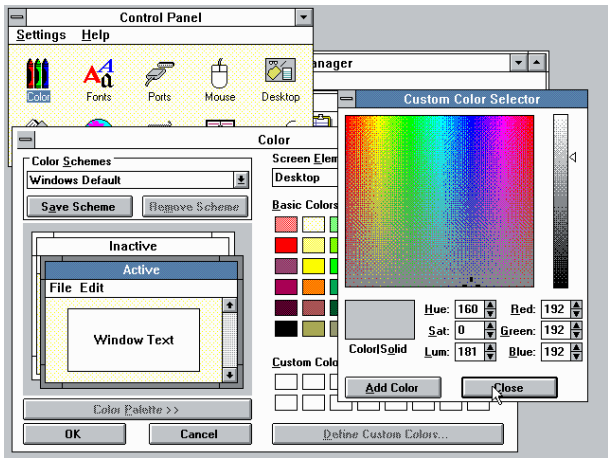
- Win 1.0 (1985)
- Win 2.0 (1987)
- Win 3.0 (1990)
- Win 3.11 (1992)
- WinNT 3.1 (1993)
- WinNT 3.51 (1995)
- Win95 (1995)
- WinNT 4.0 (1996)

# Historie MS Windows 1/2



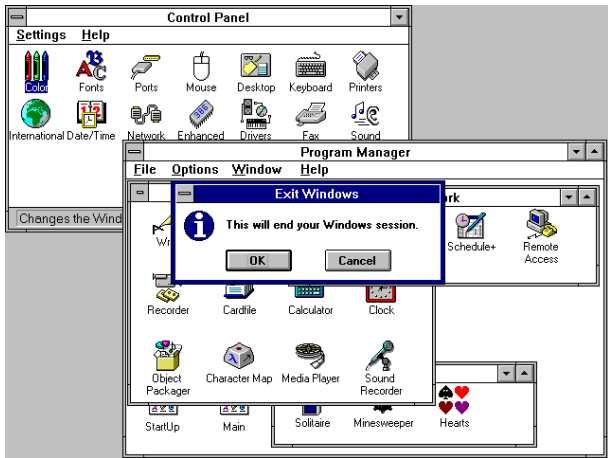
- Win 1.0 (1985)
- Win 2.0 (1987)
- Win 3.0 (1990)
- Win 3.11 (1992)
- WinNT 3.1 (1993)
- WinNT 3.51 (1995)
- Win95 (1995)
- WinNT 4.0 (1996)

# Historie MS Windows 1/2



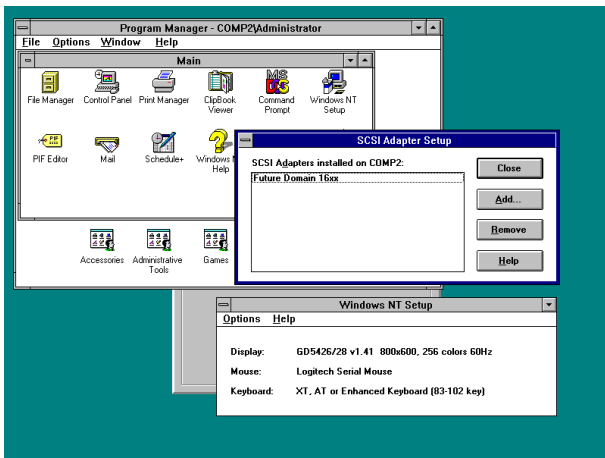
- Win 1.0 (1985)
- Win 2.0 (1987)
- Win 3.0 (1990)
- Win 3.11 (1992)
- WinNT 3.1 (1993)
- WinNT 3.51 (1995)
- Win95 (1995)
- WinNT 4.0 (1996)

# Historie MS Windows 1/2



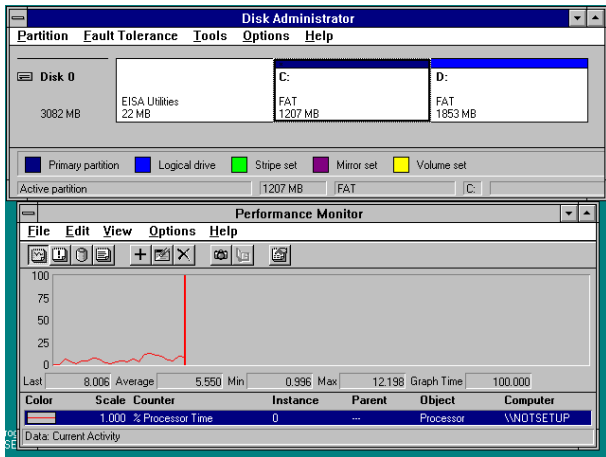
- Win 1.0 (1985)
- Win 2.0 (1987)
- Win 3.0 (1990)
- Win 3.11 (1992)
- WinNT 3.1 (1993)
- WinNT 3.51 (1995)
- Win95 (1995)
- WinNT 4.0 (1996)

# Historie MS Windows 1/2



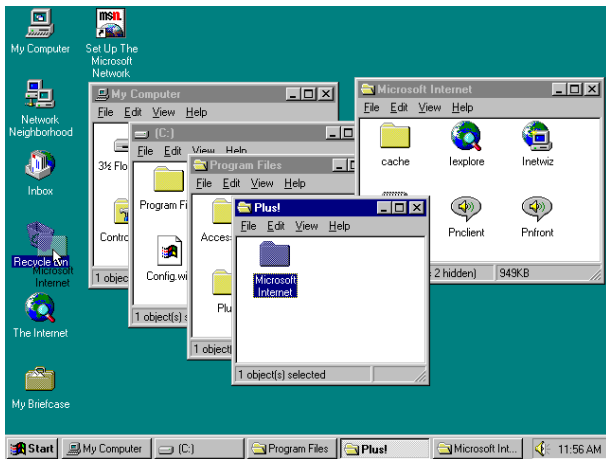
- Win 1.0 (1985)
- Win 2.0 (1987)
- Win 3.0 (1990)
- Win 3.11 (1992)
- WinNT 3.1 (1993)
- WinNT 3.51 (1995)
- Win95 (1995)
- WinNT 4.0 (1996)

# Historie MS Windows 1/2



- Win 1.0 (1985)
- Win 2.0 (1987)
- Win 3.0 (1990)
- Win 3.11 (1992)
- WinNT 3.1 (1993)
- WinNT 3.51 (1995)
- Win95 (1995)
- WinNT 4.0 (1996)

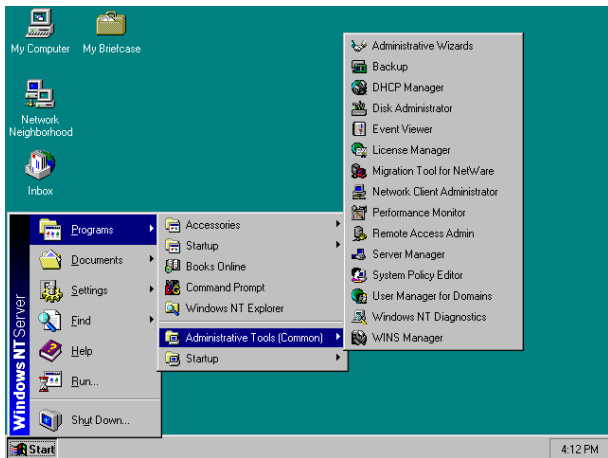
# Historie MS Windows 1/2



- Win 1.0 (1985)
- Win 2.0 (1987)
- Win 3.0 (1990)
- Win 3.11 (1992)
- WinNT 3.1 (1993)
- WinNT 3.51 (1995)
- Win95 (1995)
- WinNT 4.0 (1996)



# Historie MS Windows 1/2



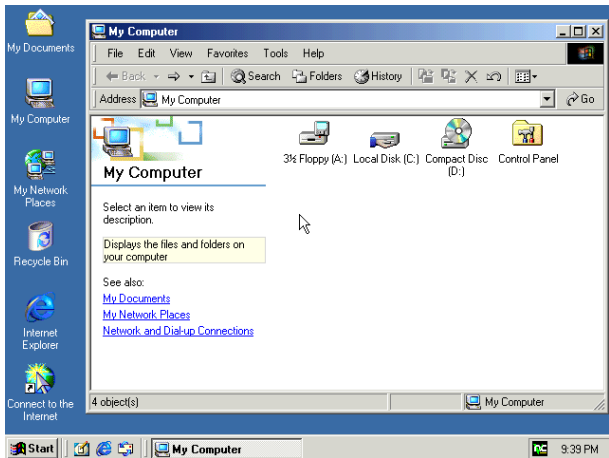
- Win 1.0 (1985)
- Win 2.0 (1987)
- Win 3.0 (1990)
- Win 3.11 (1992)
- WinNT 3.1 (1993)
- WinNT 3.51 (1995)
- Win95 (1995)
- WinNT 4.0 (1996)

# Historie MS Windows 2/2



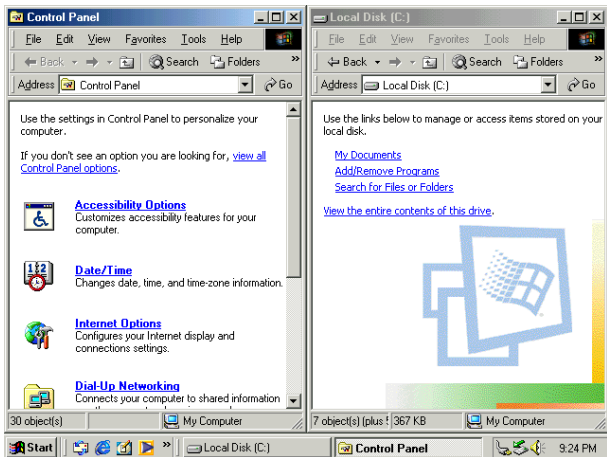
- Win98 (1998)
- Win2000 (2000)
- WinME (2000)
- WinXP (2001)
- Win Vista (2006)
- Windows 7 (2009)
- Windows 8 (2012)
- Windows 8.1 (2012)
- Windows 10 (2015)

## Historie MS Windows 2/2



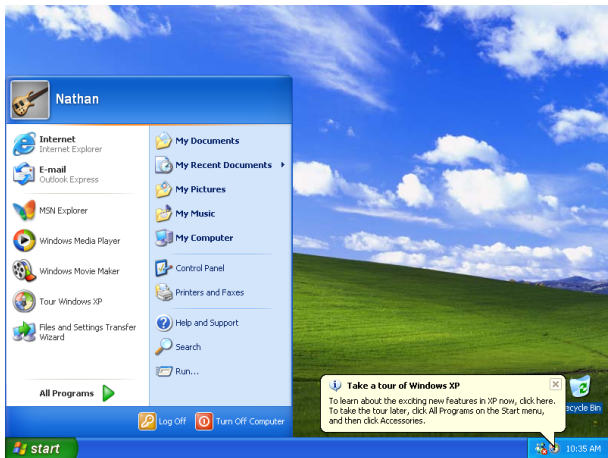
- Win98 (1998)
- Win2000 (2000)
- WinME (2000)
- WinXP (2001)
- Win Vista (2006)
- Windows 7 (2009)
- Windows 8 (2012)
- Windows 8.1 (2012)
- Windows 10 (2015)

# Historie MS Windows 2/2



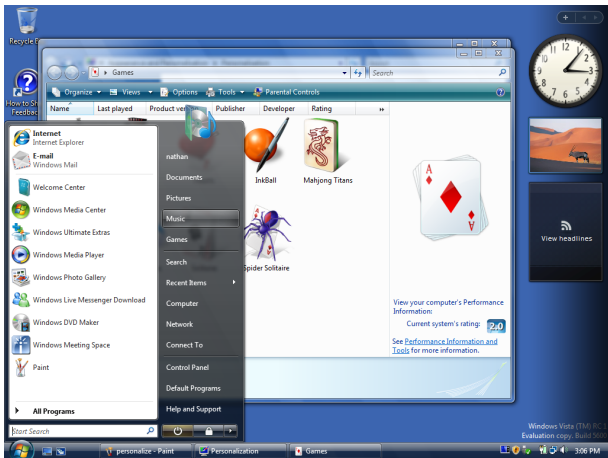
- Win98 (1998)
- Win2000 (2000)
- WinME (2000)
- WinXP (2001)
- Win Vista (2006)
- Windows 7 (2009)
- Windows 8 (2012)
- Windows 8.1 (2012)
- Windows 10 (2015)

# Historie MS Windows 2/2



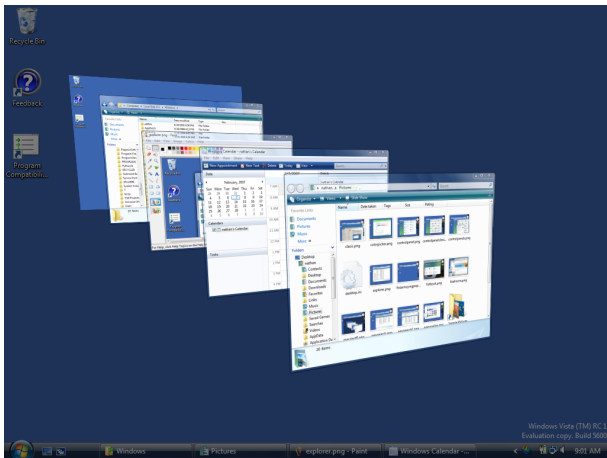
- Win98 (1998)
- Win2000 (2000)
- WinME (2000)
- WinXP (2001)
- Win Vista (2006)
- Windows 7 (2009)
- Windows 8 (2012)
- Windows 8.1 (2012)
- Windows 10 (2015)

# Historie MS Windows 2/2



- Win98 (1998)
- Win2000 (2000)
- WinME (2000)
- WinXP (2001)
- Win Vista (2006)
- Windows 7 (2009)
- Windows 8 (2012)
- Windows 8.1 (2012)
- Windows 10 (2015)

## Historie MS Windows 2/2



- Win98 (1998)
- Win2000 (2000)
- WinME (2000)
- WinXP (2001)
- Win Vista (2006)
- Windows 7 (2009)
- Windows 8 (2012)
- Windows 8.1 (2012)
- Windows 10 (2015)

# Historie MS Windows 2/2



- Win98 (1998)
- Win2000 (2000)
- WinME (2000)
- WinXP (2001)
- Win Vista (2006)
- Windows 7 (2009)
- Windows 8 (2012)
- Windows 8.1 (2012)
- Windows 10 (2015)

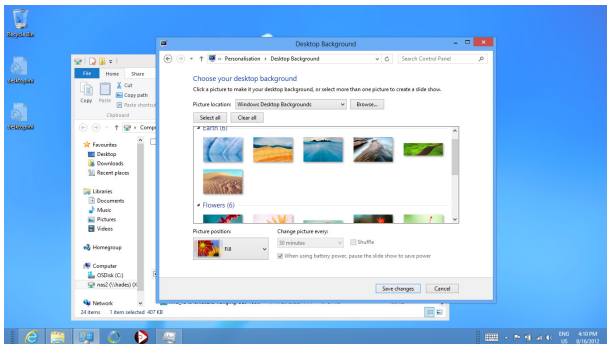


# Historie MS Windows 2/2



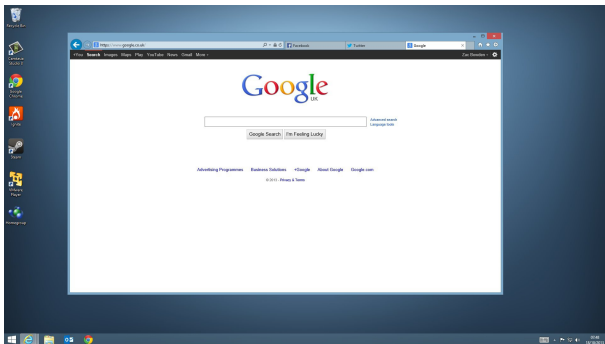
- Win98 (1998)
- Win2000 (2000)
- WinME (2000)
- WinXP (2001)
- Win Vista (2006)
- Windows 7 (2009)
- Windows 8 (2012)
- Windows 8.1 (2012)
- Windows 10 (2015)

# Historie MS Windows 2/2



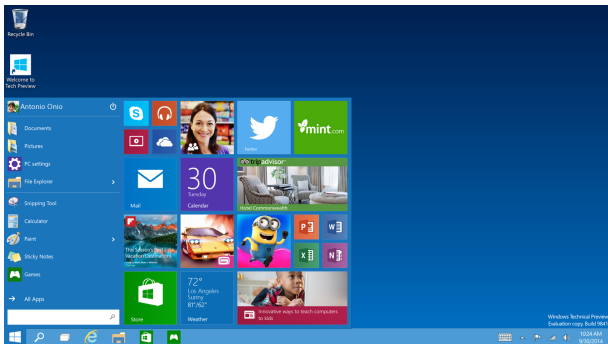
- Win98 (1998)
- Win2000 (2000)
- WinME (2000)
- WinXP (2001)
- Win Vista (2006)
- Windows 7 (2009)
- Windows 8 (2012)
- Windows 8.1 (2012)
- Windows 10 (2015)

# Historie MS Windows 2/2



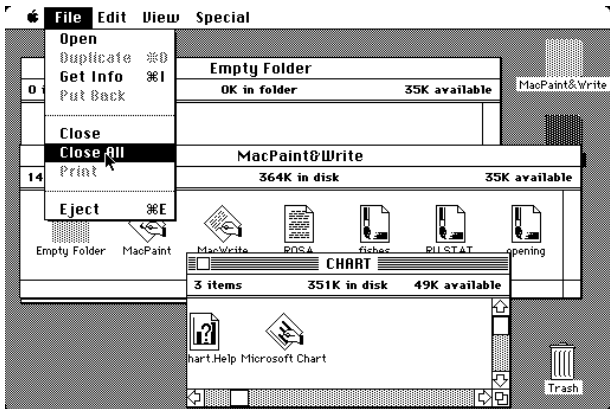
- Win98 (1998)
- Win2000 (2000)
- WinME (2000)
- WinXP (2001)
- Win Vista (2006)
- Windows 7 (2009)
- Windows 8 (2012)
- Windows 8.1 (2012)
- Windows 10 (2015)

## Historie MS Windows 2/2



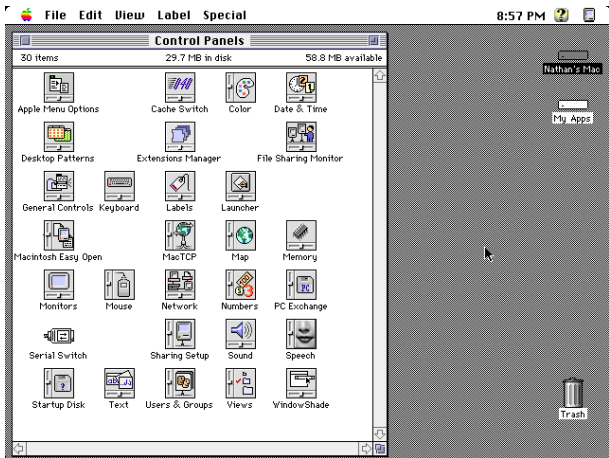
- Win98 (1998)
- Win2000 (2000)
- WinME (2000)
- WinXP (2001)
- Win Vista (2006)
- Windows 7 (2009)
- Windows 8 (2012)
- Windows 8.1 (2012)
- Windows 10 (2015)

# Výběr z historie Mac OS



- System 1.1 (1984)
- System 7 (1991-1997)
- Mac OS 8.1 (1997)
- Mac OS 9 (1999)
- Mac OS X 10.1 Puma (2001)
- Mac OS X 10.5 Leopard (2007)
- Mac OS X 10.6 Snow Leopard (2009)
- Mac OS X 10.10 Yosemite (2014)

# Výběr z historie Mac OS

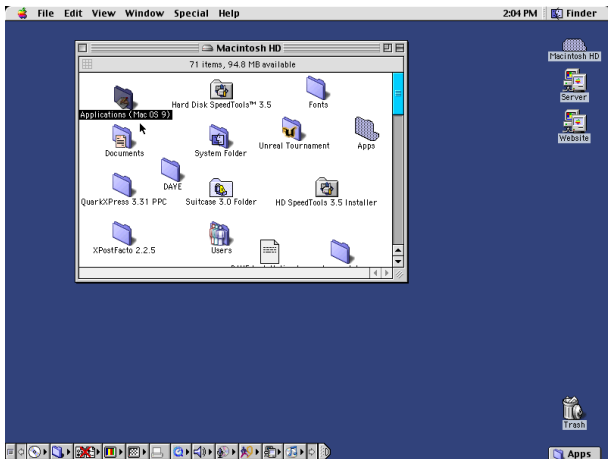


Mac OS 7.5

- System 1.1 (1984)
- System 7 (1991-1997)
- Mac OS 8.1 (1997)
- Mac OS 9 (1999)
- Mac OS X 10.1 Puma (2001)
- Mac OS X 10.5 Leopard (2007)
- Mac OS X 10.6 Snow Leopard (2009)
- Mac OS X 10.10 Yosemite (2014)



# Výběr z historie Mac OS

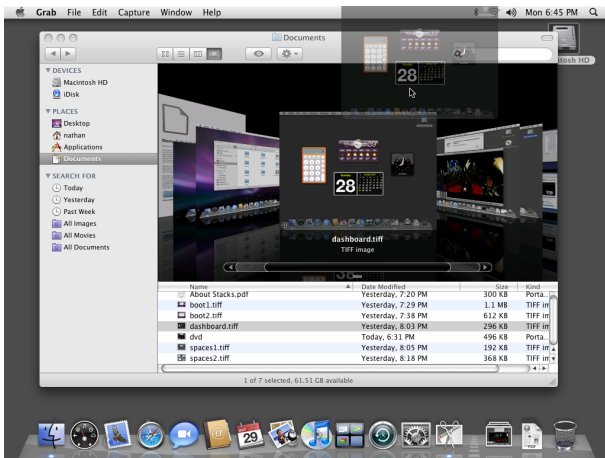


- System 1.1 (1984)
- System 7 (1991-1997)
- Mac OS 8.1 (1997)
- Mac OS 9 (1999)
- Mac OS X 10.1 Puma (2001)
- Mac OS X 10.5 Leopard (2007)
- Mac OS X 10.6 Snow Leopard (2009)
- Mac OS X 10.10 Yosemite (2014)





# Výběr z historie Mac OS



## Mac OS X 10.5

- System 1.1 (1984)
- System 7 (1991-1997)
- Mac OS 8.1 (1997)
- Mac OS 9 (1999)
- Mac OS X 10.1 Puma (2001)
- Mac OS X 10.5 Leopard (2007)
- Mac OS X 10.6 Snow Leopard (2009)
- Mac OS X 10.10 Yosemite (2014)



# Výběr z historie Mac OS

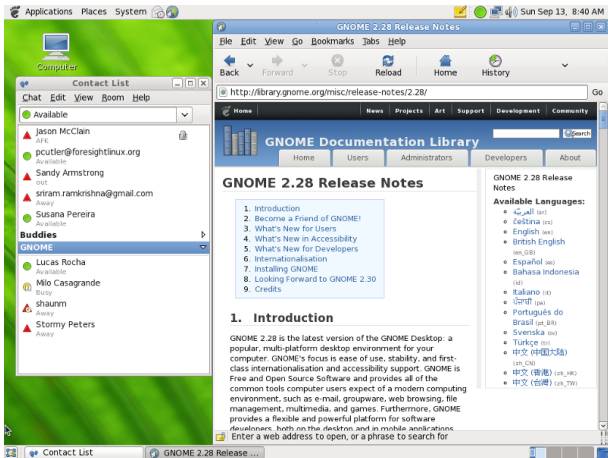


- System 1.1 (1984)
- System 7 (1991-1997)
- Mac OS 8.1 (1997)
- Mac OS 9 (1999)
- Mac OS X 10.1 Puma (2001)
- Mac OS X 10.5 Leopard (2007)
- Mac OS X 10.6 Snow Leopard (2009)
- Mac OS X 10.10 Yosemite (2014)





# X Window system



Gnome

- začátky na MIT v r. 1984
- model klient/server, komunikační protokol
- současná verze R11 již z r. 1987
- implementace pro UNIX, Windows, Mac OS X
- využíván různými desktopovými prostředími
- viz dále...









# X Window system

... a další:

LXDE, Enlightenment, OpenBox, FluxBox,  
IceWM, Razor-Qt, ...

- začátky na MIT v r. 1984
- model klient/server, komunikační protokol
- současná verze R11 již z r. 1987
- implementace pro UNIX, Windows, Mac OS X
- využíván různými desktopovými prostředími
- viz dále...

# Obsah

- 1 Historie GUI
  - Xerox Alto
  - PERQ
  - Xerox Star
  - Apple Lisa
  - VisiOn
  - GEM
  - OS/2
  - NeXTstep
  - InView
  - Windows
  - Mac OS
  - X-Window
- 2 **Koncepce GUI**
- 3 Nástroje pro tvorbu aplikací s GUI
- 4 Desktopové prostředí
- 5 Vzdálená plocha

# Paradigma WIMP

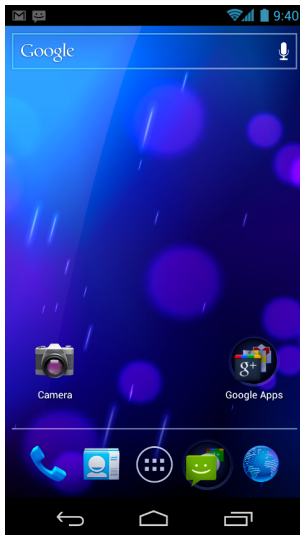
- Windows, Icons, Menus, Pointing device
- pro všechna GUI již přes 30 let (!) podobné principy ovládání a zobrazování
- pojem virtuálního „**pracovního stolu**“ (*desktop*)
- **okno** (*window*) sdružuje související objekty (např. patřící jedné aplikaci)
- zobrazují se různé **ovládací prvky** (*controls, widgets*)
  - tlačítka (*button*)
    - menu (*pop up menu, drop-down menu*)
    - vypínače (*checkbox*)
    - vícepohové 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 (G)UI

- 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*)

# UI pro tablety a smartphony

- koncept oken ustupuje
  - malá obrazovka, práce s okny nepohodlná
- elementy ovladatelné prstem
  - *multi-touch*: zoom, rotace atd..
- XML layouts (Android)
- QML (Qt Modeling/Meta Language)
- ovládání pomocí dalších zařízení (senzor natočení, gyroskop)



# UI pro tablety a smartphony

- koncept oken ustupuje
  - malá obrazovka, práce s okny nepohodlná
- elementy ovladatelné prstem
  - *multi-touch*: zoom, rotace atd..
- XML layouts (Android)
- QML (Qt Modeling/Meta Language)
- ovládání pomocí dalších zařízení (senzor natočení, gyroskop)

```
import QtQuick 1.0

Rectangle {
    id: canvas
    width: 200
    height: 200
    color: "blue"

    Image {
        id: logo
        source: "pics/logo.png"
        anchors.centerIn: parent
        x: canvas.height / 5
    }
}
```

## Program řízený událostmi 1/3

- *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*)



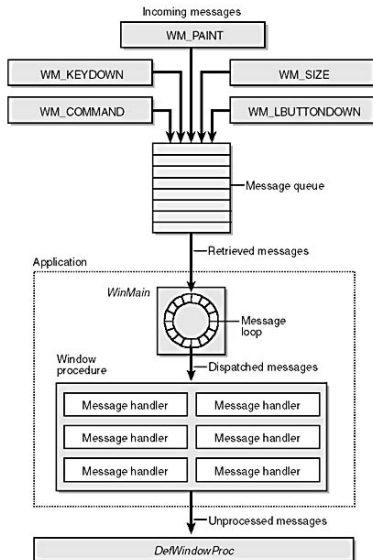
## Program řízený událostmi 2/3

- vytvoření funkčního GUI spočívá v
  - 1 **návrhu rozložení ovládacích prvků**
  - 2 **implementaci reakcí na události** v podobě funkcí - *event handlerů*
  - 3 **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 3/3

- **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:** funkce `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`

# Události ve Win32



## Objektově orientovaný přístup 1/3

- **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.
- např. 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 2/3

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

## Objektově orientovaný přístup 3/3

- např. Borland Delphi (C++ Builder)
  - *property* - datová složka třídy, při jejímž čtení, resp. zápisu se volají určité funkce
  - např. barva tlačítka:

```
Button.color=0x3366AA;
```

- $\implies$  automaticky se volá funkce, která skutečně změní barvu na obrazovce
- inicializace vlastností ovládacích prvků, registrace handlerů **nejsou součástí kódu** (uloženo v souborech `.dfm` se speciální syntaxí)
- další data (např. ikony atd.) uloženy v souboru zdrojů (*resources*)

# Příklad vývoje GUI v C#

```
using System;
using System.Drawing;
using System.Windows.Forms;

namespace ProgrammingWinApps
{
    public class HelloWorld : System.Windows.Forms.Form
    {
        private Button btn;

        public HelloWorld( )
        {
            Text = "Hello World";

            btn = new Button( );
            btn.Location = new Point(50,50);
            btn.Text = "Goodbye";
            btn.Click += new System.EventHandler(btn_Click);
            Controls.Add(btn);
        }

        static void Main( )
        {
            Application.Run(new HelloWorld( ));
        }

        private void btn_Click(object sender, EventArgs e)
        {
            Application.Exit( );
        }
    }
}
```

Vytvoření tlačítka

Registrace handleru

Implementace handleru

- ovládací prvky generují události (*publish*), k jejichž zpracování se může **přihlásit** (*subscribe*) jedna či více tříd
- přihlášení k události = registrace event handleru prostřednictvím **delegáta** (*delegate*) - v podstatě objektově implementovaného ukazatele na metodu

## Prostředky pro rychlý vývoj aplikací

- **RAD** - *Rapid Application Development*
- vizuální návrh dialogových oken, rozvržení ovládacích prvků
- vizuální nastavování jejich vlastností (*properties*) a registrace handlerů (*Object Inspector, Property Inspector*)
- začátky: MS Visual Basic
- nyní stejný princip u mnoha dalších vývojových prostředí
  - Microsoft Visual Studio (VB, C#)
  - MonoDevelop (Mono = C# a .NET pro UNIX/Linux)
  - Borland Delphi, Borland C++ Builder, Lazarus
  - Glade (designér dialogových oken pro GTK+)
  - Qt Creator
  - KDevelop
  - MATLAB GUIDE
  - ...



# Příklad RAD - MS Visual Studio 2008

The screenshot displays the Microsoft Visual Studio 2008 Express Edition interface for a Windows Forms application named 'WindowsFormsApplication1'. The main window is in Design mode, showing a form titled 'Form1' with the following controls:

- A button labeled 'button1'.
- A group box labeled 'groupBox1' containing a checkbox labeled 'checkBox1'.
- A list box labeled 'listBox1'.

The Toolbox on the left shows 'All Windows Forms' selected, with various controls like Pointer, Button, CheckBox, etc. The Solution Explorer on the right shows the project structure, including 'Form1.cs' and 'Form1.Designer.cs'. The Properties window on the right shows the properties for 'Form1', including:

Property	Value
MaximumSize	0; 0
MinimizeBox	True
MinimumSize	0; 0
Opacity	100%
Padding	0; 0; 0
RightToLeft	No
RightToLeftLayout	False
ShowIcon	True
ShowInTaskbar	True
Size	389; 412
SizeGripStyle	Auto
StartPosition	WindowsDefaultLocat
Tag	
Text	Form1

# Obsah

- 1 Historie GUI
  - Xerox Alto
  - PERQ
  - Xerox Star
  - Apple Lisa
  - VisiOn
  - GEM
  - OS/2
  - NeXTstep
  - InView
  - Windows
  - Mac OS
  - X-Window
- 2 Koncepce GUI
- 3 Nástroje pro tvorbu aplikací s GUI**
- 4 Desktopové prostředí
- 5 Vzdálená plocha

## Přehled prostředků pro vývoj GUI

- mnoho tzv. toolkitů (knihovny, API, vývojové nástroje) pro tvorbu GUI
- Borland TurboVision („GUI“ v textovém režimu), ObjectWindows, VCL
- MFC (*Microsoft Foundation Classes*) - objektový interface k Win32 pro C++
- Windows Forms (API pro GUI v .NET)
- JFC (*Java Foundation Classes*), Swing

### snaha o přenositelný kód (*cross-platform*)

- GTK+
- Qt
- FLTK
- FOX

# GTK+

- Gimp ToolKit
- multiplatformní knihovna pro C, API rovněž pro C++, Ruby, Python, Javu, .NET, PHP, Perl
- používána v desktopovém prostředí GNOME, v profesionálních *closed-source* komerčních produktech pro Linux (Adobe Reader, VMWare, Nero for Linux, ...)
- obsahuje vizuální designér Glade
- GTK+ v X11 používá jako interface k nízkoúrovňovým funkcím grafiky knihovnu **GDK** (*Gimp Drawing Kit*)
  - vykreslování geometrických primitiv, bitmap, fontů, implementace funkce drag&drop

# GTK+ Hello world

```
#include <gtk/gtk.h>

int main (int argc, char *argv[])
{
    GtkWidget *window;
    GtkWidget *label;

    gtk_init (&argc, &argv);

    /* create the main, top level, window */
    window = gtk_window_new (GTK_WINDOW_TOPLEVEL);

    /* give it the title */
    gtk_window_set_title (GTK_WINDOW (window), "Hello World");

    /* Connect the destroy signal of the window to gtk_main_quit
     * When the window is about to be destroyed we get a notification and
     * stop the main GTK+ loop
     */
    g_signal_connect (window, "destroy",
                     G_CALLBACK (gtk_main_quit), NULL);

    /* Create the "Hello, World" label */
    label = gtk_label_new ("Hello, World");

    /* and insert it into the main window */
    gtk_container_add (GTK_CONTAINER (window), label);

    /* make sure that everything, window and label, are visible */
    gtk_widget_show_all (window);

    /* start the main loop, and let it rest there until the application is closed */
    gtk_main ();

    return 0;
}
```

# Qt

- vyslovuje se „*cute*“
- multiplatformní prostředí vyvíjené firmou Trolltech, pak Nokia Qt Development Frameworks, nyní Digia
- původně pro C++, API rovněž pro Ruby, Python, PHP, Perl, Pascal
- obsahuje tzv. *meta object compiler* (moc) - preprocesor C++ kódu, umožňující zahrnout do C++ další funkce nedostupné v standardu
- využívané desktopovým prostředím KDE, programy Opera, Skype, Google Earth, VirtualBox, Mathematica
- obsahuje vizuální návrhář GUI **Qt designer** a nověji i kompletní IDE **Qt Creator**

# Qt Hello world

```
#include <QtGui>

int main(int argc, char *argv[])
{
    QApplication app(argc, argv);
    QLabel label("Hello, world!");
    label.show();
    return app.exec();
}
```

# Qt

## Signály a sloty 1/2

- systém obsluhy událostí alternativní k použití obecných *callback* funkcí jako event-handlerů
- podporován pomocí kompilátoru moc, nová klíčová slova `signals`, `slots`, `emit`
- **signál:** deklarovaný (z hlediska uživatele) jako speciální prototyp metody s lib. formálními parametry a návratovým typem `void`

```
signals:
```

```
void valueChanged(int newValue);
```

- **slot:** (z hlediska uživatele) libovolná metoda deklarovaná v sekci třídy označené „`slots`“.

```
public slots:
```

```
void ProcessNewValue(int value);
```

- signály mohou být propojeny se sloty se stejným prototypem (pomocí metody `connect`)



# Qt

## Signály a sloty 2/2

- vyvolání (emise) signálu: `emit` + „volání“ metody s konkrétními argumenty

```
emit valueChanged(10);
```

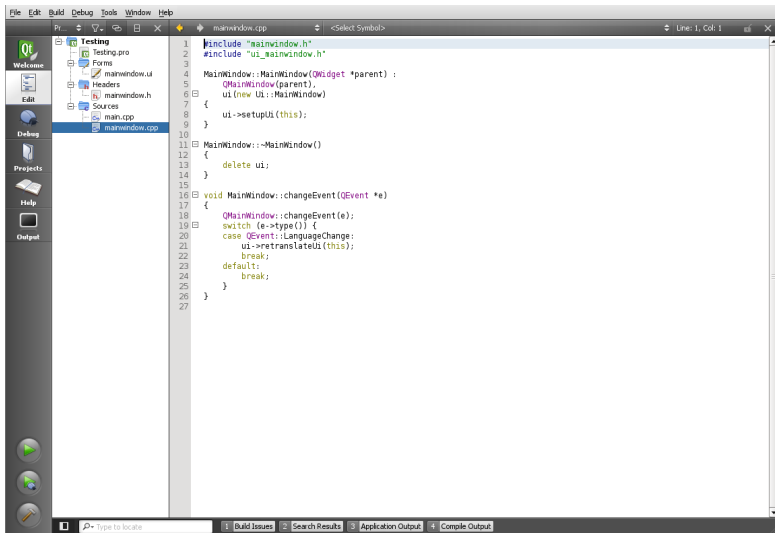
- následek: s danými argumenty (zde číslo 10) se zavolají všechny sloty napojené na daný signál

### Výhody:

- typově bezpečné volání
- možnost navázat více slotů na stejný signál, více signálů na stejný slot atp.
- korektní ošetření odpojení slotů při zániku objektů
- mnoho předdefinovaných signálů i slotů k různým prvkům GUI
  - např. tlačítko: signál `clicked` atd.

# Qt Creator

## Editace zdrojového kódu

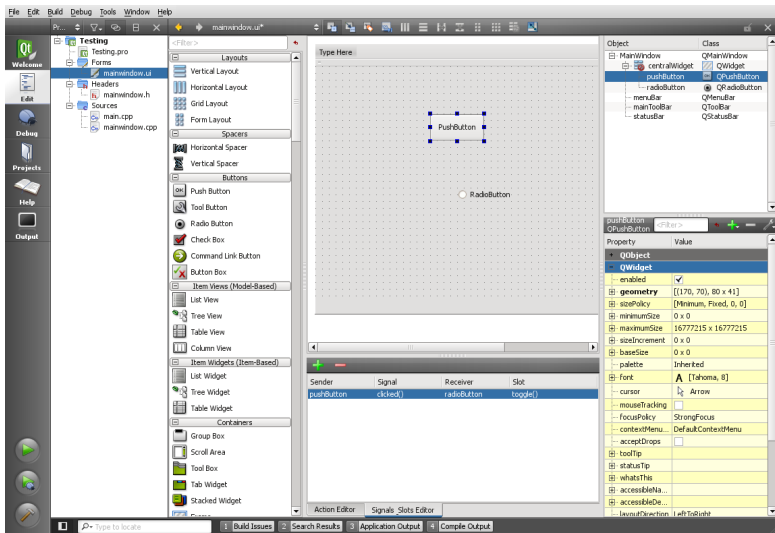


The screenshot shows the Qt Creator IDE interface. On the left is a sidebar with icons for Welcome, Edit, Debug, Projects, Help, and Output. The main area is divided into a Project Explorer on the left and a Code Editor on the right. The Project Explorer shows a project named 'Testing' with a file tree containing 'mainwindow.ui', 'mainwindow.h', 'main.cpp', and 'mainwindow.cpp'. The Code Editor displays the source code for 'mainwindow.cpp'.

```
1 |include "mainwindow.h"  
2 |include "ui_mainwindow.h"  
3  
4 MainWindow::MainWindow(QWidget *parent) :  
5     QMainWindow(parent),  
6     ui(new Ui::MainWindow)  
7 {  
8     ui->setupUi(this);  
9 }  
10  
11 MainWindow::~MainWindow()  
12 {  
13     delete ui;  
14 }  
15  
16 void MainWindow::changeEvent(QEvent *e)  
17 {  
18     QMainWindow::changeEvent(e);  
19     switch (e->type()) {  
20     case QEvent::LanguageChange:  
21         ui->retranslateUi(this);  
22         break;  
23     default:  
24         break;  
25     }  
26 }  
27
```

# Qt Creator

## Qt Designer



# Qt Creator

## Signály a sloty

The screenshot shows the Qt Creator IDE with a project named 'Testing'. The main window is 'mainwindow.ui'. The central area displays a visual representation of the signal-slot connection. A blue box labeled 'QPushButton' with the signal 'clicked()' is connected to a red box labeled 'radioButton' with the slot 'toggle()'. The 'radioButton' is further connected to a red box labeled 'MainWindow' with the slot 'setDockNestingEnabled(boo)'. Below the visual area is a table summarizing the connection:

Sender	Signal	Receiver	Slot
pushButton	clicked()	radioButton	toggle()
radioButton	clicked(boo)	MainWidow	setDockNestingEnabled(boo)

On the right side, the 'Object' and 'Class' panels show the widget hierarchy: MainWidow (QMainWindow) contains centralWidget (QWidget), pushButton (QPushButton), radioButton (QPushButton), menuBar (QMenuBar), mainToolBar (QToolBar), and statusBar (QStatusBar). The 'Property' panel shows the 'enabled' property of the selected 'radioButton' widget is checked.

# FLTK

- FLTK - *Fast Light ToolKit* (vyslovuje se „fulltick“)
- pro C++
- obsahuje interface k OpenGL
- obsahuje vývojové prostředí FLUID (*FLTK User Interface Designer*)

```
#include <FL/Fl.H>
#include <FL/Fl_Window.H>
#include <FL/Fl_Button.H>

int main(int argc, char *argv[]) {
    Fl_Window* w = new Fl_Window(330, 190);
    new Fl_Button(110, 130, 100, 35, "Okay");
    w->end();
    w->show(argc, argv);
    return Fl::run();
}
```

# FOX

- FOX - *Free Objects for X*
- pro C++, API i pro Python, Ruby, Eiffel
- obsahuje styl Windows95, který je k dispozici pro Windows i X Window (UNIX)

```
#include "fx.h"

int main(int argc, char *argv[]) {
    FXApp application("Hello", "FoxTest");
    application.init(argc, argv);
    FXMainWindow *main=new FXMainWindow(&application, "Hello", NULL, NULL, DECOR_ALL);
    new FXButton(main, "&Hello, World!", NULL, &application, FXApp::ID_QUIT);
    application.create();
    main->show(PLACEMENT_SCREEN);
    return application.run();
}
```

# Obsah

- 1 Historie GUI
  - Xerox Alto
  - PERQ
  - Xerox Star
  - Apple Lisa
  - VisiOn
  - GEM
  - OS/2
  - NeXTstep
  - InView
  - Windows
  - Mac OS
  - X-Window
- 2 Koncepce GUI
- 3 Nástroje pro tvorbu aplikací s GUI
- 4 Desktopové prostředí**
- 5 Vzdálená plocha

## Desktopové prostředí ve Windows 1/2

- **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*)  
⇒ význam hlášky „85% volných prostředků“
- od Windows XP modernizace - GDI+



## Desktopové prostředí ve Windows 2/2

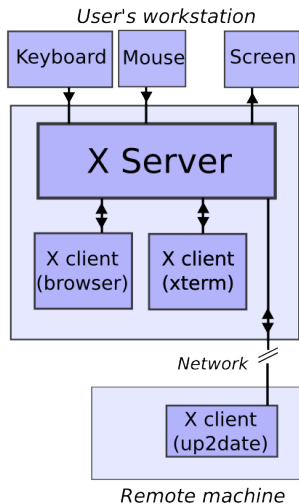
- ve Windows Vista, Windows 7 a 8 – náhrada:  
**DWM** (Desktop Window Manager)
- kompozitní manažer oken (viz dále)
- „staré“ aplikace používající GDI běží pod DWM (GDI již nekomunikuje přímo s ovladačem grafiky a není hardwarově akcelerováno)
- akcelerace 2D grafiky - Direct2D (DirectDraw součástí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

# X Window system

- 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 Window 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, ...)
  - 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 atd.)
- okna uspořádána v hierarchii



## X Window 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, ...)
  - 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 atd.)
- okna uspořádána v hierarchii
- pokud se aplikace zobrazuje na počítači, kde běží  
⇒ lze použít DRI (*Direct Rendering Infrastructure*)
  - aplikace využívá hardware bez komunikace s X serverem
  - 3D akcelerace atd.

# X servery

- **X.Org**

- oficiální X server X.Org Foundation, která v podstatě vede (spravuje) vývoj X Window systému
- implicitní pro většinu UNIXových systémů (Linux, Solaris, FreeBSD)
- nástupce projektu **XFree86**

- **Cygwin/X** - X server v Cygwin

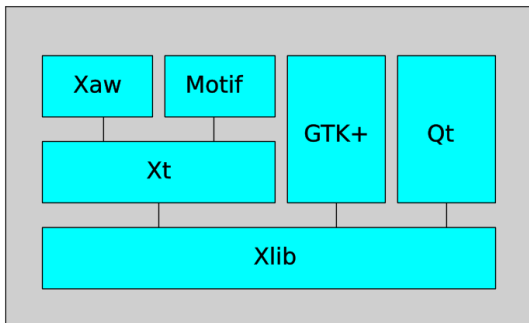
- **XMing** - implementace X serveru pro Windows, vychází z
- další komerční implementace pro Windows (WinaXe Plus SSH atp.)

- **X11.app** - systém vestavěný v Mac OS X postavený na X serveru **XQuartz**

- jeho Aqua GUI **nativně nepoužívá X**, ale systém **Quartz** inspirovaný DPS systému NeXTstep)

# Xlib

- knihovna implementující komunikaci s X serverem
- aplikace běžící pod X používá Xlib
- GUI toolkity izolují programátora od Xlib, samy ji však používají



## X Window Manager

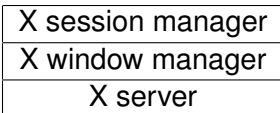
- manažer oken běžící nad X serverem
- vykresluje rámečky kolem oken (*reparenting* - pokud chceme po X serveru okno, WM jej vloží do nadřazeného okna (*parent window*), které obsahuje ještě titulek a rámeček)
- definuje interakci uživatele s okny (přepínání, změna aktivního okna (*focus*) atd.)
- ošetřuje překrývání oken (*stacking WM*)
- **kompozitní WM** (*compositing WM*) - vykresluje okna do **samostatných** offline frame-bufferů, poté složí výsledek dohromady (umožňuje průhlednost, kombinovat 2D a 3D-akcelerované objekty atd.)
- populární WM: Metacity, Kwin, tw, Xfwm, IceWM, Enlightenment, AfterStep, ...
- populární kompozitní WM: Compiz, Beryl, Compiz Fusion, Kwin 4.0, Metacity 2.20+





## Další komponenty X desktopu

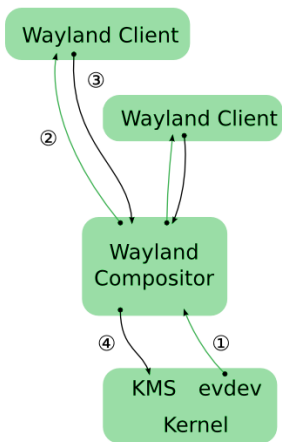
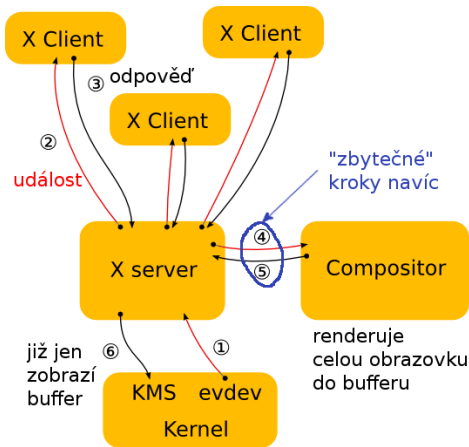
- **X session manager** - manažer desktopu (xsm, ksmserver pro KDE atd.)
  - má na starosti „globální“ stav celého desktopu
- desktopové prostředí má svůj implicitní window manager, ale obvykle může používat i jiný (Metacity v KDE apod.), pokud splňuje standard ICCCM (*Inter-Client Communication Conventions Manual*)
- **X display manager** - manažer přihlášení (KDM, GDM, ...)



# Náhrady systému X Window

- systém je příliš složitý a již neslouží svému účelu
  - většina aplikací zobrazuje na lokálním displeji
  - Qt, GTK atd. samy renderují fonty
  - kompozitní window managery spravují celý desktop (fullscreen)
  - X server vlastně jen zobrazuje vyrenderované bitmapy
- snaha o nový, jednodušší a moderní systém
  - **Wayland** ... podpora KDE, GNOME a většiny distribucí
  - **Mir** ... vyvíjen firmou Canonical pro Ubuntu jeho Unity 8
  - oba používají pro rendering **EGL**, tj. **OpenGL ES** (OpenGL for *E*mbedded Systems)

# X Window system vs. Wayland



# Obsah

- 1 Historie GUI
  - Xerox Alto
  - PERQ
  - Xerox Star
  - Apple Lisa
  - VisiOn
  - GEM
  - OS/2
  - NeXTstep
  - InView
  - Windows
  - Mac OS
  - X-Window
- 2 Koncepce GUI
- 3 Nástroje pro tvorbu aplikací s GUI
- 4 Desktopové prostředí
- 5 **Vzdálená plocha**

# Protokoly pro vzdálenou plochu

- model klient-server
- **RDP** (*Remote Desktop Protocol*)
  - vzdálená plocha Windows
  - umí na vzdálený terminál přeměrovat 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 přenášení komprimované jako JPEG
  - umí přeměrovat i tisk
- **XDMCP** (*X Display Manager Control Protocol*)
  - (nekomprimovaný) přenos celého X desktopu včetně přihlašovací obrazovky

# Literatura

-  D. Hurwitz, J. Liberty: *Programming .NET Windows Applications*. O'Reilly, 2003.
-  J. Prosis: *Programming Windows with MFC*. Microsoft Press, 1999.
-  <http://toastytech.com/guis/index.html>
-  <http://www.gtk.org>
-  <http://qt.nokia.com>
-  <http://www.fltk.org>
-  <http://www.x.org>
-  <http://www.wikipedia.org>