Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

# Bakalářská práce

# Analýza vývoje uživatelských rozhraní

Plzeň 2017 Markéta Bošková

# Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracovala samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 2. května 2017

Markéta Bošková

# Poděkování

Děkuji Ing. Richardu Lipkovi, Ph.D. za cenné rady a čas, který mi věnoval při vedení mé bakalářské práce.

# Abstract

This bachelor thesis provides analysis and overview of operating systems user interface development in history. It explains basic concepts related to graphical interface and various ways of its implementation. Part of the thesis is detailed description of every system interface and its individual functions and components with their first occurrence and possibilities they offer to users. The next section contains introduction to an aplication programming interfaces of the analyzed systems and code examples for creating individual interface components. In the conclusion is evaluation of mentioned API depending on their implementation difficulty. This complex comparison is created by comparing created programs in those API.



# Abstrakt

Tato bakalářská práce se zabývá analýzou a vývojem uživatelských rozhraní operačních systémů v historii. Vysvětluje základní pojmy související s grafickým rozhraním a různé způsoby jeho implementace. Součástí je detailní popis každého prostředí systému a jeho jednotlivých funkcí a komponent, u kterých je uveden jejich první výskyt a možnosti, které nabízejí uživatelům. Následuje seznámení s programovým rozhraním analyzovaných systémů a ukázky kódu pro tvorbu jednotlivých komponent rozhraní v těchto API. V závěru práce je zhodnocení zmíněných API v závislosti na jejich náročnosti při použití. Srovnání složitosti je vytvořeno na základě porovnání vytvořených programů v uvedených API.

# Obsah

# 1 Úvod

Grafická uživatelská rozhraní se stala důležitou a pro většinu lidí neoddělitelnou součástí každodenního života a práce. Tento typ rozhraní je dnes standardem, který se očekává ve vysoké kvalitě a měl by co nejvíce zvyšovat efektivitu práce na počítači. Moderní operační systémy obvykle obsahují vlastní grafické rozhraní, které se v průběhu let neustále vyvíjelo. Ve většině případů tak značně ulehčovalo práci a vylepšilo uživatelský zážitek.

Tato práce má za cíl vytvořit ucelený přehled jednotlivých verzí operačních systémů využívaných v dnešní době na osobních počítačích a postupného vývoje jejich uživatelských rozhraní. Součástí jsou také první systémy využívající grafické prostředí, které se staly průkopníky v této oblasti a udaly směr vývoje nadcházejících systémů.

V první části práce se věnuji rozdělení uživatelských rozhraní na jednotlivé druhy a poté obecným pojmům souvisejících se způsoby implementace grafických rozhraní do různých operačních systémů. Poté se zaměřím na vývoj grafického prostředí uvedených systémů a jeho jednotlivých komponent. Dále se seznámím a popíšu jejich aplikační programové rozhraní, tedy Windows API, knihovny GTK+ a Qt a Cocoa API. Pro tato API poté naprogramuji jednoduché grafické rozhraní. Ta pak mezi sebou porovnám v závislosti na jejich náročnosti použití.

# 2 Druhy uživatelských rozhraní

Uživatelské rozhraní představuje způsob, jakým uživatel komunikuje s určitým systémem. Hlavním cílem dobře navrženého uživatelského rozhraní je umožnit uživateli pracovat rychle a efektivně. Dále uvedu příklady nejznámějších typů uživatelských rozhraní.

## 2.1 Textové uživatelské rozhraní

Textové uživatelské rozhraní je uživatelské rozhraní, které pracuje v textovém režimu. Obrazovka je pevně rozdělena na jednotlivé úseky (sloupce, řádky). Vyskytují se zde prvky obvyklé pro grafické uživatelské rozhraní, jako jsou okna, tlačítka nebo seznamy. Programy s textovým uživatelským rozhraním se vyskytovaly například v operačním systému DOS v souborovém manažeru Norton Commander nebo v unixových systémech v programu Midnight Commander. Na těchto příkladech lze ukázat, že v čistě textovém režimu můžeme okna simulovat.

Textové uživatelské rozhraní tedy neznamená příkazovou řádku, do které se pouze zadávají příkazy. Příkazový řádek je další druh uživatelského rozhraní, ve kterém uživatel komunikuje s počítačem pomocí příkazů zapisovaných do příkazového řádku.

## 2.2 Grafické uživatelské rozhraní

Grafické uživatelské rozhraní, dále jen GUI (z anglického Graphic User Interface), bylo navrženo především kvůli usnadnění práce s počítačem. Při práci s grafickým rozhraním si uživatel nemusí pamatovat žádné příkazy, samotný monitor se totiž stává místem pro zadávání vstupu od uživatele. Monitor zobrazuje různé grafické objekty v podobě ikon a dalších vstupních prostředků, jako jsou tlačítka nebo posuvníky. Pomocí klávesnice nebo polohovacího zařízení (např. myši) může uživatel přímo manipulovat s těmito objekty. Vzájemná komunikace mezi uživatelem a počítačem se tak stává více osobní a jednoduchá [13].

### 2.2.1 WIMP

WIMP (windows, icons, menus, pointer) označuje grafické uživatelské rozhraní a způsob interakce mezi počítačem a uživatelem založené právě na těchto čtyřech prvcích [11]. WIMP rozhraní bylo poprvé představeno na počítači Xerox Star a o několik let později na jeho populárnějším následníkovi Apple Macintosh. Poté se WIMP rozšířil mezi další počítače a jejich operační systémy a stal se dominantním uživatelským rozhraním, které používáme dodnes. Vizuální prvky WIMP rozhraní zůstávají zachovány v různých aplikacích a verzích operačních systémů, tudíž dnešní aplikace mají tendenci vypadat velice podobně jako aplikace na prvních počítačích s grafickým uživatelským rozhraním. Rozdíl je vidět v propracovanosti komponent, v průběhu vývoje byly stále více realisticky vypadající, například díky použití stínování tlačítek. Ne všechna grafická uživatelská rozhraní používají stejné prvky jako WIMP, tedy ne všechny GUI jsou WIMP. WIMP rozhraní dodržuje následující pravidla:

1. Okno sdružuje související objekty samostatně běžícího programu, odděleného od ostatních programů.
2. Ikona funguje jako zkratka k akci, kterou počítač vykonává (například spuštění programu).
3. Menu je textová nebo na ikonách založená nabídka, ze které vybíráme a spouštíme funkce programu.
4. Ukazatel/polohovací zařízení je reprezentován kurzorem na obrazovce, který slouží pro zobrazení pohybu hardware zařízení, typicky myši.

### 2.2.2 post-WIMP

Další generací uživatelských rozhraní jsou takzvaná post-WIMP uživatelská rozhraní. Tato rozhraní mají pokročilejší, respektive odlišný způsob ovládání než WIMP rozhraní a hodí se pro specifické účely. Bývají typicky založena na rozeznávání hlasu nebo gest, tedy ovládání pomocí více smyslů. Příkladem je použití ve virtuální realitě nebo operačních systémech na dotykových mobilech nebo tabletech, které sice obsahují ikony, ale často postrádají menu a okna. Také není potřeba kurzor, který je součástí WIMP rozhraní [8].

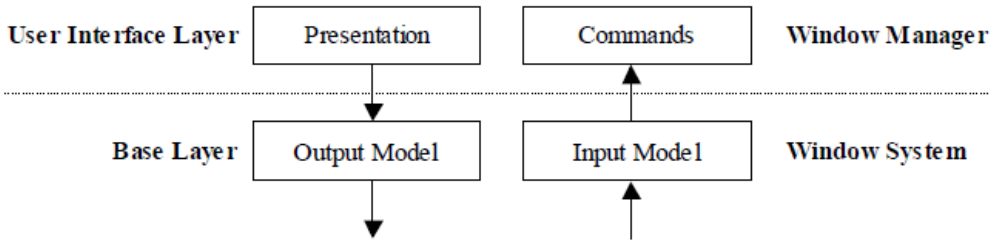
## 2.3 Systémy oken

Systém oken je software, který pomáhá uživateli sledovat a kontrolovat různé kontexty jejich fyzickým rozdělením do několika oddělených úseků na jedné nebo více obrazovkách [13]. Obecně systém oken podporuje rozdělení obrazovky na různé oblasti (obvykle obdélníkové) nazvané okna. X system dělí funkcionality oken do dvou vrstev: systém oken, což je programové rozhraní a správce oken, což je uživatelské rozhraní. Poskytuje tedy procedury, které umožňují aplikacím vykreslovat obrázky na obrazovku a získávat vstup od uživatele. Správce oken (Window Manager) pak umožňuje koncovému uživateli manipulovat s okny (přesun a změna velikosti okna) a je zodpovědný za konkrétní vzhled oken, tedy zobrazování názvů oken, překrývání, ohraničení a ikon oken. Některé operační systémy, jako například Macintosh a Microsoft Windows, se nerozdělují do uvedených dvou vrstev.

Ačkoli většina dnešních systémů poskytuje toolkity (knihovny) fungující nad systémem oken, jak bude vysvětleno dále, toolkity se obecně zabývají pouze vykreslováním komponent, jako jsou tlačítka, menu nebo posuvníky. Když chce programátor vytvořit pro aplikaci specifické prvky rozhraní a umožnit uživateli manipulaci s nimi, musí použít přímo systém oken. Programové rozhraní systému oken má tedy velký dopad na většinu uživatelských rozhraní. První systémy oken byly implementovány jako součást samotného programu nebo systému. Příkladem jsou Smalltalk a DLISP, které měly své vlastní rozhraní. Pozdější systémy implementovaly systém oken jako součást operačního systému. S cílem umožnit rozdílným systémům oken pracovat na stejném operačním systému, fungovaly některé systémy oken (např. X Window System) jako samostatný proces a využívaly meziprocesovou komunikaci operačního systému pro připojení k aplikacím.

Strukturu systému oken lze logicky rozdělit do dvou vrstev, kde každá má další dvě části, viz Obrázek 2.1. Window system, neboli základní vrstva, implementuje základní funkcionality. Dvě části této vrstvy zajišťují zobrazení grafiky v oknech (output model) a přístup k různým vstupním zařízením, typicky klávesnice a myši (input model). Primární interface základní vrstvy je procedurální a nazývá se Application programmer interface (API).

Druhá vrstva je správce oken nebo uživatelské rozhraní, to obsahuje veškerý obsah viditelný uživateli. První část této vrstvy je presentation, která se skládá z obrázků, které Správce oken zobrazuje. Druhou částí jsou commands, což jsou akce prováděné uživatelem (manipulace s okny a jejich obsahem).




Obrázek 2.1: Vrstvy systému oken [13]

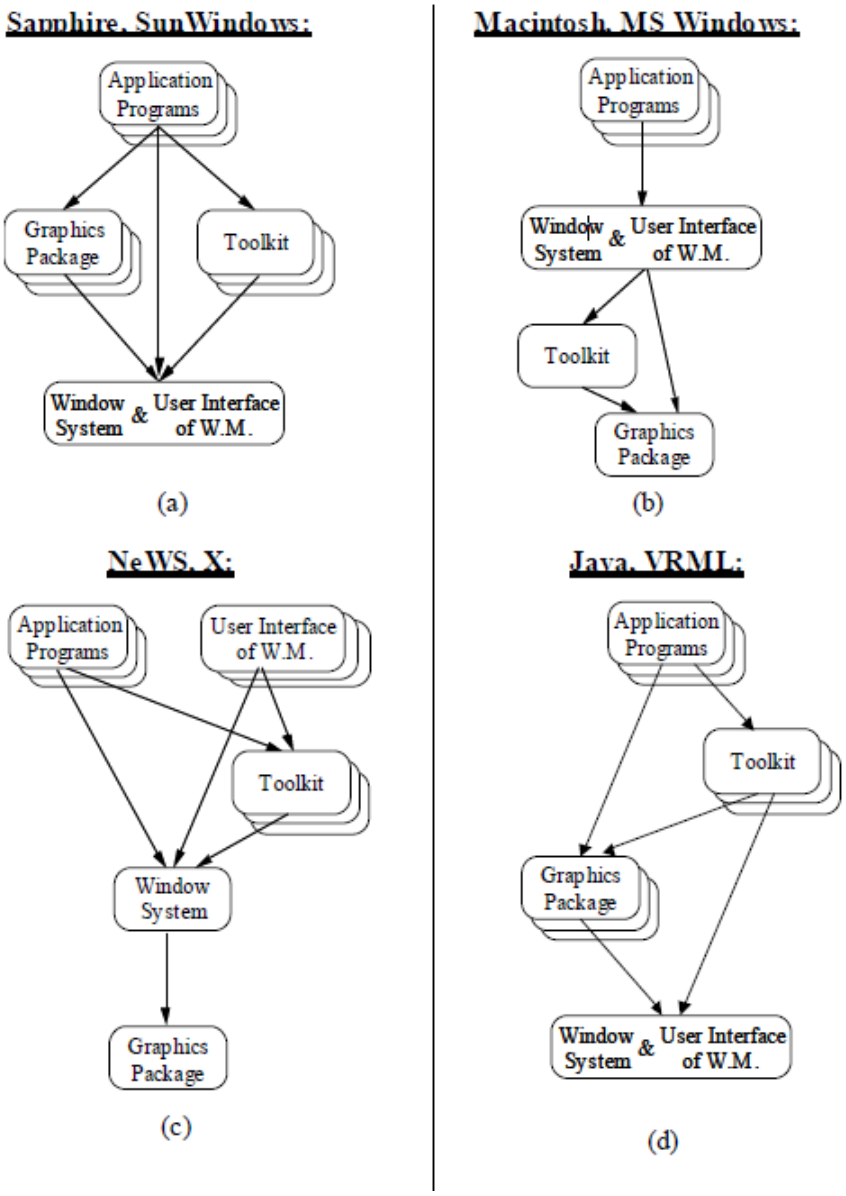
### 2.3.1 Toolkit

Nad systémem oken je toolkit, který obsahuje běžně používané grafické prvky, jako jsou menu, tlačítka, posuvníky nebo textová pole. Nad toolkity mohou být higher-level nástroje, které pomáhají při práci s prvky toolkitů, viz Obrázek 2.3. Toolkit je knihovna grafických prvků, které mohou být volány programy aplikací. Výhoda v používání toolkitu spočívá v podobně vypadajících uživatelských rozhraní, která byla vytvořena stejným toolkitem, a každá aplikace pak nemusí přepisovat standardní funkce, jako jsou menu. Nevýhodou toolkitů je omezené množství stylů interakce a zobrazení. Například může být složité vytvořit slider, který obsahuje dva indikátory pro zobrazení rozsahu a jeho minimální a maximální hodnoty, pokud taková možnost není v toolkitu připravena.

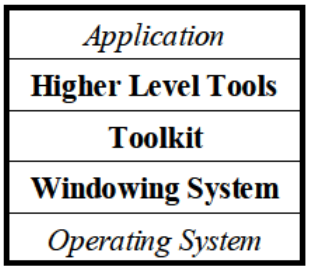
Toolkit lze implementovat buď tak, že využívá systém oken a nebo je jím využívaný. Je tedy ve vrstvě nad nebo pod systémem oken, viz Obrázek 2.2. Starší systémy poskytovaly pouze minimum prvků uživatelského rozhraní (např. jen menu a tlačítka) a očekávaly, že aplikace poskytnou ostatní, viz Obrázek 2.2-a.

V operačních systémech Macintosh a Microsoft Windows je toolkit ve spodní vrstvě a správce oken je vybudován na jeho základě. Správce oken může díky tomu použít stejné prostředky a postupy pro vytvoření svého uživatelské rozhraní, viz Obrázek 2.2-b.

Při vývoji X systému se vývojáři nemohli shodnout na jediném toolkitu, který by použili, a proto ho nechali ve vrstvě nad systémem oken. Při programování v X lze tedy použít různé sady nástrojů, příkladem jsou Gnome GTK+, Motif nebo Amulet. Systém oken ale musí být většinou schopný implementovat uživatelské rozhraní sám bez použití toolkitů[13], viz Obrázek 2.2-c. Dalším příkladem je knihovna Java Swing je implementovaná nad grafickým balíčkem Java 2D, který je nad systémem oken, viz Obrázek 2.2-d.




Obrázek 2.2: Různé způsoby implementace toolkitu [13]




Obrázek 2.3: Komponenty související se systémy oken [13]

### 2.3.2 X Window system

Systém X Window je distribuované grafické uživatelské prostředí. Unix a Linux nezačleňují uživatelská rozhraní do jádra systému a místo toho využívají pro implementaci programy uživatelské úrovně. To se týká jak textového módu, tak grafického uživatelského prostředí. Systém tedy není bezprostředně závislý na operačním systému, ve kterém byl implementován a díky distribuovanosti můžeme zobrazovat aplikace, které běží na jiném počítači. X Window ale neimplemenuje přímo uživatelské rozhraní, pouze zavádí systém oken a sadu nástrojů, pomocí kterých může být grafické uživatelské rozhraní implementováno. X systém tedy vykresluje prvky GUI na obrazovku a obsahuje metody pro zpracování interakcí mezi uživatelem a aplikacemi. Díky tomu je jednoduché implementovat pro každý program různá uživatelská rozhraní [7]. Uživatel může snadno změnit uživatelské rozhraní tím, že ukončí aktuálně používaný window manager a spustí jiný. Některé z původních správců oken fungujících pod X systémem byly například Motif window manager nebo uwm, který neobsahoval žádné ohraničení nebo popisky oken. Později se objevují kompletní desktopová prostředí, která kombinují správce oken s dalšími GUI utilitami. Mezi dvě populární desktopová prostředí patří KDE se správcem oken KWin a Gnome, který dává na výběr několik různých správců oken. Programy pokračují bez přerušení v běhu i po přepnutí na jiného správce oken. Je například možné, aby aplikace využívající widgety Motifu běžela v okně, které funguje na KWin manageru.

# 3 Historie a vývoj operačních systémů

V této kapitole se věnuji jednotlivým operačním systémům, jejich historii a popisu jednotlivých verzí. Existuje velké množství operačních systémů a jejich verzí a nebylo by možné je všechny v rozumném rozsahu rozebrat. Proto jsem vybrala ty systémy, které jsou v dnešní době nejznámější a nejpoužívanější na osobních počítačích a také systémy, které významně přispěly ve vývoji grafického uživatelského rozhraní.

## 3.1 Xerox Alto

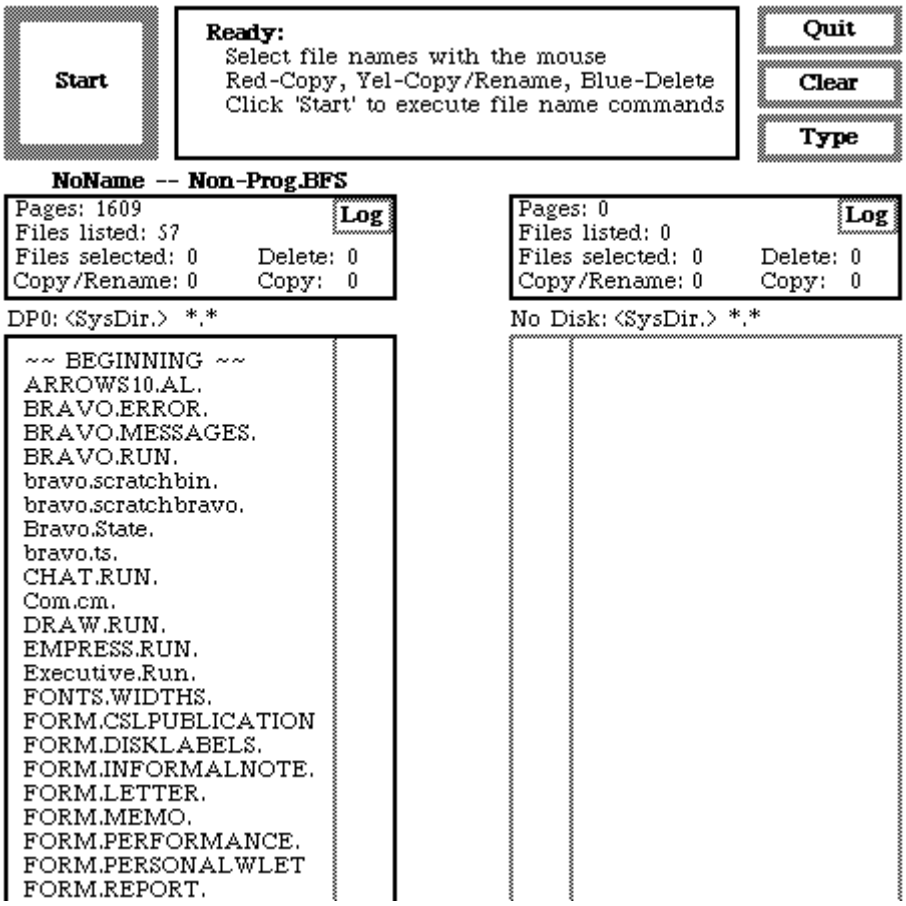
Roku 1973 se lidé v Xerox Corporation rozhodli vytvořit nový osobní počítač, který by sloužil pro výzkumné účely. Výsledkem byl Xerox Alto, první počítač využívající grafické uživatelské rozhraní.

První Xerox Alto počítače byly představeny v roce 1973, přibližně deset let před masivním rozmachem GUI založených operačních systémů. Xerox Alto představil mnoho prvků moderního GUI, tak jak je známe dnes a dá se tedy považovat za zakladatele grafických uživatelských rozhraní. Mnoho následujících operačních systémů využívajících GUI bylo odvozeno právě z tohoto systému.

První verze Alto nebyly zcela graficky založené, některé akce bylo možné provést pouze pomocí příkazů. V pozdějších verzích se už objevily nám známé grafické prvky, jako jsou okna, menu, ikony, zaškrtávací pole, přepínače (radio buttony) nebo pop-up menu. Princip pracovní plochy (Desktopu) pro zobrazení základní obrazovky byl představen také zde.

Manipulace s okny probíhala pouze pomocí kontextového menu viz 3.1, vyvolaného kliknutím pravého tlačítka na název okna. Toto menu obsaho-


Obrázek 3.1: Ukázka kontextového menu [10]




Obrázek 3.2: Souborový systém Neptune

valo nabídku se základními příkazy pro změnu velikosti, přesun, zavření a přejmenování okna.

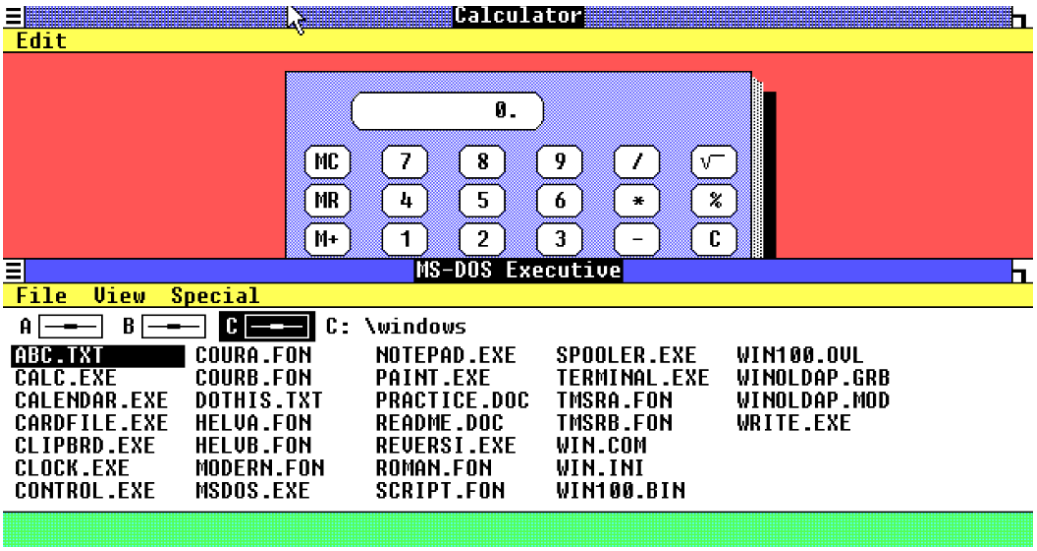
Při zobrazování grafiky a formátovaného textu připraveného pro tištěný dokument se zde kladl důraz na princip WYSIWYG, což je zkratka z anglického „What you see is what you get“, v překladu „Co vidíš, to dostaneš“. Na tomto principu je založena většina moderních textových editorů, které text a veškerý obsah stránky zobrazují tak, aby byl shodný s vytištěnou verzí dokumentu. Z toho důvodu měl Alto monochromatický displej s rozlišením 808x606 pixelů, neobvykle orientovaný na výšku, který měl připomínat list papíru[5]. Na principu WYSIWYG byl založen textový editor Bravo, který sice v prvních verzích neobsahoval veškeré moderní prvky GUI, ale i přesto nesmírně zjednodušoval uživatelům práci s textem. Pro výběr oblastí, na kterou budou aplikovány příkazy pro úpravu textu, se použila myš, což nám dnes přijde jako samozřejmost, ale v té době to byla revoluční věc usnadňující práci.

Na obrázku 3.2 vidíme souborový systém Neptune Directory Editor, který byl zcela ovladatelný pomocí myši. Obsahoval tlačítka, seznam souborů zobrazený ve dvou sloupcích, ale žádné ikony nebo obrázky.

Xerox počítače nebyly komerčním produktem, využívaly se především ve výzkumném centru PARC společnosti Xerox a na univerzitách. Nikdy se masivně nerozšířily mezi veřejnost jako následující systémy od Applu nebo Microsoftu.

## 3.2 Microsoft Windows

### 3.2.1 Windows 1

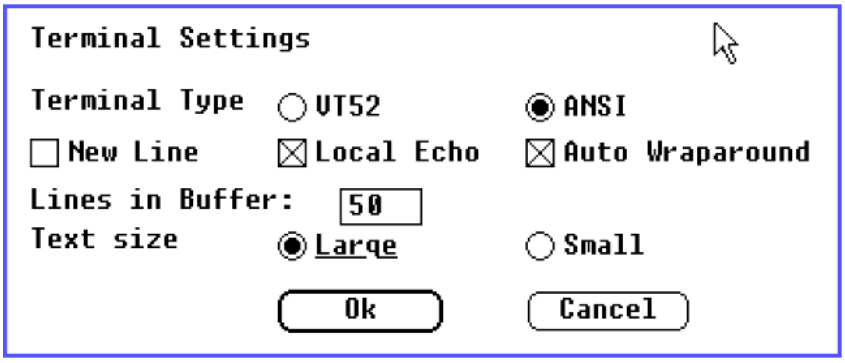



Obrázek 3.3: Zároveň otevřený MS-DOS Executive a Kalkulačka

Windows 1, vydaný v roce 1985, byl vůbec první systém od společnosti Microsoft obsahující grafické uživatelské rozhraní. Windows 1 nebyl sám o sobě operační systém, ale pouze grafická nadstavba pro MS-DOS [15].

Po spuštění Windows 1 se otevře MS-DOS Executive, což je textově řešený file manager. Na rozdíl od tehdejších Mac systémů má každé okno svůj vlastní interface, u většiny aplikací se v horní části nachází menu bar. Pod menu barem jsou ikony zobrazující připojené disky, mezi kterými se dá přepínat. Jsou to jediné ikony, které zde najdeme.

V levém horním rohu každého okna je system box, po jeho nakliknutí se objeví pull down menu s možnostmi pro dané okno. V pravém horním rohu okna je tlačítko sloužící pro změnu velikosti okna. Při dvojkliku se okno

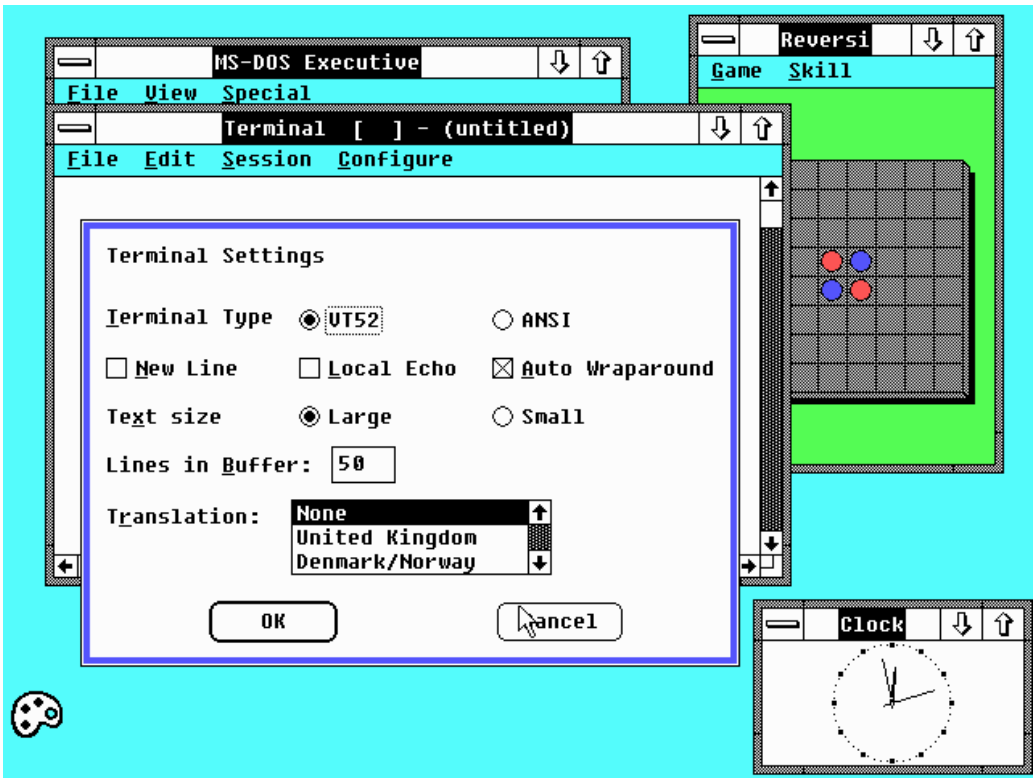



Obrázek 3.4: Dialogové okno s nastavením

zvětší přes celou obrazovku a při táhnutí ikonky se mění velikost okna. Některé programy mají tuto ikonku i v pravém dolním rohu. V nejspodnější části obrazovky je pruh vyhrazený pro ikonky minimalizovaných aplikací, které můžeme dvojklikem maximalizovat. Pokud je některé z oken maximalizované, tento pruh není vidět.

Jednotlivá okna se ve Windows 1 nemohla překrývat, pouze skládat vedle nebo pod sebe do mřížky, viz Obrázek 3.3. Jedinou výjimkou byla dialogová okna, která mohla překrývat ostatní, ale nedala se minimalizovat. Na obrázku 3.4 vidíme typické dialogové okno s nastavením. Jsou zde použity komponenty jako zaškrtávací boxy, textové pole a přepínače. Tyto prvky poskytuje Windows, takže se aplikace nemusí zabývat jejich ovládáním a vykreslováním [10].

### 3.2.2 Windows 2

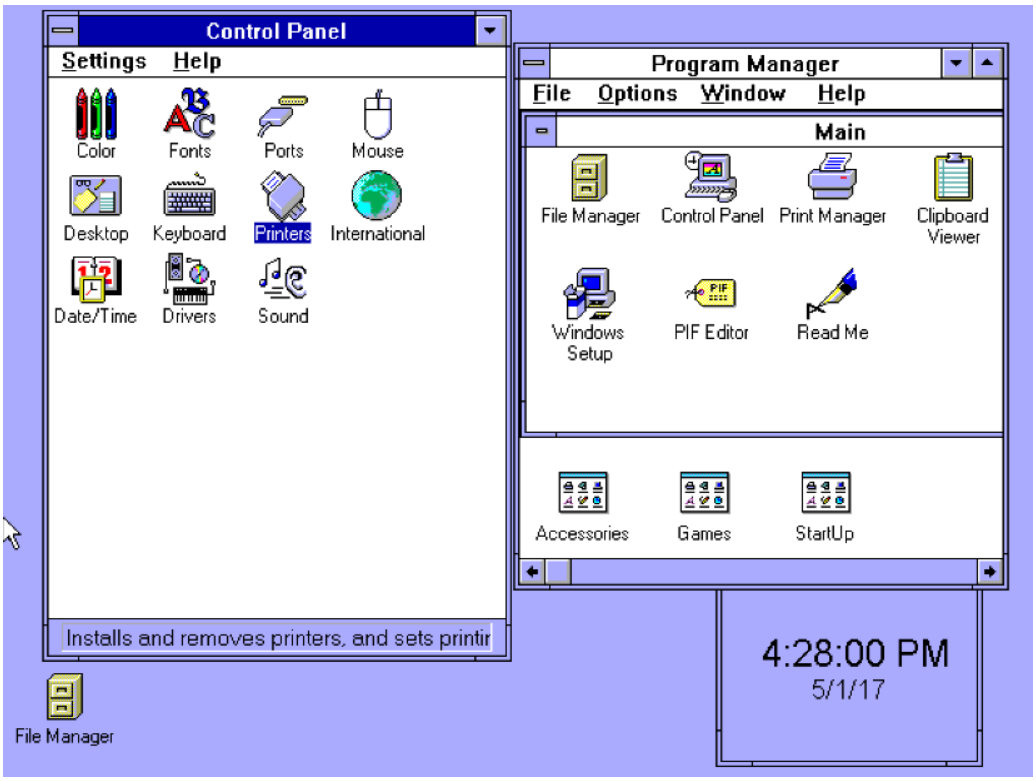



Obrázek 3.5: Ukázka prostředí Windows 2 [10]

Windows 2, stejně jako Windows 1, začíná po startu v MS-DOS Executive a je vybaven stejnými aplikacemi. Velkou inovací je vylepšená manipulace s okny. Oproti Windows 1 totiž umožňuje překrývání jednotlivých oken, libovolnou změnu velikosti a umisťování na libovolné místo na obrazovce, viz Obrázek 3.5. Stejně tak ikony minimalizovaných aplikací můžeme umisťovat na libovolné místo na ploše. Novinkou je také ovládání některých funkcí pomocí klávesových zkratek, Alt + podtržené písmeno zobrazené u slov v menu baru.

Tlačítka pro manipulaci s okny byla zcela předělána, v pravém horním rohu jsou nyní dvě tlačítka ve tvaru šipky, která slouží pro minimalizaci a maximalizaci okna. V levém horním rohu okna je pak tlačítko, které po dvojkliku zavře okno a po jednoduchém kliknutí zobrazí drop down menu s možností minimalizace, maximalizace a zavření okna. U všech těchto možností jsou vypsané klávesové zkratky. V pravém dolním rohu, kde se scházejí posuvníky, se nachází tlačítko pro změnu velikosti okna, ale se stejným výsledkem můžeme použít i rohy každého okna.

### 3.2.3 Windows 3

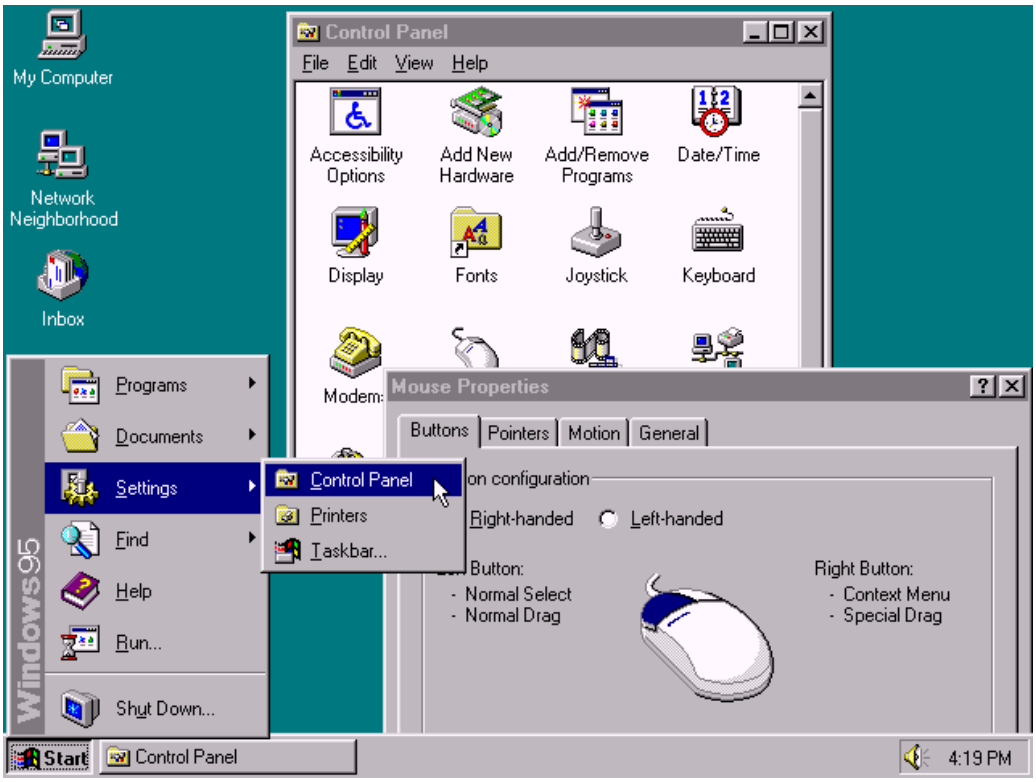



Obrázek 3.6: Ukázka prostředí Windows 3

Windows 3 byl vydán roku 1990. Manipulace s okny zůstává velice podobná jako v předchozí verzi Windows, ale celkový vzhled oken a tlačítek je vylepšený. U většiny ovládacích tlačítek se objevuje 3D vzhled, bohužel další ovládací prvky 3D vzhled nedostaly. Další změny se dočkalo okno, které je zrovna nakliknuté, nyní má zabarvený horní pruh (viz Obrázek 3.6).

MS-DOS Executive, použitý ve dvou předchozích verzích Windows je ve verzi 3 nahrazen Program Managerem, ve kterém můžeme spouštět aplikace přes jejich ikony. File Manager je pouze textový, nejsou zde žádné ikony. Díky drag-and-drop můžeme soubory snadno přehazovat mezi různými adresáři a disky. Ikony nyní fungují jako drag-and-drop a lze je libovolně umisťovat v daném okně. V okně Control panel je v dolní části okna lišta s informací o nakliknuté ikoně, např. Printers – Instaluje a maže tiskárny a mění nastavení pro tisk. Po dvojkliku na plochu se spustí Task List program, zobrazující všechny otevřená okna, který se hodí pokud máme spuštěno mnoho oken najednou.

### 3.2.4 Windows 95




Obrázek 3.7: Ukázka prostředí Windows 95 [10]

Windows 95, vydaný v srpnu roku 1995, byl velice úspěšným následníkem Windows 3. Hlavním cílem při vývoji grafického uživatelského rozhraní bylo udělat systém více intuitivně a jednoduše ovladatelný pro začátečníky i pokročilé uživatele počítačů[18].

Grafické uživatelské rozhraní se dočkalo velkých změn a je velice podobné aktuálním Windows systémům. Na první pohled velkou změnou oproti předešlým verzím Windows je zcela jistě Task bar, lišta ve spodní části obrazovky (viz Obrázek 3.7). Lišta je rozdělena do třech částí, kde každá část obsahuje různé funkcionality.

1. Tlačítko Start, které otevírá Start menu v kaskádovitém stylu, tedy položky menu se rozbalují do dalších sub-menu. Menu obsahují veškeré programy a dokumenty. Představuje snadný způsob přístupu k programům a souborům.

2. Hlavní panel umístěný uprostřed lišty se záložkami všech běžících a minimalizovaných programů.

3. Na pravé straně lišty je takzvaná oznamovací oblast, nebo také systémová část lišty. Jsou zde zobrazené hodiny, ovládání hlasitosti, systémové aplikace a aplikace běžící na pozadí operačního systému.

Program Manager byl kompletně nahrazen pomocí Start menu a File Manager pomocí Windows Explorer. Rozvržení a design okna se také změnil, v pravém horním rohu každého okna je nyní tlačítko ve tvaru křížku pro zavření okna a dvě tlačítka pro minimalizaci a maximalizaci okna. Další ze změn oproti předchozí verzi Windows 3 je 3D vzhled všech ovládacích prvků, tedy menu, ovládací tlačítka oken a veškerých dalších tlačítek.

Windows 95 je také první verze Windows systémů, která obsahuje pracovní plochu tak, jak ji známe dnes a jak byla představena v Xerox systémech. Plocha obsahuje ikony programů, složek nebo souborů. Ve Windows 3 byly na ploše zobrazeny pouze aktuálně spuštěné programy, které jsou ve verzi 95 na spodní liště.

Další novinkou je Aktivní plocha, funkcionalita propojená s prohlížečem Internet Explorer, která umožňuje pomocí HTML zobrazovat obsah internetu přímo na ploše jako pozadí nebo jednotlivé nezávislé prvky na ploše. Součástí systému byla až do verze Windows Vista. Aktivní plocha je podobná desktopovým widgetům, které umožňují upravovat informace zobrazované na ploše bez nutnosti navštívení webové stránky.

### 3.2.5 Windows 98


Obrázek 3.8: Ukázka prostředí Windows 98 [10]

Windows 98, vydané roku 1998, jsou vzhledově velmi podobné Windows 95, viz Obrázek 3.8. Změny jsou drobné, například lišta se Start tlačítkem ve spodní části obrazovky je nyní rozdělena do čtyř částí. Od předchozí verze zde přibyl úsek lišty s panelem snadného či rychlého spuštění. Na tento panel si můžeme přidat ikony programů, které často používáme a chceme je mít rychle k dispozici. Pokud přejedeme myší přes položku v menu baru, objeví se okolo ní 3D rámeček.

Hlavní změnou je Windows shell. Základní shell je zde Windows Explorer, který je integrovaný s webovým prohlížečem Internet Explorerem. Tímto je ovlivněn celkový vzhled uživatelského rozhraní. Tento styl zobrazení ale nemusí všem vyhovovat, proto bylo možné vybrat si ze tří různých typů zobrazení.

1. Webový styl - již zmíněný styl s integrovaným webovým prohlížečem, prohlížení plochy a adresářů funguje stejně jako prohlížení webové stránky, pomocí jednoho kliknutí. Veškeré soubory, složky a aplikace se otevírají ve stejném okně.

2. Klasický styl - je velmi podobný Windows 95, používá se dvojklik pro otevírání položek a pro každý nový soubor, složku nebo aplikaci se otevře vlastní okno.

3. Vlastní styl - zde je možné namíchat nastavení z obou předchozích možností dohromady. Nastavit lze způsob procházení složek, vzhled oken a vybírání a otevírání položek. Například si vyberu webový styl zobrazení oken a otevírání položek dvojklikem.

Rozložení okna u webového stylu rozhraní je rozděleno do několika částí:

1. Standardní tlačítka - např. krok zpět, kopírování, smazání.

2. Bar s adresou, ve které se právě nacházíme, např. Tento počítač.

3. Lišta se záložkami odkazů.

### 3.2.6 Windows XP


Obrázek 3.9: Ukázka prostředí Windows XP [10]

Windows XP (zkratka z anglického eXPerience) je operační systém vycházející z řady Windows NT, který se dostal na trh roku 2001. Předchůdcem

tohoto systému byl Windows 2000, z jehož názvu vyplývá, že předchůdcem byl Windows 98, ale není tomu tak. Windows 2000 vychází z řady systémů Windows NT. Následníkem Windows 95 a 98 je Windows Millenium Edition (zkráceně Windows ME), který stále běžel na základě MS-DOS stejně jako všechny předchozí verze.

Windows XP přišel se zcela přepracovaným grafickým prostředím (viz Obrázek 3.9), změnou menu, ovládacích panelů a řadou dalších funkcí. Nově upravené rozhraní má dvě základní formy:

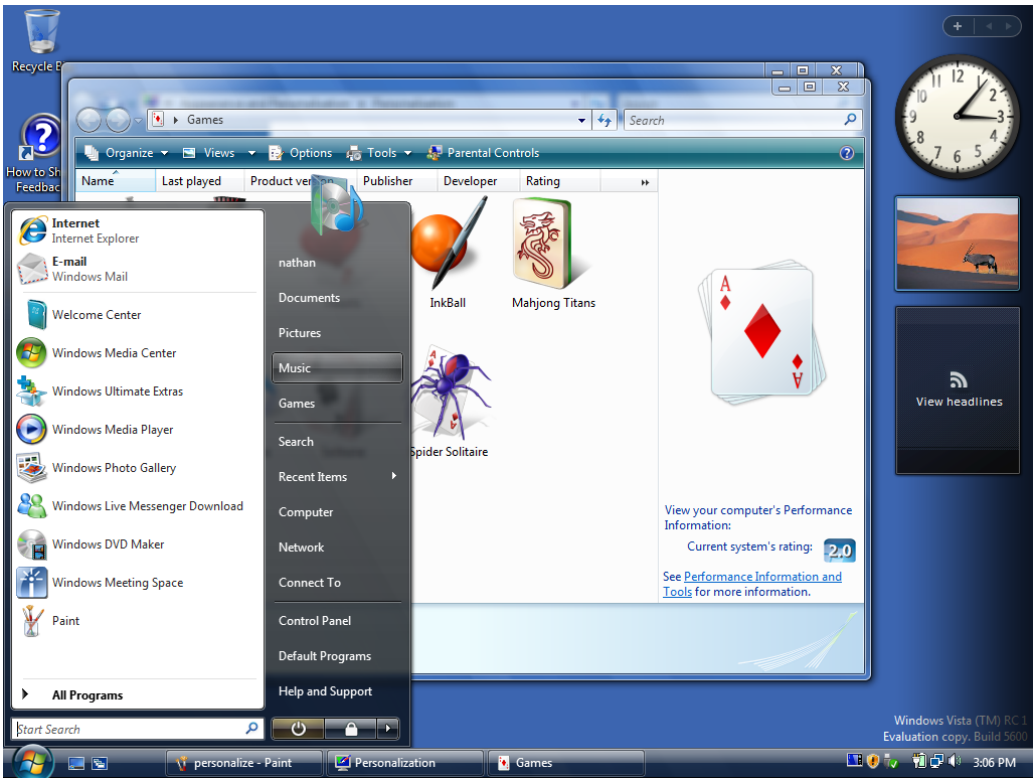
1. nová nabídka Start menu, původně označovaná jako Start panel, již není implementována jako kaskádové menu s dalšími sub-menu, vše je zobrazeno v jednom panelu.

2. nová uvítací obrazovka, která nahrazuje staré dialogové okno pro přihlašování uživatelů.

Nově je možné nastavit si jednotlivé části rozhraní podle vlastního uvážení. Chceme-li využívat starý vzhled Start menu a zároveň nový vzhled Exploreru nebo uvítací obrazovky, je to možné. Novinkou jsou také jinak tvarovaná okna aplikací (např. Windows Media Player). Doposud byla okna vždy obdélníkového tvaru, nyní mají ale aplikace možnost předělat vlastní uživatelské rozhraní a zcela upravit vzhled a tvar okna. Dalším novým prvkem rozhraní jsou rozbalovací dialogová okénka umístěná obvykle v levé části okna Průzkumníku a slouží k zobrazení souvisejících nastavení.

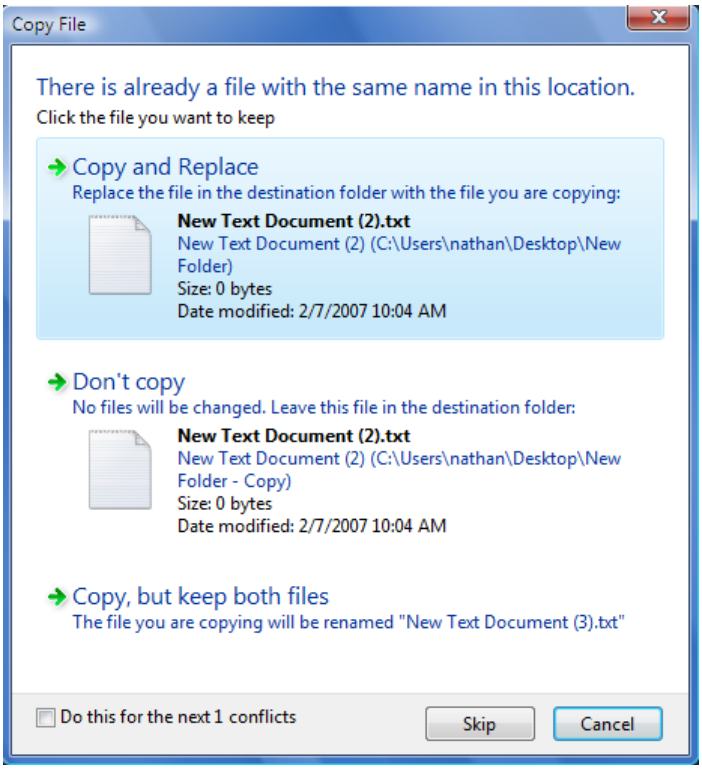
Hlavní panel je oproti starším verzím méně přeplněný a chaotický, převážně díky dvěma hlavním změnám, které ocení uživatelé využívající zároveň více aplikací. Ikony aplikací v systémové části lišty, které jsou neaktivní, nejsou zobrazeny. Druhá změna se týká více dokumentů, které jsou všechny otevřené jedinou aplikací. Každý takový dokument měl v hlavním panelu vlastní záložku, což velmi rychle zaplnilo panel spoustou nepřehledných karet. Nyní jsou všechny dokumenty otevřené ve stejné aplikaci sdruženy pod jedinou záložkou dané aplikace, která slouží jako drop-down seznam oken, která můžeme otevřít.

### 3.2.7 Windows Vista




Obrázek 3.10: Ukázka prostředí Windows Vista [10]

Windows Vista, vydaný roku 2007, je další operační systém z řady Windows NT. Systém přišel se znovu zcela přepracovaným grafickým prostředím nazvaným Aero (viz Obrázek 3.10), které podporuje průhlednost oken a nabídek, plynulý pohyb oken, trojrozměrné animace a ikony přizpůsobené i vyšším rozlišením. Rozhraní je orientováno především na celkový efekt průhlednosti, obsahuje různé odlesky, hlavně u tlačítek pro minimalizaci a zavření okna, vysvícené ikony a lesknoucí se progress bary. Aero má specifické požadavky na hardware, aby bylo možné využít všech nových efektů, které toto rozhraní poskytuje, je nutné mít v počítači grafickou kartu s WDDM ovladačem. Nové rozhraní přineslo funkci živých náhledů, které se zobrazují u aplikací na hlavním panelu jako malé okno s nápovědou. Živý náhled zobrazí ve zmenšeném okně obsah daného okna, například pohybující se video nebo spuštěnou počítačovou hru. Novinkou je také funkcionalita Flip 3D, neboli efektní přepínání oken, které zobrazuje 3D překrývající se náhledy oken spuštěných aplikací při použití Alt+Tab. Další velkou novinkou je nastavitelný Sidebar, průhledný bar v pravé části obrazovky, který obsahuje námi volitelné widgety, mezi které můžou patřit třeba analogové




Obrázek 3.11: Ukázka prostředí Windows Vista [10]

hodiny, RSS novinky, koš nebo kalendář. Změnou prošlo také Start menu, které nyní obsahuje nové pole pro vyhledávání a tlačítka pro rychlé uspání a vypnutí počítače. Celkový styl zobrazování menu je jiný, než u předchozích verzí, menu se již nerozbaluje kaskádovitě na další submenu, ale obsah jednotlivých složek se zobrazuje přímo do základního Start menu.

Vylepšen byl i Průzkumník, ve kterém je nyní možné zobrazovat náhledy jednotlivých souborů, například obrázků. Při pohledu do adresáře se zobrazují náhledy multimediálních souborů spolu s popisem typu souboru. Při pohledu na okna Průzkumníku nikde nenajdeme menu bar, který obsahuje položky pro práci se souborem, změnu zobrazení položek nebo nápovědu. Tento menu bar je defaultně skrytý, ale v nastavení ho lze opět nechat zobrazit. Oproti Windows XP je také na výběr více barev pro rámce oken. Je možné namíchat si vlastní barvu pomocí color mixeru, který byl přítomný ve Windows 1.

Obrázek 3.11 zobrazuje novou funkci uživatelského rozhraní, která se objevila na několika místech. Dialogová okna, která mají na výběr více možností se nyní zobrazují jako seznam položek, kde každá položka seznamu je, až to tak na první pohled nevypadá, tlačítko. Tlačítka jsou většinou hustě popsána a kvůli tomu jsou v některých případech dialogy špatně čitelné.

Windows Vista nabízí několik následujících volitelných stylů rozhraní:

1. Windows Aero, které je popsáno výše v této kapitole.

2. Windows Vista Standard - stejně jako Aero využívá DWM (Desktop Window Manager)technologii, ale některé efekty jako Flip 3D, průhlednost a některé animace oken nejsou funkční, ale hardwarové požadavky jsou podobné jako u plnohodnotného Aero rozhraní. Tento styl rozhraní je určen pro verzi Home Basic, kde nahrazuje Aero.

3. Windows Vista Basic - Tato verze rozhraní nevyužívá DWM a je obdobou stylu Windows XP s několika efekty navíc. Nevyžaduje speciální grafickou kartu a hardwarové nároky jsou přibližně stejné jako u Windows XP. Tento styl je určen pro verzi Starter a také je zvolen jako výchozí pro počítače s nevyhovující grafickou kartou.

4. Windows Classic - Klasický vzhled připomínající styl rozhraní Windows 95, 98 nebo 2000. Nevyužívá DWM a nevyžaduje WDDM ovladače.

### 3.2.8 Windows 7

Windows 7 byl vydán roku 2009 a využívá stejně jako Windows Vista grafické uživatelské rozhraní Aero. Vista byla kritizována za nekompatibilitu s některým softwarem a hardwarem a vysokými hardwarovými nároky. Ve Windows 7 mají prvky Aera oproti předchozí verzi více hladký průběh, především na méně výkonném hardwaru. Toho bylo dosaženo díky dvěma vylepšením grafického subsystému. Windows 7 nyní umožňuje více procesům přistupovat k GDI (graphic device interface) najednou, takže když klikneme myší nebo píšeme na klávesnici, můžeme očekávat okamžitou reakci. Druhým vylepšením je převedení části práce na GPU, což odlehčuje práci CPU a paměti. Výsledkem toho je plynulé fungování efektů Aera i na hardwarově slabších počítačích.

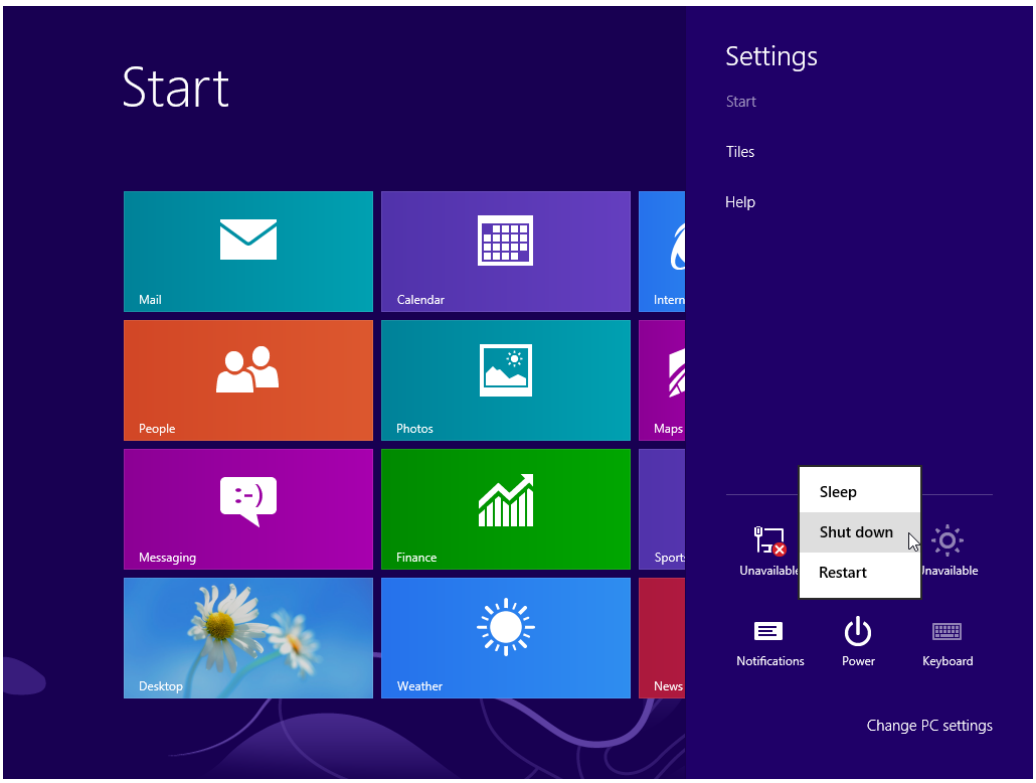
Nápadnou změnou prošel hlavní panel ve spodní části obrazovky. Styl panelu se záložkami otevřených a minimalizovaných aplikací, který se vyskytoval ve všech verzích od Windows 95, je nyní nahrazen velkými ikonami. Oblast s ikonami rychlého spuštění je odstraněna a místo ní je možno ikony spustitelných souborů nebo adresářů přichytit k hlavními panelu, pokud je pak aplikace spuštěna, její ikona je vysvícená. U ikon na hlavním panelu je možné vyvolat speciální kontextové menu, které se liší u různých programů. Menu obsahuje seznam nedávno otevřených dokumentů, možnost zavření okna a úlohy lišící se podle vybraného programu (např. Windows Media

Player nabízí možnost přehrání veškeré hudby). Do tohoto seznamu si také můžeme připnout dokumenty, které chceme mít rychle k dispozici.

Živé náhledy, představené v předchozí verzi systému, byly vylepšeny v podobnou funkci nazvanou Aero peek. Pokud máme otevřeno více oken stejné aplikace, zobrazí se nám všechny náhledy najednou. Po přejetí myší po náhledu se zobrazí všechna okna, kde vybrané okno je v popředí a ostatní jsou průhledná, což dělá vybírání či zavírání jednotlivých oken jednodušší.

Sidebar představený ve Windows Vista v této verzi systému nenajdeme. Místo toho lze umisťovat gadgety přímo na libovolné místo na ploše. K rychlému přístupu na plochu a minimalizaci všech oken slouží průhledné tlačítko na hlavním panelu v pravém dolním rohu obrazovky. Objevily se také nové varianty manipulace s okny. Pokud přetáhneme okno k bočním krajům obrazovky, automaticky se přichytí k danému okraji a vyplní polovinu obrazovky. Po dvojkliku na horní okraj okna nebo přetažení okna k horní části obrazovky se okno maximalizuje.

### 3.2.9 Windows 8




Obrázek 3.12: Rozhraní Metro [10]

Systém Windows 8, vydaný roku 2012, přinesl nejzásadnější změny v uživatelském rozhraní od příchodu Windows 95. Systém je založený na uživatelském rozhraní s názvem Modern User Interface (původně Metro), které Microsoft již dříve použil v systému Windows Phone. Rozhraní je navrženo tak, aby bylo dobře použitelné na dotykových displejích, ale je upraveno i pro klasickou práci s myší a klávesnicí na počítači. Metro se vyznačuje především dlaždicemi, velkými barevnými boxy s obrázky a textem, které spouštějí jednotlivé aplikace a slouží jako zástupci aplikací a zároveň jako jejich aktivní widgety, viz Obrázek 3.12. Aplikace vytvořené speciálně pro rozhraní Metro, které nejsou kompatibilní s předchozími verzemi systému, běží obvykle přes celou obrazovku, takové chování známe typicky z aplikací běžících na tabletech nebo chytrých telefonech. Některé vyskakovací zprávy a dialogy převzaly také vzhled Metra a neobsahují žádné ovládací prvky okna a mohou překrýt celou obrazovku (např. okno při spouštění neznámého typu souboru).

U klasického desktopového rozhraní došlo k vizuálním změnám ladících právě s vzhledem rozhraní Metro, které spočívají především v odstranění zaoblených rohů, omezené nastavitelnosti průhlednosti, odrazů a barevných přechodů. Vzhled celých Windows 8 je tak velice jednoduchý, přehledný se sytějšími a kontrastnějšími barvami. Hlavní panel na ploše neobsahuje tlačítko ani menu Start, jinak zůstal stejný jako v předchozí verzi systému. Po kliknutí do levého spodního rohu obrazovky, kde bylo vždy umístěné tlačítko Start, se přepneme zpět do hlavní nabídky Metra. Kromě hlavního panelu máme nyní dva nové postranní panely, ke kterým se dostaneme například při najetí myši k okrajům obrazovky. Pravý panel obsahuje tlačítka pro přepnutí do rozhraní Metra, nastavení, vyhledávání a další. Levý panel nahrazuje funkci Flip 3D z Windows Vista, zobrazuje běžící programy a umožňuje nám se mezi nimi rychle přepínat.

Ribbon, neboli pás karet, je ovládací prvek uživatelského rozhraní, který už se objevil v novějších verzích některých programů ve Windows 7, například v kancelářském balíku Microsoft Office a v Malování. V operačním systému Windows 8 se tento prvek vyskytuje v dalších aplikacích a je použit i v Průzkumníkovi Windows. Ribbon je horizontální široký pás nacházející se typicky v horní části okna a má podobu řady přepínacích záložek(karet), kde každá karta sdružuje související funkce(například funkce zobrazení, formátování nebo sdílení), které by mohl uživatel pravděpodobně potřebovat.

#### Windows 8.1

Windows 8.1 je první aktualizace systému Windows 8, vydaná roku 2013, a poskytuje lepší optimalizaci rozhraní Metro především pro uživatele, kteří nevyužívají dotyková zařízení, ale pracují klasicky s myší a klávesnicí, kde používání gest typických pro dotykové ovládání není na počítači ideální. Aby byla práce ve Windows 8 více efektivní, nastalo v systému několik optimalizací. Uživatel si může nastavit, zda chce po spuštění počítače skončit na pracovní ploše a nebo v nové hlavní Metro nabídce. Pokud není systém využíván na dotykovém zařízení, je rozhraní vybaveno několika funkcemi navíc (např. tlačítko pro rychlé vypnutí počítače v Metro nabídce). Další důležitá aktualizace, která přinesla nové funkce, byla vydána roku 2014 s názvem Windows 8.1 Update. Změnou prošel například hlavní panel v klasickém desktopovém rozhraní, který opět obsahuje známé tlačítko Start. Panel nyní také nabízí možnost připnout a zobrazit moderní aplikace jako ikony, stejně jako je tomu u ostatních běžných programů. Všechny aplikace, které nemáme připnuté na hlavním panelu, musíme stále spouštět přes Metro menu. Mezi funkce hlavního panelu se také vrátily živé náhledy, které známe z Windows 7. Lištu hlavního panelu lze nyní zobrazit i v hlavní nabídce Metro, díky čemuž můžeme rychle a efektivně přepínat mezi plochou a Metro nabídkou.

### 3.2.10 Windows 10

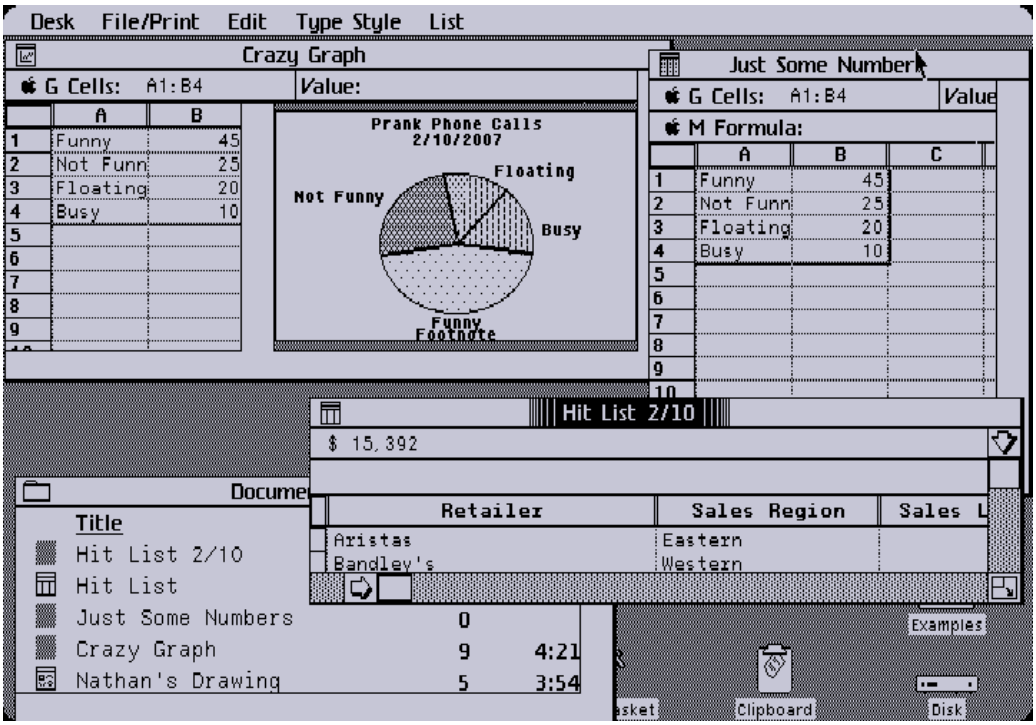
Windows 10 byl vydán roku 2015 a je to zatím poslední vydaný operační systém od společnosti Microsoft. Systém lze použít na stolních počítačích, noteboocích, tabletech, mobilech a dalších zařízeních. Grafické uživatelské prostředí spojuje prvky Modern User Interface (Metro) a Windows Aero a je jednotné pro všechna zařízení. Rozhraní se mění v závislosti na používaném vstupním zařízení, na výběr jsou dva typy rozhraní. První je optimalizované pro používání myši a klávesnice, druhé pro dotykové obrazovky. Na první pohled připomíná systém více Windows 7 než svého předchůdce Windows 8, hlavně kvůli návratu klasického hlavního panelu a Start menu. Celkový vzhled oken je ale více podobný těm z Windows 8, má ostré rohy, jednoduché a kontrastní barvy a panely v horní části oken jsou užší.

Klíčové funkce jsou opět přístupné z hlavního panelu, který nově obsahuje dvě nová tlačítka. První tlačítko v levé části panelu slouží pro zobrazení virtuálních ploch, běžících úloh a přepínání se mezi nimi. Druhé v pravé části panelu, otevře vertikální bar v pravé části obrazovky obsahující oznámení systému a několik tlačítek, např. nastavení, přechod do režimu tabletu nebo informace o připojení k síti. V levé části hlavního panelu je známé tlačítko otevírající vylepšené Start menu, které je spojením klasického menu napří-

klad z Windows 7 a dlaždic z Windows 8. Vzhled menu si můžeme podle vlastního uvážení upravit, lze měnit velikost menu, počet dlaždic i zobrazované aplikace. Na výběr je také možnost využití celoobrazovkového zobrazení menu. Ve starších verzích systému bylo možné přichytit okna vertikálně ke stranám obrazovky, nyní máme možnost přichycení k rohům obrazovky, velikostí tedy do čtvrtiny obrazovky.

## 3.3 Apple

### 3.3.1 Apple Lisa




Obrázek 3.13: Ukázka prostředí Apple Lisa [10]

První počítač s grafickým uživatelským rozhraním od společnosti Apple byl Apple Lisa Office System, představen roku 1983. Neměl ale příliš velký úspěch, hlavně kvůli vysoké ceně a malému výkonu.

GUI bylo silně ovlivněno grafickým rozhraním počítačů Xerox Alto a Xerox Star (viz Obrázek 3.13), ale objevila se zde spousta inovací a nových prvků. Jedním z nich je pull-down menu bar, který se vždy vyskytoval v horní části obrazovky. Další inovací byly checkmarky vedle vybrané položky v menu a koncept klávesových zkratek pro nejpoužívanější příkazy v menu.

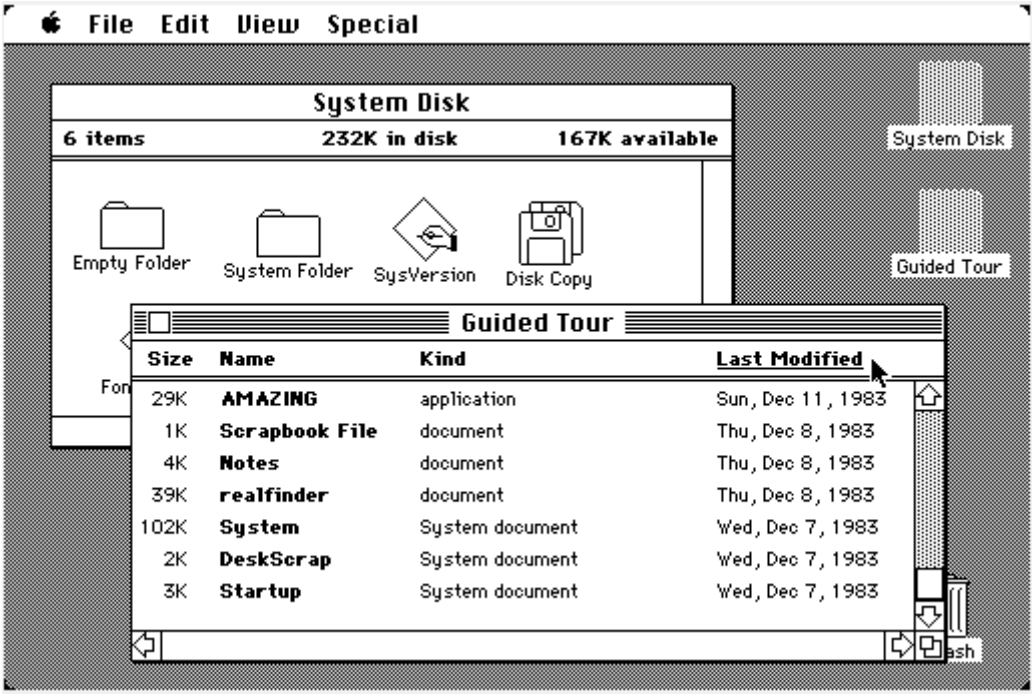
Myš prošla změnou oproti původní třítlačítkové myši používané u Alta, pro větší jednoduchost měla jen jedno tlačítko. Protože rozhraní vyžadovalo alespoň dvě akce pro každou ikonu, výběr a spuštění, byl pro tento účel vynalezen koncept dvojkliku. Dvojklik se později stal ve všech GUI standartním způsobem pro spouštění programů.

V Xerox systémech vždy reprezentovala ikona pouze jeden soubor. Rozhraní Lisy ale představilo nový pohled na ikony, ikony mohou reprezentovat všechny soubory v souborovém systému. Ten se dal pak prohlížet pomocí složek s hierarchickou strukturou, kde se každý adresář otevřel v novém okně. Ve stejné době byla vynalezena drag-and-drop funkcionalita, ze které se intuitivně vyvinul způsob manipulace se soubory. Příkladem je označení skupiny souborů a následné přetažení do jiné složky, čímž se zkopírovaly[17].

Rok po vydání Lisy představil Apple nyní slavný Macintosh, který v každém směru Lisu překonal.

### 3.3.2 Mac System 1

Mac System 1 vydaný roku 1984 si zachoval většinu GUI prvků od systému Lisa, ale na rozdíl od něj se masivně rozšířil mezi veřejnost.




Obrázek 3.14: Ukázka prostředí Mac Systemu 1 [10]

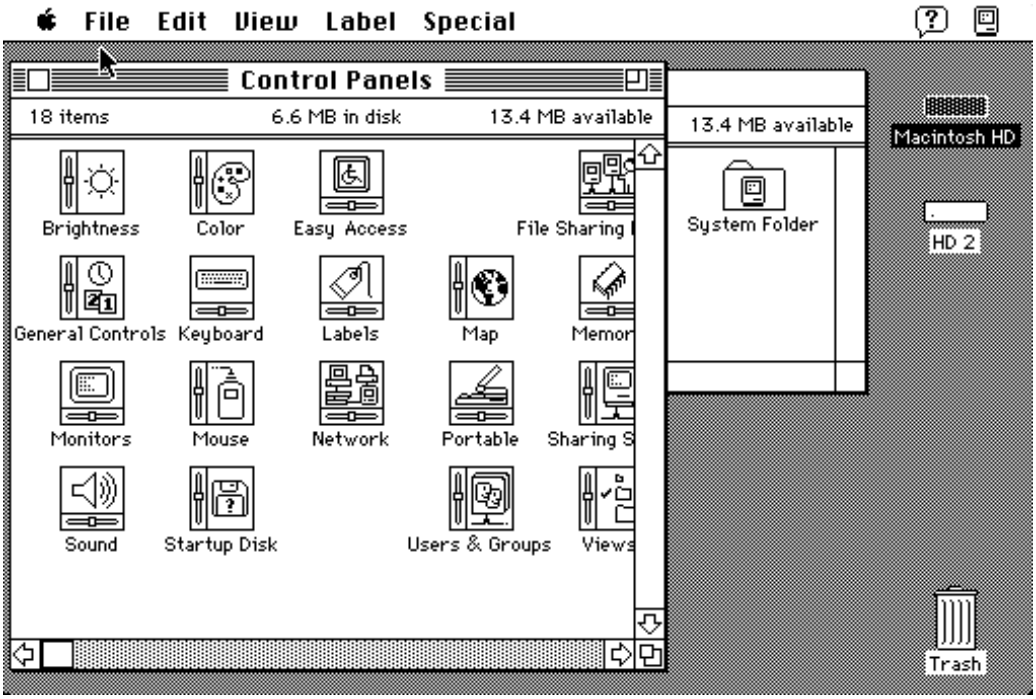
Na Obrázku 3.14 vidíme prostředí systému. V horní části obrazovky je stejně

jako u Lisa systému umístěn menu bar, který sdílí všechny spuštěné aplikace, ale ovládáme jím vždy právě aktivní okno. Po kliknutí na položku menu baru se objeví drop-down menu s nabídkou příkazů. Na rozdíl od první verze Microsoft Windows se mohou jednotlivá okna překrývat a je možné je umisťovat kdekoliv na obrazovce. V levém horním rohu okna je tlačítko (čtvereček), které zavře aktuální okno. Každé okno má v pravé a dolní části posuvníky, které se v pravém dolním rohu setkávají, na tomto místě je tlačítko pro změnu velikosti okna. Tlačítko pro zavření okna, šipky v posuvnících a tlačítko pro změnu velikosti okna jsou viditelné pouze když máme dané okno vybrané.

Veškeré aplikace a soubory jsou reprezentovány pomocí ikon, přes které je spouštíme. Můžeme je umisťovat na libovolné místo na ploše a pomocí drag-and-drop přemisťovat a kopírovat. Defaultně se na ploše nacházely ikony koše a vložených disků, pro rychlý a snadný přístup k souborům. Nové okno se vždy otevřelo se zoomovacím efektem. Adresáře mohou být zobrazeny buď jako ikony, nebo seznam. S položkami seznamu v textové podobě ale není možné manipulovat pomocí drag-and-drop.

Zatímco desktopové aplikace, například kalkulačka, běžely ve svých jednotlivých oknech, ostatní aplikace využívaly celou obrazovku. Sdílený menu bar zmizel a místo něj měly tyto aplikace své speciální menu, opět v horní části obrazovky. Tento způsob zobrazení byl kvůli šetření prostoru na obrazovce výhodný pro malé displeje. Později vydané systémy už podporovaly sdílení místa na obrazovce mezi více aplikacemi.

### 3.3.3 Mac System 7



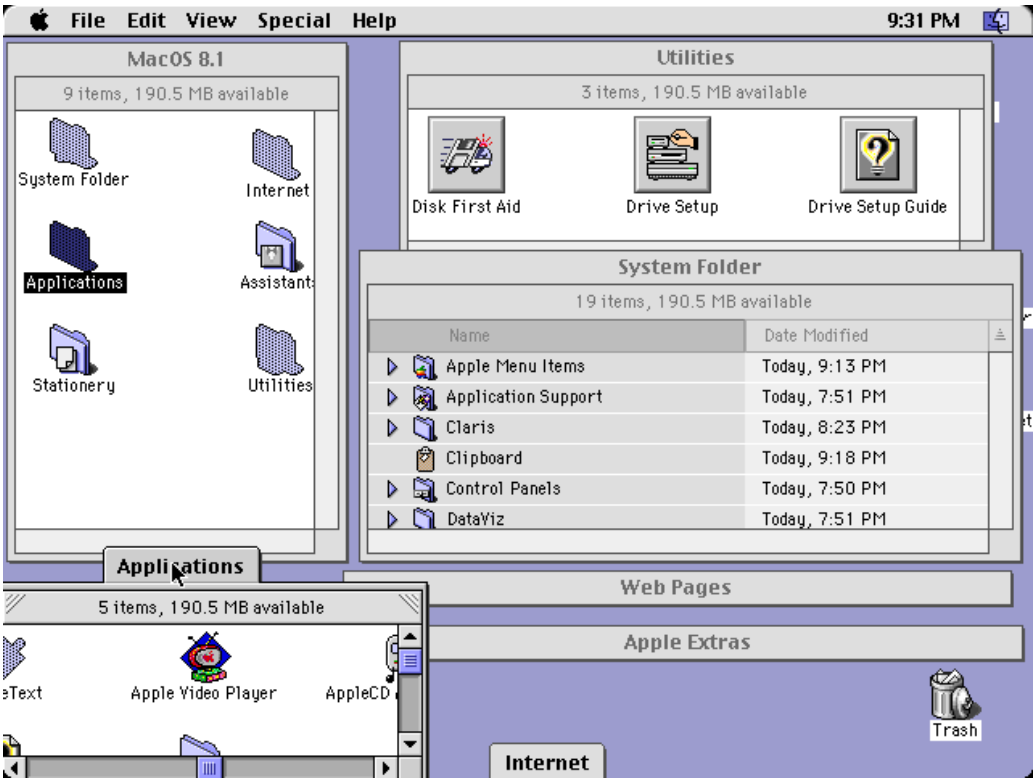

Obrázek 3.15: Ukázka prostředí Mac Systemu 7

Mac System 7 byl vydán roku 1994 a jeho další verze 7.5 roku 1996. Plocha vypadá téměř stejně jako v původním Mac System 1, viz Obrázek 3.15. Novinkou jsou vyskakovací okénka (tooltipy), která se objeví po najetí myši na ikonu nebo položku menu baru. Okénko obsahuje stručnou nápovědu, která říká co daná položka menu dělá, nebo k čemu slouží ikona. Další zajímavou novinkou je možnost měnit barvy ikon podle kategorií, kterou reprezentují.

Ve verzi 7.5 se objevil task bar (Control Strip) v levé spodní části obrazovky, viz Obrázek 3.16. Obsahoval tlačítka s často používanými akcemi, jako jsou změna hlasitosti nebo rozlišení obrazovky. Protože byl panel rychle přístupný přímo ploše, urychloval práci uživatele. Pomocí klávesové zkratky, nastavitelné v nastavení samotného Control Stripu, bylo možné panel skrýt a znovu zobrazit.

Obrázek 3.16: Control Strip v Mac System 7

### 3.3.4 Mac OS 8




Obrázek 3.17: Záložky a minimalizované aplikace v Mac OS 8 [10]

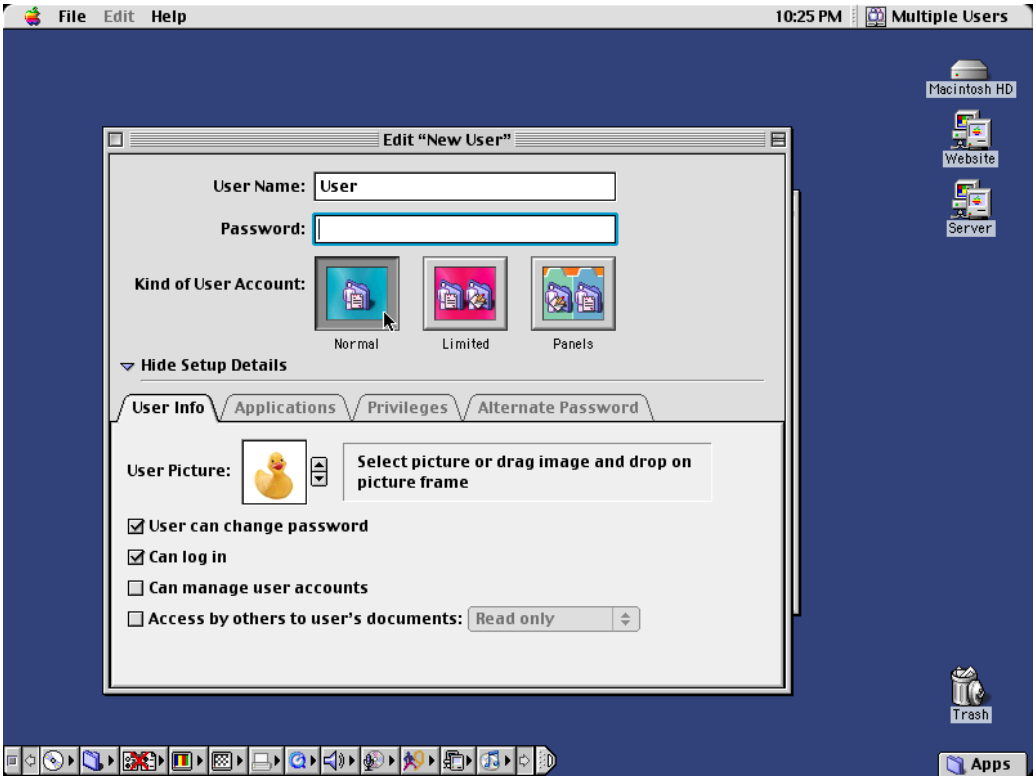
Mac OS 8 byl vydán roku 1997, a jeho poslední verze roku 1999. Asi nejpatrnější změnou je 3D vzhled oken a tlačítek, které umožňuje přidaný program jménem Appearance Manager. Další novinkou oproti předchozím Mac systémům je možnost nastavení barevného pozadí plochy a obrázků, místo klasického monochronního vzoru. Apple logo v levém horním rohu může být černé nebo barevné, podle nastavení monitoru. Na obrázku 3.18 vidíme Control Strip, umístěný ve spodní části obrazovky, který nyní obsahuje více funkcionalit než předchozí verze Macu a lze ho libovolně zmenšovat či zvětšovat a přidávat do něj další položky.

Obrázek 3.18: Control Strip v Mac OS 8 [2]

Soubory lze zobrazit jako ikony, tlačítka nebo seznamy. Okno s otevřeným adresářem lze přetáhnout do spodní části obrazovky, kde se přichytí jako záložka, kterou lze vytáhnout nahoru a zvětšit, viz Applications a Internet

na obrázku 3.17. Okna, která zrovna nepoužíváme, lze zmenšit na úzký pruh obsahující pouze název okna, viz Web Pages a Apple Extras na obrázku 3.17.

### 3.3.5 Mac OS 9




Obrázek 3.19: Ukázka prostředí Mac OS 9 [10]

Mac OS 9.2.2, vydaný roku 1999, je poslední verze Mac OS založená na originálním operačním systému Macintosh. Je to také první Mac systém, který umožňoval mít na počítači několik uživatelských účtů. Každý uživatel měl vlastní plochu a rozdílná nastavení vzhledu a zvuku, viz Obrázek 3.19.

Control Strip, který byl v Mac Systemu 7 a 8 velmi jednoduchý, se postupem času rozvíjel a obsahoval stále více funkcí. Ve verzi Mac OS 9 stačí jednoduše přetáhnout do panelu věci, které chceme přidat, místo abychom je museli složitě přidávat do modulů. Stejným způsobem funguje i odebírání položek, které nevyužíváme. V Mac OS 9 obsahoval Control Strip defaultně mnoho funkcionalit, například stav baterie, vzdálený přístup, nastavení hlasitosti a nastavení tiskáren.

### 3.3.6 OS X 10

Zatímco všechny předchozí verze Mac byly vytvořeny na základě originálního Mac Systemu 1 z roku 1984, Mac OS X 10, neboli Cheetah, je nový operační systém postavený na BSD Unix a NeXT technology. Tato změna proběhla především kvůli eliminaci technologických omezení ve starých Mac systémech. Při těchto změnách bylo také zcela přepracováno uživatelské rozhraní[10].

Grafické uživatelské rozhraní, které používají všechny následující verze systému OS X, se jmenuje Aqua. Hlavní myšlenkou Aqua rozhraní je především průhlednost a odlehčenost vzhledu. Oproti předešlým verzím je zde množství vizuálních vylepšení, která sice neposkytují žádné funkční využití, ale vypadají dobře a poskytují uživateli příjemnější pracovní prostředí. Příkladem je průhlednost překrývajících se oken, animace při manipulaci s okny nebo odlesky ikon umístěných v Docku.


Obrázek 3.20: Ukázka Docku a rozbalovacích menu [10]

Dock je prvek rozhraní dominující ve všech OS X systémech, který nahradil Control Strip z předchozích verzí OS X. Dock je průsvitná lišta ve spodní části obrazovky obsahující poměrně velké ikony nejčastěji používaných aplikací, adresářů a minimalizovaných oken. Adresáře umístěné v Docku lze procházet v sérii rozbalovacích menu, viz Obrázek 3.20.

Každé okno má v levém horním rohu tři barevná kulatá tlačítka, která slouží pro minimalizaci, maximalizaci a zavření okna.

Systém nabízel tři různé způsoby procházení adresářů v okně:

1. Standardní zobrazení s ikonami adresářů a souborů.
2. Názvy adresářů a souborů jsou vypsány v prvním sloupci a v dalších sloupcích jsou informace o nich(poslední modifikace, velikost a další).
3. Zobrazení ve sloupcích, které je pro Mac systémy nové, tento způsob prohlížení byl převzat z uživatelského rozhraní systémů NeXT. Obsah každého otevřeného adresáře se zobrazuje v novém sloupci.

Následující verzí systému byl OS X 10.1 (Puma), který byl dostupný jako bezplatný upgrade a nepřinesl žádné zásadní novinky. Poté následoval OS X 10.2 (Jaguar), který využíval větší průhlednosti.

### 3.3.7 OS X 10.3

Tento systém s názvem Panther byl vydaný roku 2003 a přinesl několik změn v uživatelském rozhraní. Jednou z výraznějších změn je nový vzhled většiny oken, místo pruhovaného stylu z původního rozhraní Aqua mají nyní vzhled broušeného kovu. Finder má nyní na levé straně nový postranní panel, který zobrazuje ikony nejpoužívanějších adresářů, mezi které patří například Dokumenty, Hudba a Aplikace.

Novou funkcí je Exposé, které najednou zobrazí uživateli všechna otevřená okna. Při použití této funkce se nejdříve všechna okna oddálí a přeorganizují, aby byla vidět všechna bez překrývání.

Při vytažení ikony z Docku se ikona automaticky vymaže, bez jakéhokoliv varování nebo nutnosti potvrzení, což může být velice matoucí pro uživatele systémů Windows, kteří očekávají přesun ikony na plochu.

### 3.3.8 OS X 10.4

Tato verze systému má název Tiger a byla vydána roku 2005. Uživatelské prostředí je velice podobné předchozí verzi systému, ale obsahuje několik nových funkcionalit. Místo stylu broušeného kovu jsou nyní okna lesklá v bílé barvě. S různými modely Macu je spojen určitý software a hardware, což ovlivní defaultní ikony zobrazené v Docku, který vypadá víceméně stejně jako u předchozí verze systému. Novinkou je funkce nazvaná Dashboard, která po spuštění zobrazí námi zvolené widgety překrývající veškerý další obsah na ploše či jinde.

### 3.3.9 OS X 10.5

Systém s označením Leopard, vydaný roku 2007, přinesl mnoho změn v rozhraní Aqua. Veškerá okna s Aqua i kovovým vzhledem mají nyní jednotnou šedou barvu, bez klasických šedých proužků. Rozdíl mezi právě používaným a neaktivním oknem je více zřetelný díky výraznějším stínům a barvám panelů oken. Lehce se změnil tvar kontextových oken, které mají nyní zaoblené rohy. Dock dostal nový 3D vzhled s odlesky a souvisí s ním nová funkce nazvaná Stacks. Pokud do Docku vložíme složku, zobrazí se soubory dané složky jako ikony s názvem souboru, viz Obrázek 3.21. Defaultně je takto přidána do Docku složka s dokumenty a staženými soubory. Nástroj Spaces změnil způsob práce s virtuálními plochami, zobrazuje náhledy virtuálních ploch vedle sebe do mřížky a umožňuje otevírat a přehazovat aplikace mezi plochami.

Následující verzí systému byl OS X 10.6, neboli Snow Leopard, který ale nepřinesl v rozhraní Aqua žádné výrazné změny.


Obrázek 3.21: Stack v Docku

### 3.3.10 OS X 10.7

Další v řadě OS X systémů je systém s označením Lion, který opět přinesl větší změny uživatelského prostředí. Byly odstraněny klasické posuvníky s šipkami, místo nich máme poloprůhledné posuvníky, které se zobrazí pouze když se obsah okna posouvá. Novinkou je také funkce Fullscreen, která zcela maximalizuje aplikaci tak, že na obrazovce nezůstane nic jiného než obsah aplikace, zmizí tedy Dock i horní lišta s tlačítky a menu. Při manipulaci s okny se také objevily nové animace, např. nově otevřené okno přiletí na

vrch ostatních oken. Následující systémy byly OS X Mountain Lion a OS X Mavericks, které opět nepřinesly žádné velké změny v rozhraní.

### 3.3.11 OS X 10.10

Tato verze s názvem Yosemite byla vydána roku 2014 a přinesla výrazně přepracované grafické prostředí ve stylu mobilního systému iOS. V celém prostředí převládá hlavně průhlednost a plochý, minimalistický vzhled. Většina prvků a ikon z rozhraní Aqua byla zjednodušena, vše má nyní jednoduchý, plochý design, podobně jako v systému iOS 7. Některé komponenty jako přepínače nebo zaškrtávací políčka získala animace, zatímco u jiných prvků (např. animace při odstraňování položky z Docku) byly animace odstraněny. Další novinkou je možnost měnit téma celého systému na světlé nebo tmavé.

Následující verzí systému je OS X 10.11, neboli El Capitan, jehož prostředí zůstalo téměř celé zachováno z předchozí verze. Novinkou je například funkce Split View, která rozdělí dvě aplikace vedle sebe přes celou obrazovku ve fullscreen režimu.

Aktuálně nejnovější verzí operačního systému je MacOS 10.12 Sierra, která vyšla roku 2016, viz Obrázek 3.22. Uživatelské prostředí zůstalo ve stejném stylu jako předešlé systémy a nenastaly zde žádné razantní změny. Zajímavou novinkou je implementace Siri a možnost otevírání nových oken stejné aplikace nebo Finderu do jednotlivých karet, namísto nového okna.


Obrázek 3.22: Ukázka prostředí MacOS Sierra

## 3.4 Linux

Linux nebo také GNU/Linux je svobodný a open-source operační systém založený na Unixovém jádru a jeho autorem je Linus Torvalds.

Je rozdíl mezi distribucí Linuxu a Linuxem. Distribuce je konkrétní operační systém s předpřipravenými aplikacemi a uživatelským prostředím (textovým nebo grafickým), zatímco samotný Linux je pouze jádro operačního systému.

Grafické prostředí, které Linux využívá primárně se jmenuje X Window System, viz kapitola 2.3. Existuje několik výchozích desktopových rozhraní, mezi nejznámější patří GNOME a KDE. Tato různá prostředí mohou bez problémů koexistovat na jediném systému – a to nejen tak, že jsou nainstalována současně, ale lze z jednoho spouštět aplikace patřící k prostředí druhému. K tomu jsou potřeba všechny knihovny a další potřebné součásti. Existují programy, které nepatří do žádného grafického prostředí, přestože využívají grafický framework, na němž je nějaké prostředí založeno. Příkladem jsou webové prohlížeče (Mozzila Firefox využívá GTK+, Opera Qt). Místo celého grafického prostředí lze instalovat jen část, např. aplikace nebo knihovny. Jelikož je Linux open-source, existuje velké množství různých distribucí. Jednotlivé distribuce mohou být zcela rozdílné, velké množiny distribucí ale sdílejí podstatnou část své struktury. Vyberu a popíšu tedy několik zástupců od každého druhu uživatelských rozhraní.

### 3.4.1 KDE

KDE[1] (K Desktop Environment) je jedním z velkých komplexních grafických prostředí pro Unixové systémy. Při vývoji KDE bylo hlavním cílem vytvořit jednoduché a přehledné GUI pro koncové uživatele a především jednotný a moderní vzhled pro všechny aplikace. KDE má svého správce oken, různé pomocné aplikace prostředí, mnoho uživatelských aplikací a je založeno na frameworku Qt. Qt je framework, který ulehčuje vytváření aplikací, především grafických uživatelských rozhraní a nabízí velké množství knihoven. Je využit jako základ pro většinu KDE aplikací a také pro Plasma rozhraní. Mezi distribuce, které defaultně využívají KDE, patří např. Kubuntu[2] nebo Chakra[3].
Základním prvkem prostředí je Kicker (panel podobný hlavnímu panelu ve Windows) ve spodní části obrazovky, na kterém je hlavní nabídka a spuštěné

[1] `https://www.kde.org`
[2] `http://www.kubuntu.org`
[3] `https://chakralinux.org`

programy. Tím může připomínat práci v prostředí Windows, ale na rozdíl od tohoto systému je možné přidávat další panely na libovolnou stranu obrazovky. Panely se také mohou skrývat a zobrazovat až po najetí kurzoru myši. V levé části panelu je několik tlačítek, která otevírají nabídku aplikací nebo nastavení a mohou obsahovat další sub-menu. V pravé části panelu jsou systémové ikony. Panel na oknech obsahuje tři typická tlačítka pro zavření, minimalizaci a maximalizaci okna.

KDE se vyznačuje především velkou přizpůsobitelností většiny prvků rozhraní a spoustou nastavitelných funkcí. Můžeme si libovolně nastavit vzhled, počet a umístění panelů a tlačítek, která obsahují. Úpravy nejsou limitovány pouze na oblast plochy a hlavního panelu, měnit můžeme i prvky oken – např. si zvolíme tlačítka, která chceme zobrazit v panelu okna nebo veškerá tlačítka vymažeme. Nastavitelné jsou také různé animace oken.

Vývoj KDE lze rozdělit do pěti hlavních verzí, které pak obsahovaly další menší verze a aktualizace.

Na obrázku 3.23 vidíme KDE 1, první verzi systému KDE vydanou roku 1998. Již v této verzi platila velká přizpůsobitelnost vzhledu, přes barevná schémata po rozvržení umístění, počtu a vzhledu ikon na panelech.

KDE 2, vydaný roku 2000, přinesl mnoho technologických změn a vylepšení. Ve vývoji GUI bylo cílem především více intuitivní a uživatelsky přívětivé ovládání. Byla rozšířena konfigurovatelnost jednotlivých prvků prostředí, například nastavitelnost různých motivů pro celkový vzhled i ikony nebo volitelné rozvržení prvků hlavního panelu a menu, viz Obrázek 3.24. Pro snadnější práci uživatele byly přidány konfigurovatelné klávesové zkratky, navigování na ploše pomocí klávesnice místo myši a výběr z přibližně třiceti jazyků prostředí.

Na obrázku 3.25 vidíme verzi KDE 3 vydanou roku 2002. Rozhraní systému bylo lokalizováno do více než padesáti jazyků a novinkou bylo velké množství nových aplikací a nástrojů, například nástroj SuperKaramba, který umožňuje přidávat na plochu různé widgety. Oproti starší verzi systému je možné více přizpůsobit plochu, na výběr je mnoho motivů, widgetů a setů ikonek. Nastavit si také můžeme průhlednost konzole.

KDE Plasma 4, později označována jako KDE Plasma Workspace, je termín pro čtvrtou generaci grafických prostředí od KDE a byla vydána roku 2008. Plasma nahradila čtyři klíčové prvky prostředí z předchozích verzí systému:

1. Hlavní panel, který nahradil Kicker z předchozích verzí systému.

2. Plocha, tedy hlavní pracovní prostředí, nahradila původní KDestop shell.

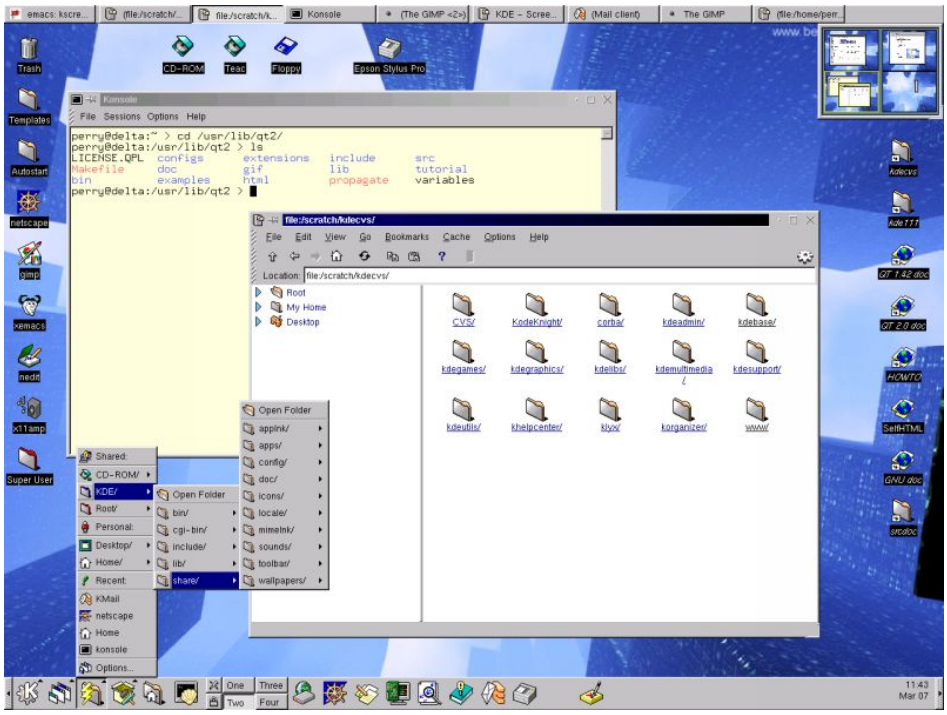
3. Folder Views je widget, který shlukuje adresáře zobrazené na ploše a poskytuje rychlý přístup k základní práci se soubory a adresářem. Ve verzi 4.0 nebylo možné umístit ikony adresářů a souborů přímo na plochu, ale pouze do tohoto widgetu, viz Obrázek 3.26. V následujících verzích se od této funkce odstoupilo a bylo možné umisťovat ikony přímo na plochu.

4. Plasma toolbox v levém horním rohu obrazovky a na hlavním panelu slouží k rychlému přístupu ke konfiguraci widgetů a dalším nastavením.

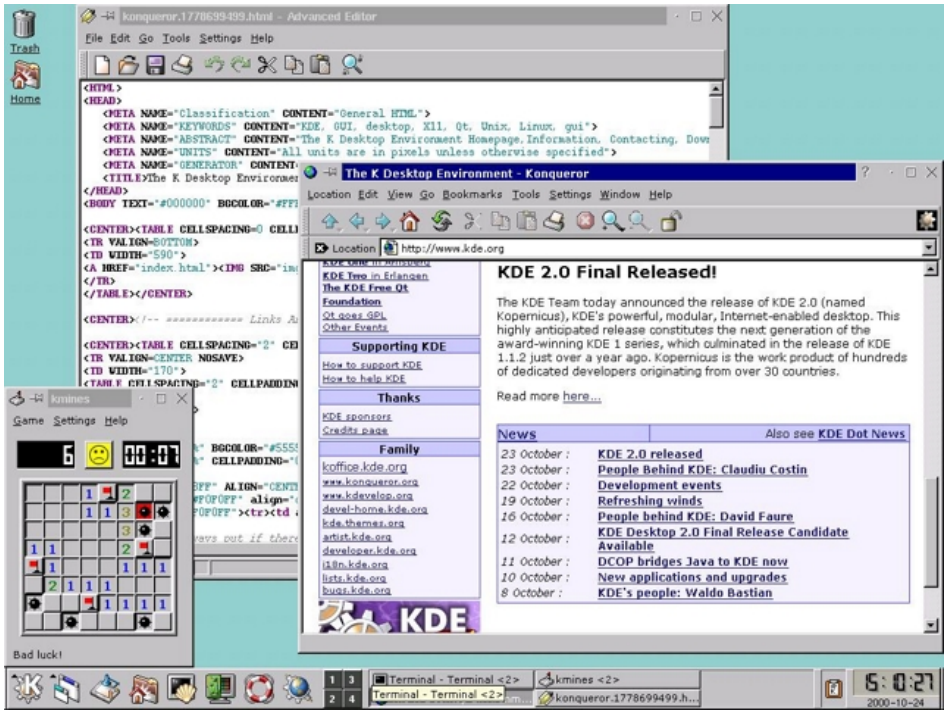
Při vývoji Plasmy se dbalo především na vytvoření dynamického a vysoce upravitelného prostředí. Tato verze systému přinesla mnoho vylepšení GUI, například různé animace, hladkou a plynulou práci s okny a podporu widgetů. Velikost ikon je snadno nastavitelná a téměř každý prvek rozhraní lze snadněji konfigurovat. Některé z nápadnějších změn zahrnují nové realističtější ikony a zvuky.

KDE Plasma 5 byla poprvé vydána roku 2014 a její zatím poslední verze v lednu 2017. Hlavní změny spočívají v aktualizovaném, modernějším a čistějším grafickém prostředí, nazvaném Breeze. Tento motiv je vysoce kontrastní a jednoduchý a odstraňuje nepřehlednost a nepořádek v celém pracovním prostředí, viz Obrázek 3.27

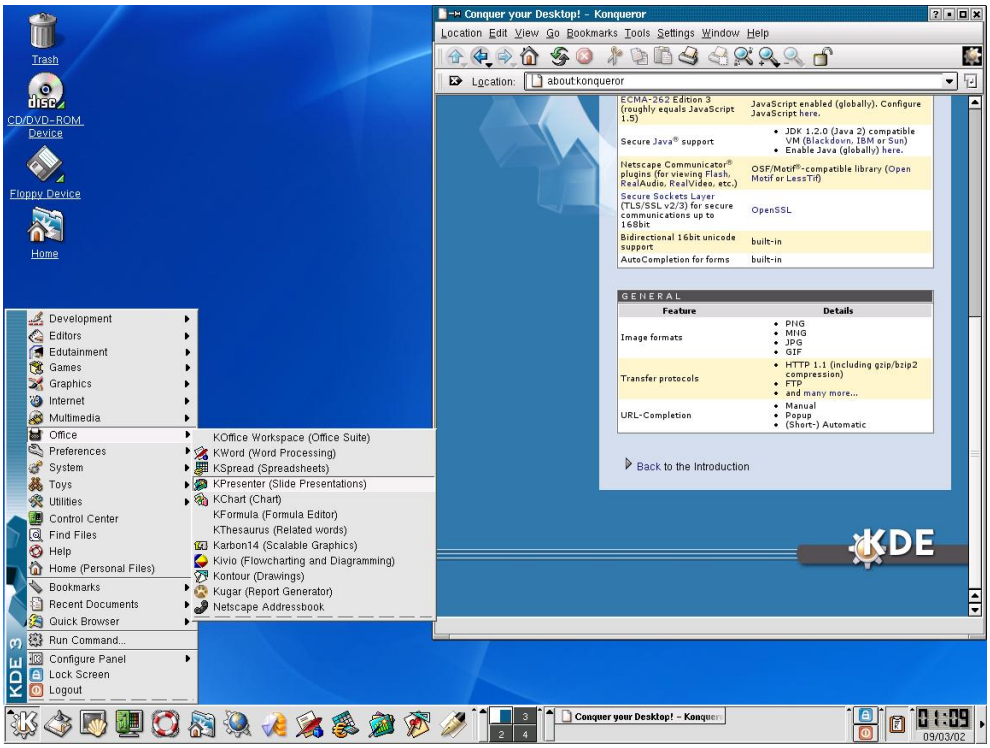
Detailní popis jednotlivých verzí KDE a vývoje prvků jejich grafického prostředí naleznete `https://www.kde.org/announcements/`.



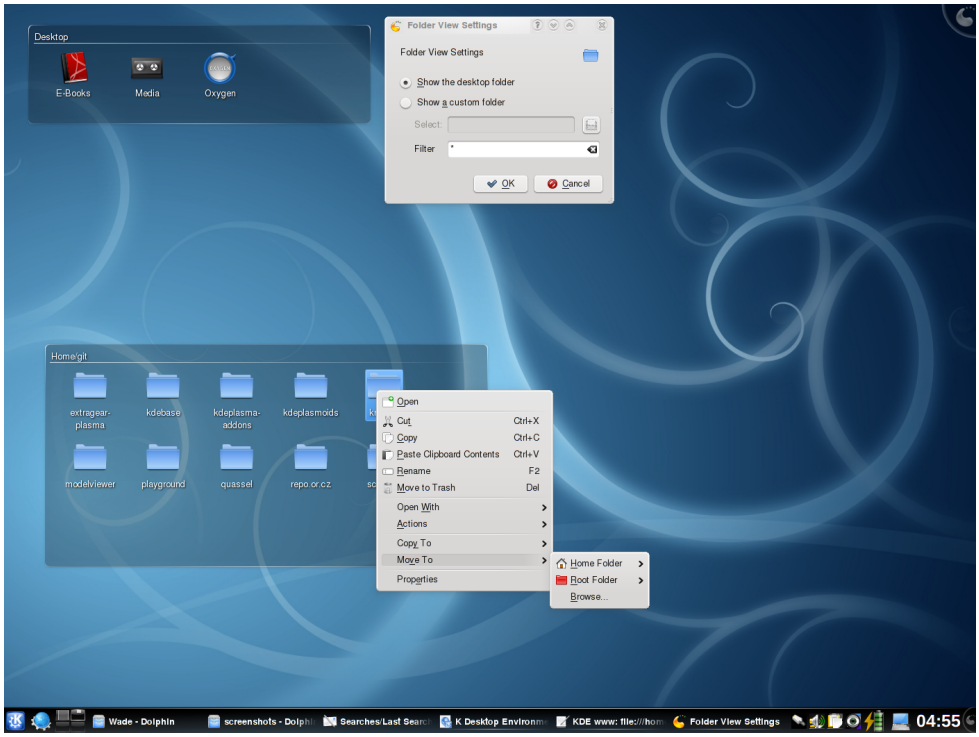
Obrázek 3.23: Ukázka prostředí KDE 1



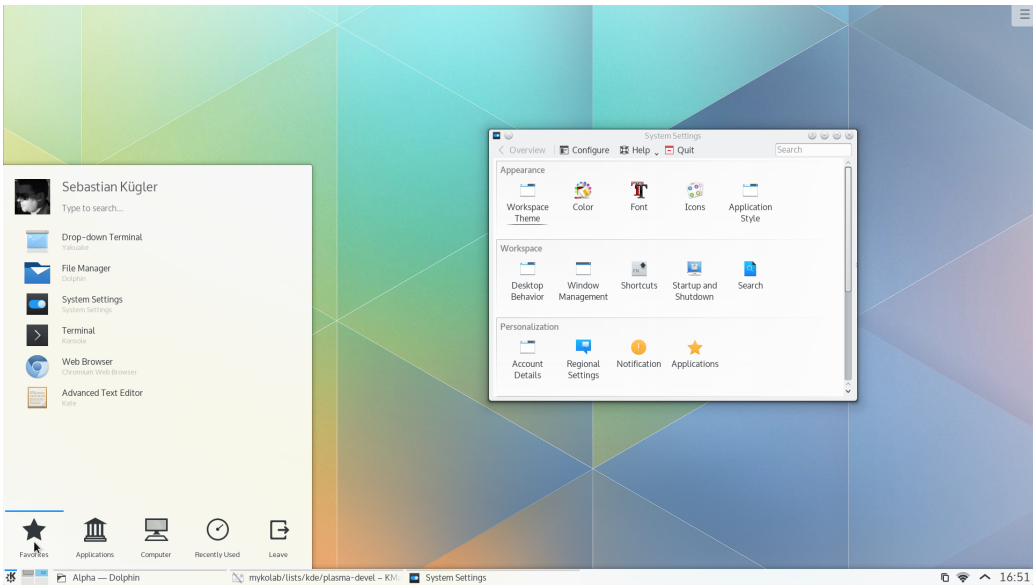
Obrázek 3.24: Ukázka prostředí KDE 2




Obrázek 3.25: Ukázka prostředí KDE 3




Obrázek 3.26: Ukázka prostředí KDE Plasma 4




Obrázek 3.27: Ukázka prostředí KDE Plasma 5

### 3.4.2 GNOME

GNOME[4] je druhé z populárních desktopových prostředí Linuxu a je založen na knihovně GTK+. První verze, kterou si můžete prohlédnout na Obrázku 3.28, byla vydána roku 1999. GNOME prostředí má typicky dva hlavní panely – jeden v horní části a druhý naopak v části spodní. Na horním panelu najdeme hlavní nabídku programů a oznamovací oblast, obsahující systémové informace (připojení, hlasitost a další) a ikonky programů běžících na pozadí. Na spodním panelu najdeme záložky spuštěných programů a přepínač virtuálních ploch. Dalším typickým prvkem tohoto prostředí je panel v levé části obrazovky (Dash), který obsahuje ikony právě spuštěných aplikací a dalších volitelných aplikací. Panel lze nastavit tak, že se objeví až po najetí myši, jinak bude skrytý. Dalším typickým prvkem rozhraní je obrazovka s přehledem všech aplikací, zobrazených jako velkých ikon. Přesné rozvržení a obsah jednotlivých panelů, prvků a vzhled celkového vzhledu prostředí záleží na konkrétní verzi a distribuci, která využívá GNOME. Všechny aplikace sdílejí stejný vzhled a dělají tím rozhraní jednodušší a přívětivější pro uživatele, kteří s Linuxem začínají. Na první pohled nemusí GNOME působit stejně elegantně a profesionálně jako KDE, ale má stejně jako KDE mnoho možností, jak si prostředí přizpůsobit. GNOME obecně preferuje výběr nejdůležitějších ovládacích prvků, které jsou často zvětšené. Díky tomu je rozhraní výrazně jednodušší a více čitelné. Mezi distribuce, které mají

[4] `https://www.gnome.org`

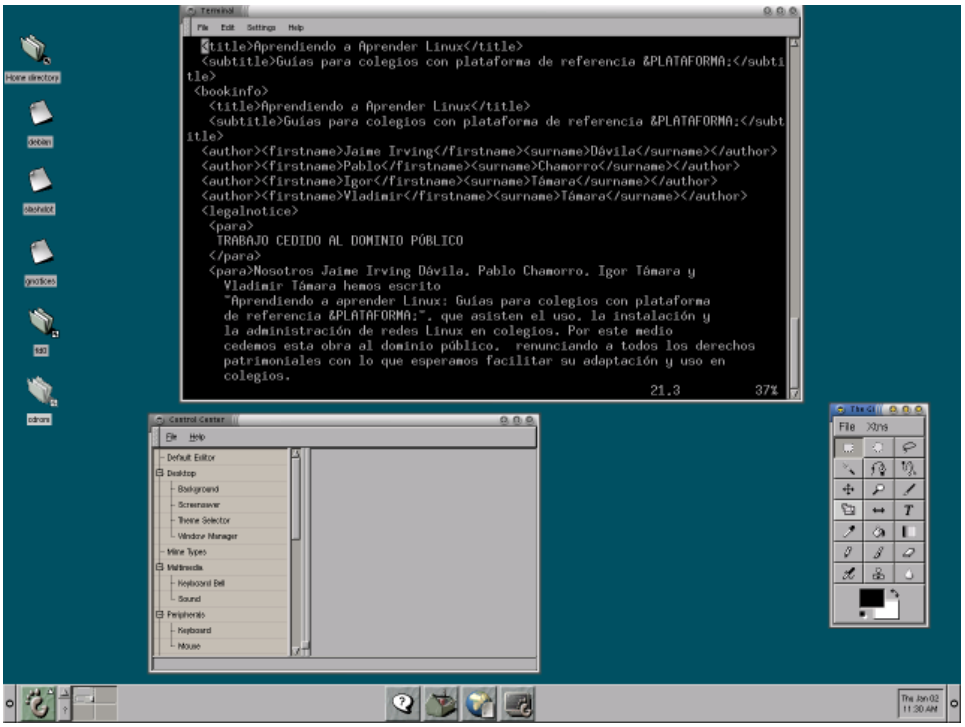
GNOME jako defaultní desktopové prostředí patří například Fedora[5] nebo Ubuntu GNOME[6].

GNOME 2.0, vydaný roku 2002, byl založen na nové a vylepšené GTK+ 2. Celkový vzhled prostředí vypadá na první pohled velice podobně jako předchozí GNOME 1, viz Obrázek 3.29. Novinkou je práce s textem, je podporován Unicode a možnost pracovat s vloženými obrázky a vyhlazenými styly písma. K dispozici jsou nové sady ikon a barevných palet a motivů. S postupným vydáváním dalších aktualizací a verzí vypadá prostředí a jeho prvky stále více moderně a realisticky, ale základní rozvržení prostředí zůstává téměř stejné, viz předchozí odstavec.

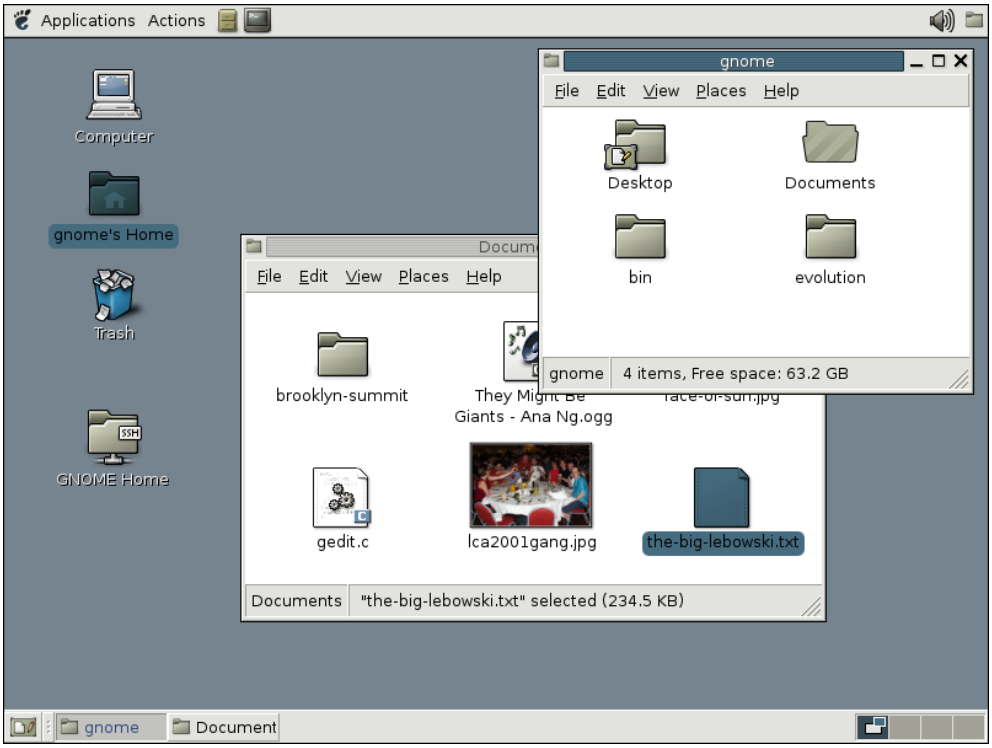
GNOME 3.0 byl vydaný roku 2011 a přinesl velké změny ve vzhledu prostředí, jak můžeme vidět na Obrázku 3.30. Tradiční panel ve spodní části obrazovky je pryč, stejně jako tlačítka pro manipulaci s okny. Přetažením okna do horní části obrazovky dojde k maximalizaci, minimalizaci provedeme kliknutí na panel pravým tlačítkem myši. Prostředí ale podporuje zobrazení každého okna zvlášť na vlastní virtuální ploše. V základním nastavení tedy zbývá pouze horní panel s tlačítkem Aktivit, názvem aktuálně používané aplikace a systémovými informacemi. Správu oken či aplikací najdeme v přehledu aktivit, což je obrazovka obsahující klasický postranní panel s aplikacemi, uprostřed přehled ikon všech aplikací nebo otevřených oken a vpravo přehled virtuálních ploch. Veškeré vyhledávání, spouštění aplikací a přepínání mezi virtuálními plochami a okny probíhá v tomto režimu.

Detailní popis jednotlivých verzí GNOME a vývoje prvků jejich grafického prostředí naleznete `https://help.gnome.org/misc/release-notes/`.

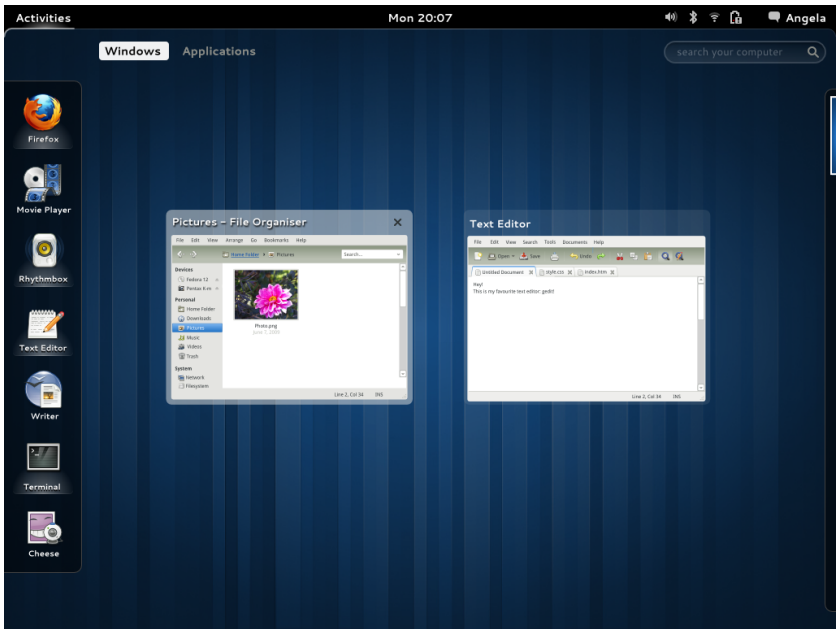
[5] `https://getfedora.org`
[6] `https://ubuntugnome.org`



Obrázek 3.28: Ukázka prostředí GNOME 1.0




Obrázek 3.29: Ukázka prostředí GNOME 2.6




Obrázek 3.30: Otevřená okna v přehledu Aktivit v GNOME 3

### 3.4.3 Další desktopová prostředí

1. Xfce[7] vzniklo jako odlehčená verze grafického prostředí v kontrastu k prostředím KDE a GNOME. Xfce je velmi rychlé prostředí, které disponuje mnoha nenáročnými programy a oproti předešlým prosředím má jednodušší vzhled. Díky hardwarové nenáročnosti lze tak distribuce využívající Xfce nasadit i na slabších počítačích, které disponují slabším procesorem a menším množství paměti. Mezi distribuce, které defaultně využívají Xfce, patří např. Xubuntu[8].

2. LXDE[9] je další hardwarově nenáročné prostředí, je tedy vhodné především pro starší počítače. I přes svojí úspornost je prostředí bohaté na grafické prvky, které jsou jednoduché a snadno ovladatelné. V základním nastavení připomíná rozhraní systémů Windows, s jediným hlavním panelem ve spodní části a menu v levé části panelu. Ale stejně jako u ostatních prostředí je zde mnoho možností přizpůsobitelnosti vzhledu. Toto prostředí je defaultně nastavené například v distribuci Lubuntu.

3. Existuje mnoho dalších desktopových prostředí, kde většina z nich vychází z nějakého již zmíněného (např. GNOME 2.x nebo KDE).

[7] `https://www.xfce.org`
[8] `https://xubuntu.org`
[9] `http://lxde.org`

# 4 Analýza GUI komponent

Obecně bylo v prvních verzích analyzovaných systémů méně grafických komponent a postupem času se vylepšovaly a objevovaly nové. Samozřejmostí je neustálé vylepšování prvků po grafické stránce, vypadaly stále více realisticky, nejdříve například použitím 3D vzhledu, stínování a později průhledností a změnou tvaru (např. zaoblené rohy nebo ostré linie prvků). Většina základních komponent, jako jsou tlačítka, posuvníky, přepínače, tabulky a další byly již v prvních verzích systémů. Čím více komponent v systému přibývalo, tím měli uživatelé větší možnosti přizpůsobitelnosti prostředí. Některé nové komponenty či funkce systému jsou popsány ve třetí kapitole u jednotlivých verzí systémů. Pokud byla některá komponenta nebo funkce v novějších verzích systému odstraněna, nalezneme varování a její možnou náhradu v API dokumentaci[4].

Nastaly i takové situace, kdy byla jedna komponenta zaměněna za jinou, např. klasický posuvník (scroll bar) se využíval pro úpravu hlasitosti nebo pro zobrazení průběhu nějaké akce (progress bar). Dnes už existují vlastní komponenty pro tyto funkce a není tedy třeba je zaměňovat za jiné[6].

Zde je příklad některých dalších komponent, které se objevily až v pozdějších verzích Windows systémů a nebyly popsány v předchozí kapitole:

1. ListBox a ComboBox – Windows 3,
2. Tool bar – Windows 3.11,
3. Date/Time picker – Windows 3,
4. ComboBox Ex – Windows 95 (obsahuje funkce navíc oproti obyčejnému comboBoxu - lze přidat obrázek vedle textu a je možné použít různé obrázky pro jednotlivé položky seznamu),
5. Tooltip – Windows XP (tooltipy v bublině, většina programů ale stále využívá klasické tooltipy - obdélníkový tvar a černý text na žlutém pozadí),
6. Split button – Windows Vista (tento typ tlačítka má oproti obyčejnému tlačítku rozbalovací seznam, typicky sloužící pro změnu chování (funkce) tlačítka před kliknutím).

Při porovnání systémů Windows a Mac se můžeme zaměřit na hlavní menu bar, který oba systémy implementují jiným způsobem. GUI ve Windows i Mac systémech podporuje práci na více monitorech, které se postupem času stále více zvětšují. Tato funkcionalita byla přidána kvůli měnícím se požadavkům uživatelů. Podpora a nastavení pro práci na více monitorech byly přidány ve Windows 98 a Mac je implementoval jako standardní v roce 1987. Při detailní analýze prvků uživatelského rozhraní Windows a Mac systémů najdeme mnoho podobností, ať už ve vzhledu nebo funkčnosti komponent. Oba operační systémy ale zvolily rozdílný způsob implementace menu barů. Ve Windows má každé okno svůj vlastní menu bar na rozdíl od Mac systémů, kde všechna okna sdílejí jeden menu bar v horní části obrazovky. Umístění menu baru pak může ovlivnit efektivitu práce uživatele v závislosti na používaném počtu monitorů a jejich velikosti. Na menších monitorech je logicky vzdálenost (tedy i přesun) mezi menu barem a příslušným oknem menší než na velkých monitorech, kde tato činnost zabere více času[9]. U Linuxových distribucí si pak můžeme nastavit libovolný počet i umístění menu barů.

# 5 Emulátory operačních systémů

Při psaní této práce jsem využila několik emulátorů operačních systémů, které mi pomohly především při práci na třetí kapitole – Historie a vývoj operačních systémů, viz kapitola 3, kde jsem popisovala uživatelská rozhraní a jeho grafické prvky.

Na stránce `http://www.pcjs.org` v kapitole Demos je k dispozici několik emulátorů spustitelných přímo v prohlížeči. Vyzkoušet si můžete například Windows 1.01, Windows 3, Windows 95 a další operační systémy, které jsem v této práci nepopisovala.

Na stránce `https://jamesfriend.com.au/pce-js/mobile/` naleznete emulátor Mac Systemu 1.

Na stránce `https://jamesfriend.com.au/pce-js/pce-js-apps/` naleznete emulátor systému Mac OS 7.

Poslední je emulátor systému Xerox Alto, který bohužel nelze spustit přímo v prohlížeči, proto jsem ho i s návodem na spuštění a použití přidala do příloh této práce.

# 6 Programátorská rozhraní pro tvorbu uživatelských rozhraní

Z pohledu programátora je operační systém definován svým API (Application Programming Interface), což je rozhraní určené pro programování aplikací. Obsahuje volání funkcí, které může aplikační program žádat od operačního systému a také obsahuje definice souvisejících datových typů a struktur [16]. API také určuje, jakým způsobem jsou funkce knihoven volány ze zdrojového kódu programu.

## 6.1 Windows API

Obecně platí, že Windows API zůstává konzistentní od Windows 1.0, můžeme tedy vzít zdrojový kód napsaný ve Windows 1 a zkompilovat ho bez větších změn na Windows 10. Změny v jednotlivých verzích API se týkají především rozšiřování a přidávání nových funkcí. Windows 1.0 podporovaly méně než 450 funkcí, zatímco dnes jich jsou tisíce[16]. Určité změny ve Windows API a jeho syntaxi nastaly při přechodu z 16bitové na 32bitovou architekturu. V 16bitových verzích systému (od Windows 1.0 po Windows 3.1) byla velikost datového typu *int* v C 16 bitů, stejně jako ukazatel na segment a offset, což bylo matoucí a způsobovalo to rozdíly např. mezi typy *long* a *far*. Počínaje Windows NT a Windows 95 začaly Windows podporovat 32bitový paměťový model, kde byl datový typ int v C rozšířen na 32bitovou šířku.

API pro 16bitové verze se dnes nazývá Win16. Všechny 32bitové verze systému podporují jak Win16 kvůli kompatibilitě se staršími aplikacemi, tak Win32 pro nové aplikace. Některé funkce, které byly k dispozici ve Win16 však byly odstraněny, je proto nutné je zkontrolovat při přenášení aplikací mezi různými verzemi API. Veškeré komponenty a funkce nalezneme v API dokumentaci[4]. Současně existovaly dvě další sady API. Win32s („s“ jako subset) bylo rozhraní, které umožňovalo psát 32bitové aplikace pracující pod Windows 3.1. Windows 95 API se také nazývalo Win32c („c“ jako compatibility), ale od tohoto termínu se upustilo. Posledním typem je Win64, což je varianta API pro 64bitové systémy. Programování zůstává téměř stejné jako ve Win32 s rozdílem velikosti ukazatelů, která je zde 64 bitů.

### 6.1.1 Struktura programu

Tak jako v progamu C je výchozím bodem programu funkce *main*, výchozím místem programu pro Windows je funkce *WinMain*, jejíž syntaxi vidíme v ukázce 6.1:

```
1 int WINAPI WinMain(HINSTANCE hInstance,
2                     HINSTANCE hPrevInstance,
3                     LPSTR lpCmdLine,
4                     int nCmdShow);
```

Ukázka kódu 6.1: Funkce WinMain

První parametr je handle instance aplikace, což je unikátní identifikátor instance naší aplikace. Handle obecně je číslo odkazující na nějaký objekt. Druhý parametr je handle předchozí instance, který měl význam ve Win16, nyní je vždy *NULL*. Ve Win16 sloužil tento parametr pro překopírování paměti z předchozí instance do nové. Další parametr udává jak bude okno zobrazeno (maximalizované, minimalizované nebo v nastavených rozměrech) a poslední parametr je příkazový řádek používaný pro spuštění programu. U některých pojmenování proměnných si můžete všimnout zvláštních názvů. Je běžné používat pro pojmenování proměnných konvenci známou jako Maďarská notace. Název proměnné začíná malým písmenem nebo písmeny, která určují její typ. Příkladem je *HINSTANCE*, kde h znamená handle a *nCmdShow*, kde n znamená typ *short* [16].

Další částí programu je smyčka zpráv, která přijímá zprávy odesílané operačním systémem nebo funkcní *main*. Když aplikace obdrží zprávu, tato smyčka ji předá funkci *WndProc* k dalšímu zpracování. Ve funkci *WndProc*[1] implementujeme příkaz *switch*, který má několik základních *casů* (viz. ukázka 6.2) [12]. *Case WM_PAINT* slouží pro vykreslení okna a jeho prvků. Další *case* je *WM_DESTROY*, který zničí okno a jeho prvky po jeho zavření a *WM_CREATE*, ve kterém vytváříme jednotlivé prvky rozhraní (např. tlačítka a seznamy), viz příklady v kapitole 7. *Case WM_COMMAND* slouží pro zpracování zpráv odeslaných prvky okna a následných akcí (např. při stisknutí tlačítka).

[1] `https://msdn.microsoft.com/en-us/library/windows/desktop/ms633573(v=vs.85).aspx`

```
switch (message) {
    case WM_PAINT: {
        PAINTSTRUCT ps;
        HDC hdc = BeginPaint(hwnd, &ps);
        EndPaint(hwnd, &ps);
    }
    case WM_CREATE:
    case WM_COMMAND:
    case WM_DESTROY:
        PostQuitMessage(0);
        break;
}
```

Ukázka kódu 6.2: Implementace switche ve funkci WndProc

## 6.2 Knihovna GTK+

GTK+ je multiplatformní toolkit pro tvorbu grafických uživatelských rozhraní, na kterém je založeno například GNOME rozhraní. Knihovna byla původně navržena pro GIMP (program pro práci s grafikou). Knihovna GTK+ je napsána v jazyce C. Jednotlivé verze GTK+ obsahují rozdílné volání některých funkcí a určité komponenty byly odstraněny nebo změněny. Přehled všech komponent a funkcí potřebných k jejich implementaci naleznete v referenční příručce[2]. U každé zastaralé komponenty či funkce je uvedeno upozornění a nový způsob implementace.

Pro správnou funkčnost programu je nutné přidat hlavičkový soubor *<gtk/gtk.h>*. Stejně jako v knihovně Qt je výchozím bodem programu funkce *main()*. Voláním *gtk_main()* spustíme hlavní smyčku aplikace, kde aplikace čeká na akce a vstup uživatele[14].

## 6.3 Knihovna Qt

Qt je stejně jako GTK+ multiplatformní toolkit pro tvorbu grafických uživatelských rozhraní. Je na něm založeno například KDE rozhraní pro Linuxové distribuce. Přehled všech aktuálních i zastaralých komponent, funkcí potřebných k jejich implementaci a krátkých tutoriálů naleznete v Qt dokumentaci[3].

[2] `https://developer.gnome.org/gtk3/stable/`

[3] `http://doc.qt.io/qt-4.8/qtgui-module.html`

Abychom zajistili správnou funkčnost programu, musíme přidat hlavičkové soubory *<QApplication>*, *<QtGui>* a *<QtGui/*>*, kde * je každá jednotlivá komponenta, např. Button. Výchozím bodem programu je funkce *main* (viz ukázka 6.3), ve které vytvoříme objekt třídy *QApplication*, který spravuje prostředky celé aplikace a je nutný pro všechny Qt programy mající GUI.

```
1 int main(int argv, char **args){
2     QApplication app(argv, args);
3
4     return app.exec();
5 }
```

Ukázka kódu 6.3: Funkce Main v Qt

Návratová hodnota funkce *main app.exec()* spustí hlavní smyčku aplikace. Dokud aplikace běží, jejím komponentám jsou zasílány akce, příkladem takové akce je stisknutí tlačítka myši nebo vstup z klávesnice.

## 6.4 Cocoa

Cocoa je nativní objektově orientované API pro operační systémy OS X od firmy Apple založené na modelu MVC (Model-View-Controller). Cocoa se skládá ze tří hlavních frameworků:

1. Foundation Kit obsahuje funkce, které nesouvisí přímo s grafickým uživatelským rozhraním.

2. Application Kit (AppKit) vychází přímo z původního NEXTSTEP Application Kitu a obsahuje kód, který mohou programy využít pro tvorbu a následnou interakci s grafickým uživatelským rozhraním.

3. Core Data slouží pro uchování dat a práci s daty.

Vývoj aplikací pro OS X obvykle probíhá v jazycích Swift nebo Objective-C, ale k dispozici máme i další, např. C nebo C++. Mezi známá vývojová prostředí od Applu patří především Xcode a Interface Builder.

Existují i další API od firmy Apple, např. Carbon, který umožňoval převod staršího softwaru na Mac OS X systémy a poskytoval tak zpětnou kompatibilitu pro programy ze zastaralých systémů (Mac OS 8 a 9). Od vývoje Carbonu se ale ustoupilo a aktuálním podporovaným API je tedy Cocoa.

Pro správnou funkčnost programu musíme importovat hlavičkové soubory *<Cocoa/Cocoa.h>* a vytvořit funkci *Main* jako výchozí bod programu, viz ukázka 6.4. V následující ukázce vidíme příklad funkce *Main*, kde každá

aplikace musí mít právě jednu instanci *NSAApplication*. *NSApp run* poté spustí hlavní smyčku aplikace.

```
int main(int argc, const char *argv[]) {
 NSApplication *application = [NSApplication sharedApplication];
 [NSApp run];

 return 0;
}
```

Ukázka kódu 6.4: Funkce Main

Protože jsem neměla k dispozici Mac zařízení a neoficiální verze operačního systému se mi nepodařila ve Virtual Boxu zprovoznit, ukázky kódu v následující kapitole jsem neotestovala ve vlastním programu jako u ostatních API. Ukázky pochází z referenční příručky Cocoa API [1] a dalších tutoriálů.

# 7 Srovnání API pro tvorbu uživatelských rozhraní

V jednotlivých sekcích této kapitoly uvedu příklady kódu pro tvorbu různých komponent grafického uživatelského rozhraní ve Windows API, knihovnách GTK+, Qt a Cocoa API. Ukázky ve Windows API a GTK+ jsou napsány v jazyce C, ukázky v Qt v C++ a Cocoa API v jazyce Objective C.

## 7.1 Vytvoření okna

Ve Windows API můžeme vytvořit okno velmi podobnými funkcemi CreateWindow nebo CreateWindowEx, které nabízí navíc další nastavení stylů. V ukázce kódu 7.1 vytváříme prázdné okno s názvem Hello World. *WS_OVERLAPPEDWINDOW* je typ okna s ohraničením a názvem v záhlaví. *WS_HSCROLL* a *WS_VSCROLL* zajistí, že bude mít okno horizontální i vertikální posuvník. *CW_USEDEFAULT* a *CW_USEDEFAULT* nastaví defaultní pozici pro okno, stejné parametry lze použít pro velikost okna.

```
1 HWND hWnd = CreateWindowEx(
2       0,
3       CLASS_NAME,
4       L"Hello World",
5       WS_OVERLAPPEDWINDOW | WS_HSCROLL | WS_VSCROLL,
6       CW_USEDEFAULT, CW_USEDEFAULT,
7       500, 400,
8       NULL, NULL, hInstance, NULL);
```

Ukázka kódu 7.1: Vytvoření prázdného okna ve Windows API

V následující ukázce 7.2 vytváříme nové okno v GTK+, které může obsahovat další komponenty. Místo parametru *GTK_WINDOW_TOPLEVEL* můžeme zvolit parametr *GTK_WINDOW_POPUP*, což je vyskakovací okno použitelné např. pro implementaci tooltipů.

```
1 GtkWidget *window;
2 window = gtk_window_new(GTK_WINDOW_TOPLEVEL);
3 gtk_widget_show(window);
```

Ukázka kódu 7.2: Vytvoření prázdného okna v GTK+

V ukázce 7.3 vytvoříme prázdné okno a nastavíme ho jako viditelné.

```
QWidget window;
window->show();
```

Ukázka kódu 7.3: Vytvoření prázdného okna v Qt

V následující ukázce 7.4 vytvoříme prázdné okno v Cocoa API. Na druhém řádku nastavujeme styly okna, např. okno s měnitelnou velikostí nebo s možností minimalizace.

```
NSRect frame = NSMakeRect(0, 0, 200, 200);

NSWindow* window = [[NSWindow alloc] initWithContentRect:frame
      styleMask:NSClosableWindowMask|NSResizableWindowMask
      backing:NSBackingStoreBuffered
      defer:NO];
```

Ukázka kódu 7.4: Vytvoření prázdného okna v Cocoa

## 7.2 Nastavení vlastností okna

V následující ukázce 7.5 nastavujeme další vlastnosti okna ve Windows API. Ve *WinMain* funkci vytvoříme strukturu třídy okna typu *WNDCLASSEX* [1]. Tato struktura obsahuje informace o vlastnostech okna, např. barvu pozadí, ikonu aplikace, typ kurzoru a další.

```
WNDCLASSEX wcex;
wcex.lpfnWndProc    = WndProc;
wcex.hInstance      = hInstance;
wcex.hCursor        = LoadCursor(NULL, IDC_ARROW);
wcex.hbrBackground  = CreateSolidBrush(RGB(255, 255, 255));
```

Ukázka kódu 7.5: Nastavení vlastností okna ve Windows API

V ukázce 7.6 nastavíme v GTK+ defaultní velikost otevřeného okna, umístění uprostřed obrazovky a název okna. Pro nastavení dalších vlastností okna využíváme funkce *gtk_window_** [2]

```
gtk_window_set_title(GtkWindow(window), "Hello world");
gtk_window_set_default_size(GtkWindow (window), 400, 400);
gtk_window_set_position(GTK_WINDOW(window), GTK_WIN_POS_CENTER);
```

Ukázka kódu 7.6: Nastavení vlastností okna v GTK+

[1] https://msdn.microsoft.com/cs-cz/library/windows/desktop/ms633576

[2] https://developer.gnome.org/gtk3/stable/GtkWindow.html

V následující ukázce 7.7 nastavíme název, velikost a umístění nově otevřeného okna, souřadnice x a y se musí spočítat podle velikosti a rozlišení monitoru.

```
window->setWindowTitle("Hello world");
window->resize(400, 400);
window->move(x, y);
```

Ukázka kódu 7.7: Nastavení vlastností okna v Qt

V poslední ukázce této sekce nastavíme název a pozadí nově otevřeného okna v Cocoa API, viz ukázka 7.8

```
[window setTitle:@"Hello World"];
[window setBackgroundColor:[NSColor blueColor]];
```

Ukázka kódu 7.8: Nastavení vlastností okna v Cocoa

## 7.3 Menu bar

V ukázce 7.9 vytváříme hlavní Menu bar v horní části okna, viz Obrázek 7.1. *CreateMenu* vytvoří prázdné menu, do kterého voláním funkce *AppendMenu* přidáváme položky, vždy na konec menu baru. Funkce *SetMenu* pak zajistí, že bude menu v okně zobrazené.


Obrázek 7.1: Menu Bar ve Windows API

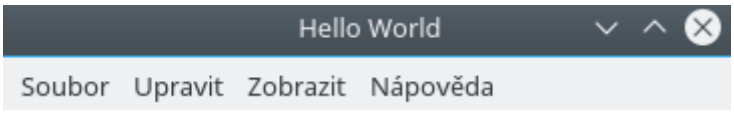
```
HMENU menu = CreateMenu();
        HMENU file = CreateMenu();
        HMENU edit = CreateMenu();

        AppendMenu(menu, MF_POPUP, (UINT_PTR)file, L"Soubor");
        AppendMenu(menu, MF_POPUP, (UINT_PTR)edit, L"Upravit");

        SetMenu(hwnd, menu);
```

Ukázka kódu 7.9: Vytvoření Menu baru ve Windows API

V ukázce 7.10 vytvoříme stejný Menu bar, tentokrát v GTK+, viz Obrázek 7.2 Nejdříve zaregistrujeme jednotlivé prvky typu GtkWidget a poté přiřadíme proměnným konkrétní typ, tedy celý menu bar nebo jeho položky. Poslední řádek zajistí připojení konkrétních položek do Menu baru.




Obrázek 7.2: Menu Bar v GTK+

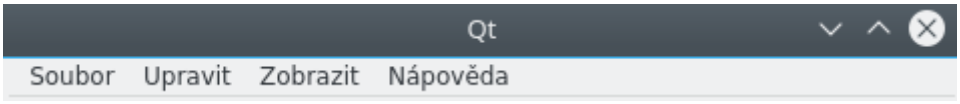
```
GtkWidget *menu;
GtkWidget *file;
GtkWidget *edit;

menu = gtk_menu_bar_new();
file = gtk_menu_item_new_with_label("Soubor");
edit = gtk_menu_item_new_with_label("Upravit");

gtk_menu_shell_append(GTK_MENU_SHELL(menu), file);
```

Ukázka kódu 7.10: Vytvoření Menu baru v GTK+

V následující ukázce 7.11 nejprve vytvoříme prázdný menu bar a jeho jednotlivé položky, které poté musíme menu baru přiřadit, viz Obrázek 7.3




Obrázek 7.3: Menu bar v Qt

```
QMenuBar *menu = new QMenuBar();
QMenu *file = new QMenu("Soubor");
QMenu *edit = new QMenu("Upravit");

menu->addMenu(file);
menu->addMenu(edit);
```

Ukázka kódu 7.11: Vytvoření Menu baru v Qt

## 7.4 Tlačítka

Ve Windows API je většina grafických prvků tvořena jako okno, tedy funkcí *CreateWindow* nebo *CreateWindowEx*, která navíc umožňuje nastavení více stylů.


Obrázek 7.4: Tlačítko ve Windows API

```
Button = CreateWindowEx(0, L"button", / třída button
      L"Tlačítko",                    / text na tlačítku
      WS_CHILD | WS_VISIBLE |         / styly tlačítka
      BS_PUSHBUTTON,                  / typ tlačítka
      20, 20,                         / souřadnice tlačítka
      100, 40,                        / šířka a výška
      hwnd,                           / handle rodičovského okna
     (HMENU)ID_BUTTON,                / identifikátor tlačítka
      NULL, NULL);                    / instance aplikace
```

Ukázka kódu 7.12: Vytvoření tlačítka ve Windows API

Podle parametru[3] ve stylech tlačítka určíme o jaký typ tlačítka se jedná, v této ukázce kódu je např. klasické tlačítko (viz Obrázek 7.4). Pokud bychom chtěli přepínač, parametr nastavíme jako *BS_AUTORADIOBUTTON* a pro zaškrtávací políčko parametr *BS_AUTOCHECKBOX*.

Funkcí *CheckDlgButton(hwnd, 2, BST_CHECKED)* můžeme nastavit přepínač nebo zaškrtávací políčko jako defaultně zaškrtnuté. Druhý parametr funkce je identifikátor tlačítka.
Ve funkci *WndProc* implementujeme switch a jeho case *WM_COMMAND* pro zpracování akcí komponent, kde tlačítku přiřadíme po jeho stisknutí akci uzavření okna, viz ukázka 7.13.

```
if (LOWORD(wParam) == ID_BUTTON) {
            PostQuitMessage(0);
        }
```

Ukázka kódu 7.13: Nastavení akce tlačítka

V následující ukázce 7.14 vytváříme tlačítko v GTK+, viz Obrázek 7.5. Každý typ tlačítka má vlastní funkci, přepínače vytvoříme voláním funkce *gtk_radio_button_new* a zaškrtávací políčka *gtk_check_button_new*. Při vytváření těchto typů tlačítek má funkce navíc jeden parametr, který určuje skupinu, kterou tlačítka sdílejí. Skupiny je nutné nastavit hlavně u přepínačů, které by jinak ztratily svůj význam (mohli bychom zaškrtnout více než jeden přepínač najednou). Poslední volání funkce zajistí uzavření okna po stisknutí tlačítka.


Obrázek 7.5: Tlačítko v GTK+

[3] https://msdn.microsoft.com/en-us/library/windows/desktop/bb775947(v=vs.85).aspx

```
GtkWidget *button;
button = gtk_button_new_with_label("Tlačítko");
gtk_widget_set_size_request(button, 100, 40);

g_signal_connect(G_OBJECT(button), "clicked",
                 G_CALLBACK(gtk_main_quit), G_OBJECT(window));
```

Ukázka kódu 7.14: Vytvoření tlačítka v GTK+

V ukázce 7.15 vytváříme tlačítko v Qt (viz Obrázek 7.6). Pokud chceme v Qt nastavit velikost tlačítka a zároveň jeho umístění v okně, použijeme funkci *setGeometry()*. Pro vytvoření jiného typu tlačítka zvolíme místo *QPushButton QRadioButton* pro přepínače a *QCheckBox* pro zaškrtávací políčka. Pokud chceme některé z tlačítek zaškrtnuté již při otevření okna, použijeme příkaz *button->setChecked(true)*.


Obrázek 7.6: Tlačítko v Qt

```
QPushButton *button = new QPushButton("Tlačítko");
button->resize(100, 40);
QObject::connect(button, SIGNAL(clicked()), &app, SLOT(quit()));
```

Ukázka kódu 7.15: Vytvoření tlačítka v Qt

V ukázce 7.16 je vytvoření tlačítka v Cocoa API. *NSRoundedBezelStyle* nastaví styl tlačítka se zaoblenými rohy. Pro vytvoření přepínačů slouží *NSRadioButton* vložený do *NSMatrix* pro vytvoření skupin tlačítek. Zaškrtávací políčka vytvoříme pomocí *NSSwitchButton*.

```
NSButton *button = [ [ NSButton alloc ] initWithFrame:
  NSMakeRect( 0, 0, 100, 40 ) ];
[ button setTitle: @"Tlačítko" ];
[ button setBezelStyle:NSRoundedBezelStyle];
```

Ukázka kódu 7.16: Vytvoření tlačítka v Cocoa

## 7.5 Ukazatel průběhu

První volání *SendMessage* v následující ukázce 7.17 nastaví minimální a maximální hodnoty (rozsah). Druhé volání nastaví počáteční zaplněnost (viz Obrázek 7.7) a stejným způsobem poté slouží k aktualizaci hodnoty (zaplněnosti) Progress baru.

Obrázek 7.7: Progress Bar ve Windows API

```
1  ProgressBar = CreateWindowEx(0, PROGRESS_CLASS,
2          NULL,
3          WS_CHILD | WS_VISIBLE,
4          20,  235,
5          180, 30,
6          hwnd, (HMENU)ID_PROGRESS,
7          NULL, NULL);
8
9  SendMessage(ProgressBar, PBM_SETRANGE, 0, MAKELPARAM(0, 100));
10 SendMessage(ProgressBar, PBM_SETPOS, 50, 0);
```

Ukázka kódu 7.17: Vytvoření Progress baru ve Windows API

V ukázce 7.18 vytvoříme Progress bar typu GtkWidget a voláním funkce *gtk_progress_bar_set_fraction* určíme počáteční zaplněnost Progress baru. Stejnou funkcí následně aktualizujeme tyto hodnoty. V této ukázce je počáteční hodnota nastavena na 60%, viz Obrázek 7.8.



Obrázek 7.8: Progress Bar v GTK+

```
1  GtkWidget *progressBar;
2  progressBar = gtk_progress_bar_new();
3  gtk_progress_bar_set_fraction(progressBar, 0.6);
```

Ukázka kódu 7.18: Vytvoření Progress baru v GTK+

V ukázce 7.19 vytvoříme instanci *QProgressBaru* a nastavíme umístění v okně. Základní rozsah je 100, tedy minimální a maximální hodnoty 0 a 100. Pokud chceme nastavit jiný rozsah, použijeme funkci *setRange*. Funkce *setValue* pak určí počáteční zaplněnost progress baru a následnou aktualizaci hodnot. Vzhled progress baru vidíme na obrázku 7.9.


Obrázek 7.9: Progress Bar v Qt

```
QProgressBar *progress = new QProgressBar();
progress->move(20, 235);
progressbar->setValue(50);
```

Ukázka kódu 7.19: Vytvoření Progress baru v Qt

V Cocoa opět vytvoříme Progress bar s minimální a maximální hodnotou a počáteční zaplněností, viz ukázka 7.20.

```
NSRect aFrame=NSMakeRect(0,0,200,30);
NSProgressIndicator *progressBar = [[ NSProgressIndicator alloc]
    initWithFrame:aFrame ];
[progressBar setMinValue:0.0];
[progressBar setMaxValue:100.0];
[progressBar doubleValue:50.0];
```

Ukázka kódu 7.20: Vytvoření Progress baru v Cocoa

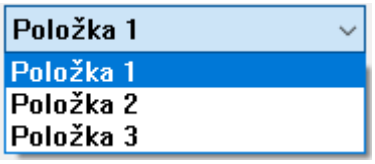
## 7.6 Rozbalovací seznam

Stejně jako u ostatních prvků ve Windows API vytvoříme rozbalovací seznam funkcí *CreateWindow*. Voláním funkce *SendMessage* přidáme do seznamu jednotlivé položky (viz ukázka 7.21 a Obrázek 7.10).

```
ComboList = CreateWindow(L"combobox",
      L"Rozbalovací seznam",
      WS_CHILD | WS_VISIBLE | CBS_DROPDOWNLIST,
       20, 200,
       180, 30,
       hwnd,
      (HMENU)ID_COMBO,
       NULL, NULL);

SendMessage(hwndCombo, CB_ADDSTRING, 0, (LPARAM) L"Položka 1");
```

Ukázka kódu 7.21: Vytvoření rozbalovacího seznamu ve Windows API




Obrázek 7.10: Rozbalovací seznam ve Windows API

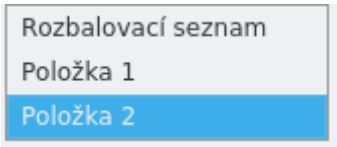
V ukázce 7.22 vytvoříme prázdný rozbalovací seznam v GTK+ a následným voláním funkce přidáme jednotlivé položky (viz Obrázek 7.11). Druhým parametrem funkce je identifikátor položky, pokud není NULL je využit jako ID řádku.


Obrázek 7.11: Rozbalovací seznam v GTK+

```
GtkWidget *combobox;
combobox = gtk_combo_box_text_new();
gtk_combo_box_text_append(combobox, NULL, "Položka 1");
```

Ukázka kódu 7.22: Vytvoření rozbalovacího seznamu v GTK+

V následující ukázce 7.23 vytvoříme v Qt prázdný rozbalovací seznam, do kterého poté přidáme jednotlivé položky (viz Obrázek 7.12).




Obrázek 7.12: Rozbalovací seznam v Qt

```
QComboBox *comboBox = new QComboBox();
comboBox->addItem("Položka 1");
```

Ukázka kódu 7.23: Vytvoření rozbalovacího seznamu v Qt

V Cocoa nejprve vytvoříme prázdný seznam, do kterého poté přidáváme jednotlivé položky, v ukázce 7.24 typu *String*.

```
NSComboBox* comboBox = [[NSComboBox alloc] initWithFrame:
                        NSMakeRect(0, 0, 150, 26)];
NSString *item = @"Položka 1";
[comboBox addItemWithObjectValue:item];
```

Ukázka kódu 7.24: Vytvoření rozbalovacího seznamu v Cocoa

## 7.7 Kalendář

Kalendář vytvoříme ve Windows API voláním funkce *CreateWindowW* s prvním parametrem určujícím prvek kalendáře, viz ukázka 7.25.

```
Calendar = CreateWindowW(MONTHCAL_CLASSW, L"",
    WS_BORDER | WS_CHILD | WS_VISIBLE | MCS_NOTODAYCIRCLE,
    260, 20, 200, 200,
    hwnd, (HMENU)ID_CALENDAR,
    NULL, NULL);
```

Ukázka kódu 7.25: Vytvoření kalendáře ve Windows API

V následující ukázce 7.26 je příklad vytvoření kalendáře v GTK+. GTK+ i Qt knihovny vytvoří kalendář se zobrazeným aktuálním rokem, měsícem a vybraným aktuálním datem.

```
GtkWidget *calendar;
calendar = gtk_calendar_new();
```

Ukázka kódu 7.26: Vytvoření kalendáře v GTK+

V následující ukázce 7.27 je příklad vytvoření kalendáře v Qt.

```
QCalendarWidget *calendar = new QCalendarWidget;
```

Ukázka kódu 7.27: Vytvoření kalendáře v Qt

V poslední ukázce 7.28 je vytvoření kalendáře v Cocoa API. Typ kalendáře nastavíme na gregoriánský a první den v týdnu na pondělí.

```
NSCalendar *calendar = [[NSCalendar alloc]
    initWithCalendarIdentifier:NSCalendarIdentifierGregorian];
calendar.firstWeekday = 2;
```

Ukázka kódu 7.28: Vytvoření kalendáře v Cocoa

## 7.8 Náročnost použití uvedených API

Poté co jsem vyzkoušela výše uvedené API je mohu zhodnotit podle jejich složitosti a časové náročnosti.

Osobně bych nejraději pro programování GUI zvolila knihovnu GTK+, která mi připadá ze všech API nejjednodušší s nejkratší syntaxí. Také její dokumentace je velmi přehledná a snadno se v ní vyhledává. Tvorba jednotlivých prvků v knihovnách GTK+ a Qt je velice podobná, což se nedá říct o přehlednosti dokumentace a jejích tutoriálů. Nejsložitější syntaxi má dle mého názoru Windows API a Cocoa API. Ve Windows API mají funkce mnoho parametrů, kde je často mnoho z nich nepotřebných a jsou tedy *NULL*. Pokud je potřeba nastavit více stylů vzhledu určité komponenty, délka funkce rychle roste a začíná být nepřehledná a zbytečně složitá. Pro programování Windows aplikací bych doporučila některý z frameworků, které pracují nad samotným API a nemají tak zdlouhavý zápis.

Pro tvorbu aplikací pro OS X v Cocoa jsou v mnoha tutoriálech a knihách o programování v tomto API doporučovány nástroje Xcode a Interface Builder, kde se jednotlivé komponenty naklikají a složí dohromady. Samotné psaní kódu pak spočívá spíše v nastavení a přiřazení akcí těmto komponentám. Obecně tedy není doporučováno psát celé aplikace ručně, ale využít tyto nástroje[3]. Jak je vidět v ukázkách kódu v předchozích sekcích, tvorba jednotlivých prvků má obdobně dlouhou syntaxi jako ve Windows API.

# 8 Závěr

Tématem práce byla analýza vývoje uživatelských rozhraní v historii. Jejím cílem bylo vytvořit přehled hlavních operačních systémů používaných v dnešní době na osobních počítačích a jejich postupného vývoje.

V první části práce jsem popsala druhy uživatelských rozhraní a rozdíly mezi nimi. Poté jsem vysvětlila některé pojmy související s grafickým uživatelským rozhraním a jeho možnou implementací v různých systémech.

V další části práce jsem se věnovala historii jednotlivých operačních systémů a sestavila jsem přehled jejich verzí v chronologickém pořadí. Konkrétně jsem se seznámila s jejich vývojem a popsala jejich uživatelské prostředí, jeho prvky a rozdíly oproti předešlým verzím či jiným systémům. Zabývala jsem se především systémy, které jsou obecně známé a využívané v dnešní době. Uvedla jsem ale i příklady prvních systémů, které obsahovaly grafické uživatelské rozhraní a určily tak směr následujícím systémům využívajících GUI. Pro zkoumání prostředí některých systémů jsem využila emulátory operačních systémů.

V následující části se věnuji jednotlivým komponentám rozhraní, kde uvádím jejich první výskyt a možnosti, které dávají uživatelům.

Cílem poslední části práce bylo porovnat programátorská rozhraní využívaná pro tvorbu uživatelských rozhraní na jednotlivých operačních systémech a zhodnotit je na základě jejich náročnosti při použití. Pro tento účel jsem vytvořila jednoduchý program s GUI pro aplikaci ve Windows API a knihovnách GTK+ a Qt. Poté jsem uvedla ukázkové příklady kódu pro tvorbu okna a jeho jednotlivých komponent jako jsou tlačítka, rozbalovací seznamy či kalendář. Toto srovnání by mělo poskytnout základní informace o syntaxi, struktuře programu a implementaci základních komponent ve zmíněných API a uvést rozdíly v implementaci a složitosti mezi nimi.

Práce by se dala rozšířit o popis dalších méně známých operačních systémů a jejich programátorského rozhraní.

# Literatura


[1] Cocoa API Reference. [online]. 2017 [cit. 2017-04-21].
URL https://developer.apple.com/reference/appkit/

[2] Graphical user interface gallery. [online]. 2006 [cit. 2017-04-25].
URL http://www.guidebookgallery.org/screenshots

[3] Programming Mac OS X with Cocoa for Beginners. [online]. 2013 [cit. 2017-04-21].
URL https://en.wikibooks.org/wiki/Programming_Mac_OS_X_with_Cocoa_for_Beginners/What_is_Cocoa%3F

[4] Windows API Control Library. [online]. 2017 [cit. 2017-04-20].
URL https://msdn.microsoft.com/en-us/library/windows/desktop/bb773169(v=vs.85).aspx

[5] The Xerox Alto computer. 1981, [online]. 2003 [cit. 2017-05-01].
URL
http://www.guidebookgallery.org/articles/thexeroxaltocomputer

[6] *Windows Programming*. 2016, [online]. 2016 [cit. 2017-04-20].
URL https://en.wikibooks.org/wiki/Windows_Programming

[7] kolektiv autorů: *Linux Dokumentační projekt*. 2003, ISBN 80-7226-114-2.

[8] van Dam, A.: Post-WIMP user interfaces. 1997, ISSN 0001-0782.
URL http://dl.acm.org/citation.cfm?id=253708

[9] Eric McCary, J. Z.: *Human Interface and the Management of Information. Information and Interaction Design*. 2013, ISBN 978-3-642-39209-2.
URL
http://link.springer.com/chapter/10.1007/978-3-642-39209-2_12

[10] Lineback, N.: The Graphical User Interface Gallery. 2017, [online]. [cit. 2017-05-01].
URL http://toastytech.com/guis

[11] Mark H. Chignell, J. A. W.: Wimps and nerds: an extended view of the user interface. 1991.

[12] Microsoft, C.: *Software Development Kit, Programmer's Reference*. 1986.

[13] Myers, B. A.: Graphical User Interface Programming. 2013.
URL https://www.researchgate.net/publication/2866174_Graphical_User_Interface_Programming

[14] Neil Matthew, R. S.: *Beginning Linux Programming 4th Edition*. 2008, ISBN 978-0-470-14762-7.

[15] O'Regan, G.: *Introduction to the History of Computing*. 2016, ISBN 978-3-319-33138-6.
URL http://link.springer.com/chapter/10.1007%2F978-3-319-33138-6_17

[16] Petzold, C.: *Programování ve Windows*. 1999, ISBN 80-7226-206-8.

[17] Reimer, J.: A History of the GUI. 2005, [online]. 2005 [cit. 2017-04-10].
URL http://arstechnica.com/features/2005/05/gui/

[18] Sullivan, K.: The Windows® 95 User Interface: A Case Study in Usability Engineering. 1996, [online]. 1996 [cit. 2017-04-15].
URL http://www.sigchi.org/chi96/proceedings/desbrief/Sullivan/kds_txt.htm