



SISTEME DE OPERARE

TEMĂ DE CASĂ

Structura sistemelor de operare Windows și Linux

GUI

Comparație - interfețe grafice cu utilizatorul

Mănucă Adrian, 431A
Țulucescu Alexandru, 431A
Bîrsan Ruben, 432A
Ciobanu Dragoș, 433A

CUPRINS

1. **Introducere. Evoluția interfețelor grafice(GUI).** ->Ciobanu Dragoș, 433A
 - Interfața liniei de comandă(CLI)
 - Interfața grafică Xerox PARC. WIMP.
2. **Interfața grafică la Linux.** ->Țulucescu Alexandru, 431A
 - X Window System
 - Introducere. Design. Principii.
 - Interfețe cu utilizatorul
 - Implementări
 - Compositing Window Manager. Compiz, Beryl.
 - Limitări și critici
 - XFCE
 - GNOME
 - Introducere. Tehnologii.
 - Aplicații și funcționare
 - KDE
 - Introducere. Evoluția în timp.
 - KDE 4
3. **Interfața grafică la Windows.** ->Mănuță Adrian, 431A
 - Începuturi
 - Windows API
 - Tema Luna(stilul Windows XP)
 - Windows Aero
 - Windows Presentation Foundation(WPF->.NET Framework 3)
 - Desktop Window Manager
 - Interfața Ribbon
 - Metro(limbaj de design)
4. **Comparație interfețe grafice.** ->Bârsan Ruben, 432A
 - Asemănări între Linux și Windows. Deosebiri.
 - Avantaje. Dezavantaje.
 - Concluzii.
5. **Bibliografie**



1.Introducere

Evoluția interfețelor grafice (GUI)

„**Interfața grafică** (în engleză: Graphical User Interface sau **GUI**) este numit sistemul de afișaj grafic-vizual pe un ecran, situat funcțional între utilizator și dispozitive electronice cum ar fi computere, dispozitive personale de tip hand-held (playere MP3, playere media portabile, dispozitive de jucat), aparate electrocasnice și unele echipamente de birou.” [1] Pentru a prezenta toate informațiile și acțiunile disponibile, un GUI oferă pictograme și indicatori vizuali, în contrast cu interfețele bazate pe text, care oferă doar nume de comenzi (care trebuie tastate) sau navigația text.

„Termenul de *GUI* este restricționat la ecrane cu display 2D cu rezoluții capabile de a descrie informația generică, în urma cercetării la PARC(Palo Alto Research Center). Termenul *GUI* este rareori aplicat altor tipuri de interfețe de rezoluție înaltă care sunt non-generice, precum jocurile video(unde termenul de HUD e preferat).”[2]

Cercetătorii de la Xerox au dezvoltat interfața **PARC** care consta din elemente precum ferestre, meniuri, butoane radio, casete de bifare și pictograme (icons). Din această interfață derivă majoritatea interfețelor grafice din sistemele de operare actuale precum Windows și Mac OS X

Un GUI folosește o combinație de tehnologii și dispozitive pentru a oferi o platformă cu care utilizatorul să interacționeze, pentru activitățile de culegere și producere de informații.

O serie de elemente conforme cu un limbaj vizual au evoluat pentru a reprezenta informația stocată pe un sistem de calcul. Aceasta ușurează activitățile oamenilor cu mai puține cunoștințe în domeniul calculatoarelor pentru a lucra cu un anumit soft. Cea mai întâlnită combinație a acestor elemente GUI este paradigma **WIMP**(Window, Icon, Menu, Pointing device), în special în rândul PC-urilor.

„Stilul de interacțiune WIMP utilizează un dispozitiv fizic pentru intrare/input pentru a controla poziția unui cursor și prezintă informația organizată în ferestre și reprezentată prin pictograme. Comenzile disponibile sunt compilate împreună în meniuri, și acțiunile sunt executate prin gesturi cu dispozitivul de indicare(pointing device/mouse). Un *manager de ferestre* facilitează interacțiunea dintre ferestre, aplicații și sistemul de ferestre. Sistemul de ferestre controlează dispozitive hardware precum dispozitive de indicare și hardware pentru grafică, la fel și poziționarea cursorului.” [3]

„În PC-uri aceste elemente sunt modelate într-o metaforă de birou, pentru a produce o simulare numită spațiu de lucru în care display-ul reprezintă un birou, pe care documentele și dosarele documentelor sunt așezate. Managerele de ferestre și alte softuri se combină pentru a simula mediul de lucru la birou cu anumite devieri de realism.”[4]

Post-WIMP include managerele de compoziție a ferestrelor în 3D, printre care sunt cunoscute **Compiz** și **Desktop Window Manager**.

GUI a fost introdus pentru a diminua utilizarea **CLI(Command Line Interface)**. Majoritatea sistemelor de operare încă mai păstrează și o interfață de tip CLI pentru anumite activități sau setări avansate.

CLI reprezintă un mecanism de interacțiune cu un computer, un sistem de operare sau un software prin introducerea de comenzi secvențial, linie cu linie. Aceasta interfață care este bazată doar pe introducerea de text contrastează cu modalitățile de interacțiune tip graphical user interface (GUI), care folosesc dispozitive de tip mouse pentru a selecta opțiuni sau meniuri.

Introducerea de comenzi prin intermediul CLI se face astfel: sistemul așteaptă ca utilizatorul să introducă o linie de comandă și să o confirme prin apăsarea tastei "Enter" de pe tastatură. Programul "Command Line Interpreter" primește, analizează (parsează) și execută comanda cerută. După executare programul prezintă utilizatorului rezultatul obținut, tot sub formă de linii de text.

Interpretatorul liniei de comandă poate rula într-un terminal text sau într-un emulator de terminal al unui client precum *PuTTY*.

Cele mai cunoscute și utilizate interpretoare CLI sunt **Command Prompt**-ul și **PowerShell** pe sistemele Windows și **Bash** pe sistemele Linux(Unix). Acestea utilizează mecanisme similare, PowerShell-ul din Windows apropiindu-se tot mai mult de CLI-urile Linux.

Interfața grafică la Linux



2. Interfața grafică la Linux

X Window System





1. Introducere

X Window System (de obicei numit X11 sau X) este un protocol de afisare care furnizeaza un sistem de ferestre pe ecrane raster. X furnizeaza un set de unelte standard si un protocol pentru construirea de interfete grafice pe majoritatea sistemelor de operare Unix-like si OpenVMS, si a fost portat pe toate sistemele de operare generaliste contemporane. [2]

X ofera o platforma de baza pentru construirea mediilor grafice: desenarea si miscarea ferestrelor si interactiunea cu mausul si tastatura, dar nu impune interfata cu utilizatorul, programele dient o ofera fiecare in parte. Din acest motiv stilul vizual al mediilor bazate pe X variaza foarte mult. X nu este o parte integranta sistemului de operare, ci este construit ca o interfata peste nucleul sistemului de operare. [2]

X a fost publicat prima data de MIT in 1984. Versiunea de protocol curenta, X11, a aparut in septembrie 1987. [2]

2. Design

X foloseste un model tip client-server: un server X comunica cu diverse programe tip client. Serverul accepta cereri pentru iesirea grafica (ferestre) si trimite inapoi intrari de la utilizator (de la tastatura, mouse sau touchscreen). Serverul poate functiona ca: [1]

- o aplicatie care afiseaza intr-o fereastră a unui alt sistem
- un program de sistem care controleaza iesirea video a unui PC
- o componenta hardware dedicata.

Aceasta terminologie client-server —terminalul utilizatorului ca "server", aplicatii locale sau distante drept "clients" — adeseori ii deruteaza pe noii utilizatori X, din cauza ca aceste termene apar inversate. Dar X priveste din perspectiva programului si nu din a utilizatorului sau a hardware-ului: displayul local X asigura servicii de afisare programelor, deci functioneaza ca un server; orice program distant foloseste aceste servicii, deci se comporta ca un client. MIT in 1984. Versiunea de protocol curenta, X11, a aparut in septembrie 1987. [1]

Protocolul de comunicatia dintre server si client opereaza network-transparent: clientul si serverul pot rula pe aceeasi masina sau pe masini diferite, posibil cu diferite arhitecturi si sisteme de operare, dar

Interfața grafică la Linux



funcționează la fel în ambele cazuri. Un client și un server pot comunica prin Internet printr-un tunel, conexiunea fiind criptată. [1]

Un client X poate conține un server X având displayul mai multor clienți. Aceasta este cunoscută drept "X nesting". Clienții open-source precum Xnest and Xephyr suportă X nesting. [1]

Pentru a porni un program client distant cu afișaj pe serverul local, utilizatorul va deschide în mod obișnuit un terminal window și telnet sau ssh pe clientul distant și va seta displayul. (e.g. export DISPLAY=[user's machine]:0 on a remote machine running bash). Aplicația client sau shell-ul se va conecta la serverul local, trimițând displayul sau sesiunea input către serverul local. În alt mod, mașina locală poate rula un mic program ajutător pentru a se conecta la mașina distantă și a porni aplicația client dorită acolo. [1]

Exemple practice de folosire a clienților distanți: [1]

- administrarea unei mașini distante în mod grafic
- rularea unei aplicații de simulare intensiva pe o mașină UNIX distantă și afișarea rezultatelor pe un desktop local Windows
- rularea unui software grafic pe mai multe mașini la un moment dat, controlate de un singur display, tastatură și mouse.

3. Principii

În 1984, Bob Scheifler și Jim Gettys au stabilit principiile timpurii ale X:

- "Nu implementa o nouă funcționalitate decât dacă un dezvoltator nu poate scrie un program real fără ea."
- "Singurul lucru mai rău decât generalizarea dintr-un singur exemplu este generalizarea fără nici un exemplu."
- "Dacă o problemă nu este complet înțeleasă, probabil că este mai bine să nu oferi nici o soluție."
- "Dacă poți obține 90% din efect făcând 10% din munca, folosește soluția mai simplă".

Interfața grafică la Linux



X a pastrat in cea mai mare dintre principii. Implementarea de referinta ramane aproape in intregime compatibila cu protocolul din 1987. [5]

4. Interfete cu utilizatorul

In mod intentionat, X nu specifica elemente grafice pentru aplicatie tip button, menu, ferestre title bar si asa mai departe. in schimb software-ul utilizatorului – precum window manager, GUI widget toolkit and desktop environment, sau specific aplicatiei graphical user interfaces – definesc toate aceste detalii. Drept urmare, nu este posibila o interfata X "tipica" deoarece sint preferate mai multe tipuri de interfete. [1]

Un window manager controleaza afisajul si aparenta ferestrelor aplicatiilor. Aceasta poate fi o interfata asemanatoare Microsoft Windows sau Macintosh (exemplele includ Metacity in GNOME, KWin in KDE sau Xfwm in Xfce) sau prezinta controale radical diferite (precum tiling window manager). Managerul de ferestre poate fi simplist (e.g. twm, managerul de ferestre livrat standard in X, sau evilwm, un window manager minimalist) sau sa ofere functionalitate unor sisteme desktop complete (e.g. Enlightenment). [6]

Multi utilizatori folosesc X cu un desktop environment complet, care include un manager de ferestre, diverse aplicatii si o interfata consistenta. GNOME, KDE si Xfce sint cele mai populare medii desktop. Mediul Unix standard este Common Desktop Environment (CDE). Initiativa freedesktop.org se ocupa de problemele de interoperabilitate dintre desktop si componentele necesare pentru a realiza un mediu X competitiv. [1]

X fiind responsabil pentru interactiunea intre diverse medii grafice , diverse keyboard shortcuts au devenit asociate cu X. Control-Alt-Backspace termina sesiuna X in mod obisnuit , in timp ce Control-Alt impreuna cu tastele F1-F12 comuta sesiunea de lucru catre terminalele virtuale UNIX. Aceasta este insa un detaliu de implementare lasat la latitudinea serverului X si nu este nici de departe universal; de exemplu , implementarea serverului X pentru Windows si Macintosh nu suporta aceste combinatii de taste. [1]

5. Implementari

Implementarea de referinta X.Org serveste drept implementare canonica. Datorita licenta software licentei nerestrictive, exista multe variatii, atat libere cat si proprietare. [1]

Pana in 2004, XFree86 a fost cea mai folosita varianta libera pentru sistemele Unix-like, si standardul de facto pentru dezvoltarea pe X. [1]

Un terminal X este un client care ruleaza pe un server X. Aceasta arhitectura a devenit foarte populara in vederea autentificarii simultane a mai multor utilizatori pe acelasi server. Aceasta tendinta este

Interfața grafică la Linux



conforma cu dorinta originala a autorilor proiectului de la MIT. [1]

Terminalele X exploreaza reseaua locala folosind X Display Manager Control Protocol pentru a genera o lista de masini disponibile unde sa poate rula programe client.

Hardware dedicat pentru terminale X devine din ce in ce mai rar ; un PC sau un client avind un X server ,dispun de aceeasi functionalitate la un pret mai redus. [1]

6. Limitari si critici

UNIX-HATERS Handbook (1994) a dedicat un intreg capitol, problemelor cu care se confrunta X. Why X Is Not Our Ideal Window System (1990) de Gajewska, Manasse si McCormack a detaliat problemele din protocol , impreuna cu recomandari pentru imbunatatiri. [3]

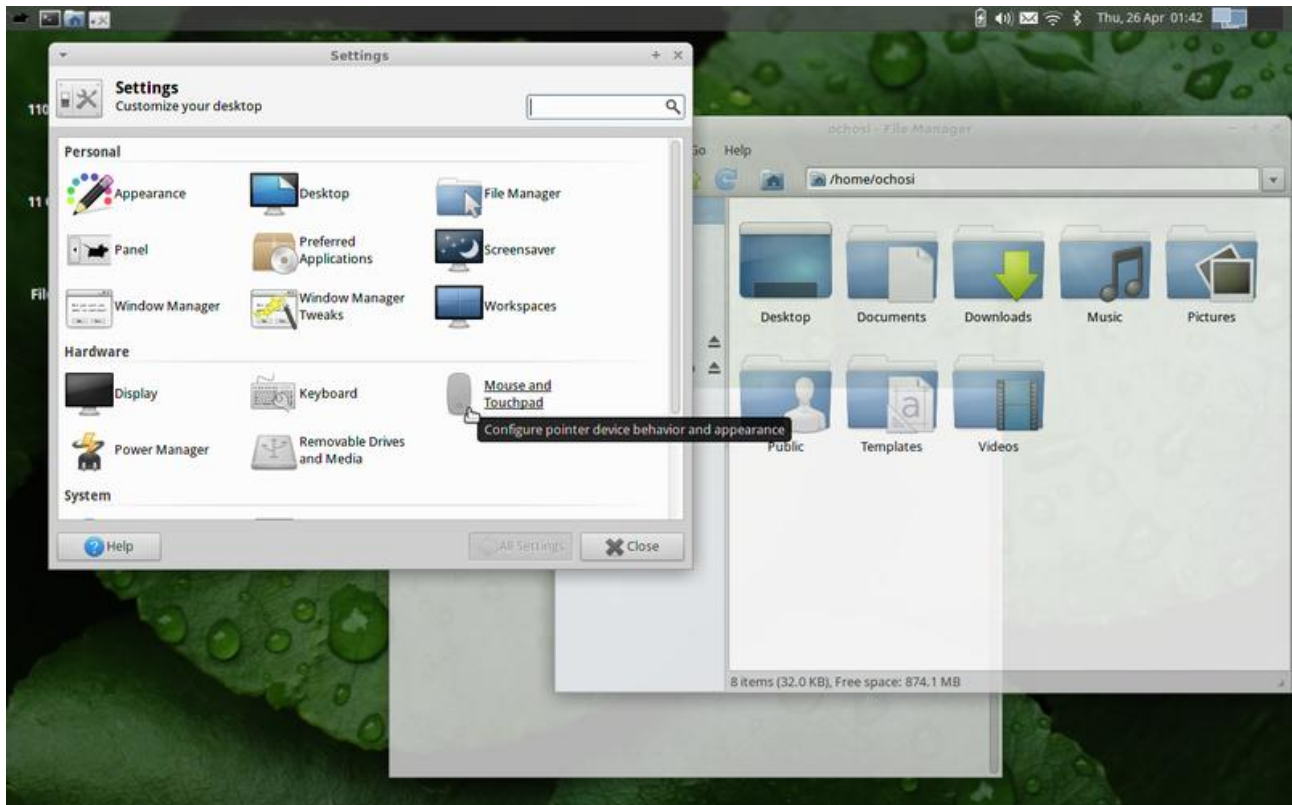
in mod intentionat X nu contine nici o specificatie referitoare la interfata utilizator si nici la comunicatiile inter-aplicatii.Din cauza aceasta s-au dezvoltat interfete foarte diferite , aplicatiile nefunctionind bine impreuna in 100% din cazuri. ICCCM, o specificatie pentru interoperabilitatea programelor client are un renume de a fi greu de implementat in mod corect. Alte standarde precum Motif and CDE nu au remediat problemele. Aceasta a fost o sursa permanenta de frustrare pentru atat pentru utilizatori cit si pentru programatori mult timp.[2] Pentru a evita problemele legate de programarea grafica,programatorii folosesc un mediu grafic sau un toolkit specific pentru a evita sa lucreze direct cu ICCCM. [4]

O cerinta de constructie X este ca serverul si clientii sa opereze in mod separat. Aceasta cerinta impreuna cu independenta de platforma hardware face ca X sa nu fie atat de rapid precum sistemele ale caror parte grafica este integrata in sistemul de operare , precum Microsoft Windows sau Mac OS. Recomandarea pentru X este de 4 pina la 8 MB de RAM pentru o performanta rezonabila. [1]

Versiunile curente de Windows sau Mac OS X Quartz au o separare a sistemelor interne similara cu client/server din X .Cea mai mare lantenta vine de la partea de retea round-trip delay time dintre client si server si nu de la protocolul insusi).Implementarile moderne X folosesc sockets locali si shared memory , diminuind astfel latentele. [1]



Xfce



Xfce este un mediu desktop liber și open-source (sub licența GPL) pentru sisteme de operare compatibile UNIX. Astfel, el rulează pe platforme Linux, BSD, Solaris, AIX etc. Xfce se bazează, ca și GNOME, pe toolkitul GTK+ pentru generarea desktopului și meniurilor, și pe Xfwm ca manager de ferestre (deși, pentru desenarea ferestrelor, se poate folosi orice manager de ferestre compatibil). A fost inițial gândit pentru a fi similar cu mediul desktop comercial CDE, și se folosea de toolkitul Xforms (de aici și numele), dar s-a îndepărtat radical de modelul CDE odată cu versiunile ulterioare (Xfce nu este primul mediu desktop care s-a vrut inspirat de CDE, KDE a pornit de la același punct).

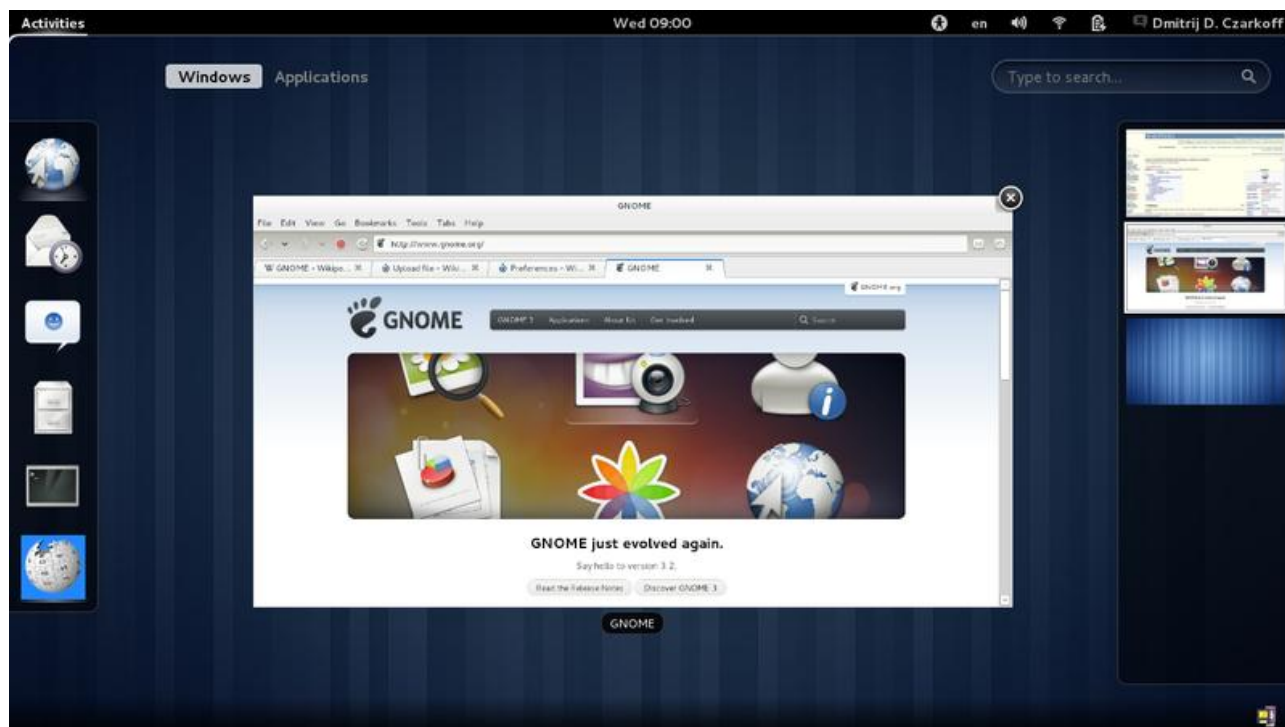
Xfce este renumit pentru simplitate și consum foarte redus de resurse în condițiile în care oferă configurabilitate completă din meniuri, de aceea, el este disponibil pe distribuții orientate spre calculatoare mai vechi (Vector Linux), sau spre utilizatori dornici de performanță (Xubuntu) sau simplitate și minimalism (Zenwalk).

Xfce conține foarte multe îmbunătățiri orientate spre desktop - printre care suport complet pentru transparente și umbre, și un sistem de compoziție a imaginilor similar cu cel disponibil în Beryl/Compiz (pentru GNOME sau KDE) și în Windows Vista. Inclusiv în Xfce este și un manager de fișiere numit Thunar, foarte asemănător cu Nautilus (managerul de fișiere al GNOME). Versiunile mai vechi de 4.4 nu suportau pictograme pe desktop - ultima versiune însă a adăugat și acest suport.

Interfața grafică la Linux



Gnome



1. Introducere

GNOME este un mediu desktop gratuit pentru sisteme compatibile UNIX. Este software (open source), sub licența liberă (GPL) (anumite biblioteci și toolkitul GTK+ sunt sub licența LGPL). Este mediul standard de lucru pe un număr mare de distribuții Linux (Ubuntu, Fedora, Red Hat etc.) și pe sistemele SUN Solaris. În plus, poate fi instalat pe aproape toate distribuțiile Linux, pe sisteme BSD, etc. [10]

GNOME a apărut în 1997, autorul inițial fiind Miguel de Icaza. Proiectul a fost început ca urmare a utilizării de către KDE a toolkit-ului Qt, care la vremea respectivă nu era open source - ducând la limitări asupra a ceea ce putea fi programat pentru KDE, și la posibilitatea ca, în cazul în care Qt își schimba strategia, KDE să nu mai poată fi folosit legal. Acest lucru nu mai este o problemă, Qt este software liber astăzi, dar atât Gnome cât și KDE continuă să se dezvolte, fiecare proiect având obiective diferite. KDE și GNOME colaborează în cadrul proiectului Free Desktop, pentru a asigura standardizarea în sisteme Unix/Linux. [8][9]

2. Tehnologii



Gnome se bazează pe limbajul de programare C și pe toolkit-ul GTK+, toolkit care originează din proiectul GIMP, și a fost extins de către GNOME. Acest toolkit este portat și pe alte sisteme (Windows, MacOS etc.), permitând aplicațiilor bazate pe acest toolkit să ruleze nativ și pe aceste sisteme. Pe lângă GTK+, GNOME a mai dezvoltat un număr de tehnologii, precum GConf, un sistem integrat de configurare pentru sistem, LibXML, NetworkManager și Orbit. În plus, GNOME a mai adoptat și un număr de tehnologii dezvoltate în parteneriat cu ceilalți parteneri FreeDesktop, precum Cairo (un sistem de prelucrare a graficii vectoriale), DBUS (un sistem de comunicare între procese), HAL (o bibliotecă de abstractizare a sistem call-urilor), etc. [7]

3. Aplicații și funcționare

Gnome se bazează pe un ghid de design foarte strict, care asigură ca aplicațiile arată și funcționează la fel. Tinta Gnome este un sistem simplu - care să permită maximum de configurare și funcționalitate cu minimum de efort. De aceea, multe opțiuni care sunt puțin folosite, au fost scoase, sau comprimate - asigurând un sistem curat și puțin încărcat, dar în același timp puternic. Aceeași mentalitate a dus, spre exemplu, la un sistem de "salvare pe măsura ce configurezi", orice modificare aplicându-se imediat, fără ca utilizatorul să apese vreun buton de tip Ok sau Aplica (de obicei există un buton "înapoi" dacă utilizatorul nu dorește să pastreze modificările). Aceeași mentalitate de sistem ușor configurabil a dus la includerea a foarte puține programe în nucleul sistemului - dar și la existența a foarte multe proiecte independente care se bazează pe GNOME, și care sunt incluse în distribuțiile de Linux alături de sistem. [7]

Astfel, în GNOME se găsește Nautilus (un navigator de fișiere și sistem), Metacity (un manager de ferestre - Gnome poate folosi orice alt manager de ferestre), Ekiga (un sistem de telefonie VOIP), Epiphany (un browser web, care însă e înlocuit în multe distribuții de Firefox, care se bazează și el pe GTK+, gedit (un editor text). [7]

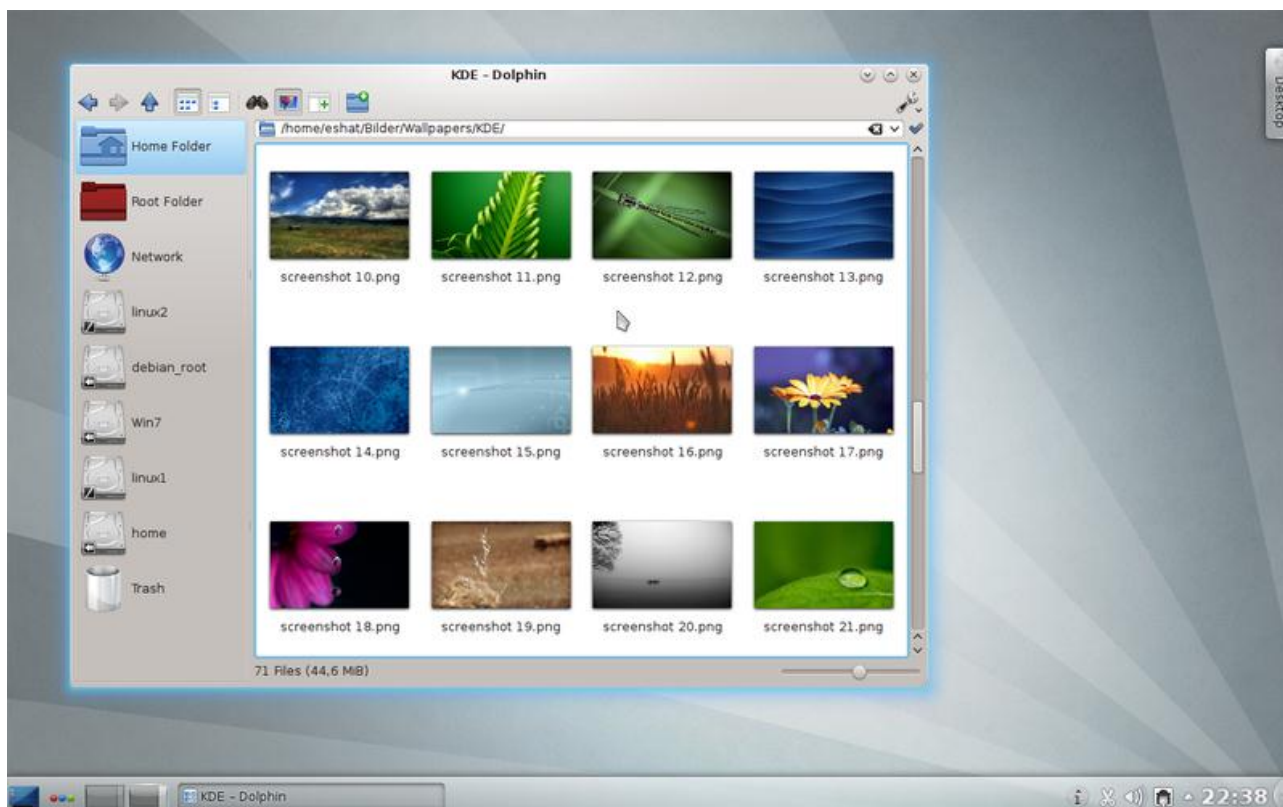
Pe tehnologii GNOME, se bazează software precum Pidgin (fostul GAIM - un sistem de mesagerie instant multi-protocol), FileRoller (un arhivator), Banshee (un player de muzică), F-Spot (un organizator de imagini), Inkscape (un editor de grafică vectorială), GIMP (un editor de imagini) etc. [7]

Gnome este un sistem extrem de configurabil, totul, de la poziționarea, conținutul și aspectul meniurilor și dialogurilor, până la forma desktopului se poate configura. Întreaga interfață este configurabilă prin utilizarea de teme, care se instalează prin drag-and-drop în managerul de teme furnizat de GNOME. Una din trăsăturile proiectului este tendința spre accesibilitate și internaționalizare, oferind suport complet (meniuri, aplicații, spell-checker) pentru limbi multiple (inclusiv limba română) și pentru tastaturi (inclusiv o tastatură corectă pentru limba română, cu trei moduri de input, la preferința utilizatorului). Alta trăsătură fundamentală e tendința spre accesibilitate - oferind un cititor de ecran, o "lupa", posibilitatea de control vocal (cu software suplimentar), configuratoare specifice de tastatură, o metodă de input bazată doar pe mouse și un toolkit pentru scrierea de aplicații noi în acest domeniu. [7]

Interfața grafică la Linux



KDE



1. Introducere

KDE (K Desktop Environment) este un Mediu Desktop gratuit pentru sisteme compatibile Unix. tînta proiectului este de a oferi o interfața completă, ușor de utilizat, și ușor de adaptat nevoilor personale ale fiecărui utilizator. KDE este mediul desktop standard pe un număr de distribuții: Mandriva, Kubuntu, SUSE, Knoppix, Slackware, este una din opțiuni pe multe alte distribuții (Debian, Gentoo), sau poate fi instalat ca mediu secundar (pe Ubuntu, Red Hat, Fedora, etc.). [10]

KDE rulează pe Linux, BSD (FreeBSD și DragonFly BSD îl folosesc ca mediu desktop standard), Solaris, HP-UX, AIX și pe alte sisteme Unix și compatibile. KDE rulează și pe MacOS - versiunea 3.5 prin Fink și X11, și este în curs de portare pe Windows. [11]

Ultima versiune KDE este 4.0. KDE și GNOME participă la elaborarea standardului FreeDesktop. [10]



2. Istoric

KDE a fost primul mediu desktop integrat disponibil pentru Linux. Dezvoltat începând cu 1996 de o echipă coordonată de Matthias Etrich, sistemul este bazat pe tool-kit-ul Qt, oferit de firma Trolltech, și a fost scris folosind limbajul C++. Paradigma principală a fost oferirea unui mediu complet desktop - interfața de bază, un set de programe, și unelte de configurare pentru acele programe și pentru sistem - toate arătând și funcționând la fel și fiind distribuite împreună (utilizatorul este liber să aleagă dacă dorește sau nu instalarea anumitor componente - dar programele sunt gândite în așa fel încât să poată colabora). Acest mod de gândire, la vremea aceea unic pentru Linux, a dus la popularizarea acestui mediu desktop încă de la prima versiune (1.0 în 1997). [10]

Cu toate acestea, au existat probleme și controverse de ordin legal - la vremea lansării KDE, Qt era software proprietar - existând astfel oricând posibilitatea ca o schimbare de viziune a firmei care comercializa toolkitul să ducă la distrugerea proiectului prin taxe de licență. Acest lucru a dus la dezvoltarea unui mediu desktop concurent, numit Gnome. Toate problemele de acest gen au dispărut, în momentul în care Trolltech a lansat Qt pentru Linux sub licența liberă (initial QPL, astăzi GPL). În plus, lansarea Qt 4, pe care se bazează KDE 4, sub licența liberă și pe alte platforme (Macintosh și Windows) duce la posibilitatea ca KDE și aplicațiile bazate pe KDE să fie disponibile și pe alte sisteme. [10]

3. KDE 4

KDE 4 este seria curentă de versiuni KDE. Prima versiunea (4.0) a fost publicată pe 11 ianuarie 2008.

Seria nouă conține părți înnoite ale mai multor componente de bază ale KDE, cum ar fi portarea la Qt 4. KDE 4 conține un nou API multimedia, numit Phonon, un sistem pentru integrarea dispozitivelor hardware, numit Solid, un nou stil și un set de pictograme numit Oxygen, un nou desktop și sistem de panouri numit Plasma. [10]

Se așteaptă ca aplicațiile KDE să fie mai rapide și să folosească mai puțină memorie, datorită portării la Qt 4 și datorită optimizării bibliotecilor de bază ale KDE. De asemenea, portarea la Qt 4 va facilita portarea KDE pe alte sisteme de operare, versiunea 4 a Qt având licența GPL și pentru Windows și Mac OS X. [10]

Vizual, cea mai notabilă schimbare sunt noile pictograme, teme și sunete oferite de proiectul Oxygen. Noul stil implicit este mai foto-realistic, spre deosebire de stilul implicit al versiunilor anterioare, care avea un aspect cartonat. [10]

Versiunea curentă este 4.3. Se preconizează ca versiunea 4.4 să apară în februarie 2009. [1]

3. Interfața grafică la



Începuturi

Compania *Microsoft* a introdus **Windows** pe piață pentru prima dată în noiembrie 1985, ca un supliment (add-on) la **MS-DOS**, deoarece interfețele grafice erau din ce în ce mai apreciate.

Versiunile vechi de Windows erau considerate doar îmbunătățiri sau variante grafice pentru sistemul de operare MS-DOS deoarece rula prin intermediul acestuia. Dar spre deosebire de MS-DOS, Windows permite utilizatorilor să execute simultan aplicații complexe, prin utilizarea așa-numitului „multitasking cooperativ”. Un factor nou este utilizarea unei interfețe grafice cu utilizatorul de tip GUI.

Windows la versiunea **1.0** nu permitea suprapunerea ferestrelor, care ocupau o anumită porțiune de ecran. Doar casetele de dialog puteau suprapune ferestrele. Acest lucru a fost rezolvat în versiunea **2.0**, iar, ulterior, la versiunile **3.0**, **3.1** interfața grafică a mai cunoscut diverse îmbunătățiri, precum înlocuirea programului shell *MS-DOS Executive* (din versiunile 1.0 și 2.0) cu *Program Manager*. La versiunea **3.11** nu a mai fost nevoie de MS-DOS pentru management-ul fișierelor, însă acesta era încă necesar pentru rularea Windows. Până la apariția Windows 95, aceste versiuni erau pornite manual de utilizator tastând în linia de comandă „win” (de la fișierul *win.com*) sau puteau să modifice *autoexec.bat* pentru a porni Windows-ul imediat după ce pornea MS-DOS.

Windows 95 a reprezentat un progres major pentru interfața grafică din Windows, folosind un concept de interfață grafică ce a fost îmbunătățit până la versiunea curentă de Windows din ziua de astăzi. Printre cele mai importante noutăți aduse de Windows 95 au fost *meniul Start* (Start Menu) și *bara de activități* (taskbar). Program Manager a fost înlocuit de *Windows Explorer*. Windows 95 putea fi rulat/pornit fără ajutorul MS-DOS. Până la **Windows XP**, **Windows NT**, **98**, **Me** și **2000** n-au mai adus modificări majore.

Windows Shell este principala componentă GUI în Microsoft Windows. Windows Shell include binecunoscutele componente Windows precum *taskbarul* și *meniul Start*, *Windows Explorer*, *Desktop-ul* și *pictogramele* (icons). Windows Shell nu este același lucru cu „Command-line Shell”, dar cele 2 concepte sunt asociate.

Windows USER este o componentă a sistemului de operare Windows care furnizează funcționalități de bază pentru a construi interfețe simple cu utilizatorul. Această componentă a existat în toate versiunile de Windows și include funcționalități pentru managementul ferestrelor, transmiterea mesajelor, procesarea datelor de intrare introduse de utilizator și controale standard.

Concret, Windows USER este responsabil pentru: *afișarea ferestrelor*, *suprapunerea ferestrelor*, *redimensionarea ferestrelor*, oferă elemente standard din ferestre precum *bara de titlu*, *butoanele de închidere*, *restaurare* și *minimizare*, *bara de meniu*, *imaginea de fundal* a Desktop-ului, elemente pentru procesarea datelor de intrare de la tastatură sau mouse.

Începând cu **Windows Vista**, *managementul pentru afișarea ferestrelor* s-a mutat de la Windows USER la **Desktop Window Manager**. La **Windows 7** bara de meniu și meniurile de context sunt înlocuite de interfața **Ribbon**, furnizată de componenta *RibbonUI*. [3]

„Windows USER implementează abstractizări pentru a dezvolta aplicații Windows. Acestea includ:

- *HWND (Handle to Window)*, unitatea de bază pentru afișaj și input pe platforma de dezvoltare Windows. Toate elementele vizuale au propriile lor HWND sau sunt părți dintr-un HWND mai mare. Multe funcții din Windows USER manipulează proprietățile HWND precum dimensiunea, poziția și titlul unei ferestre. Fiecare handler HWND are de asemenea un canal de comunicație (WNDPROC – Window Procedure) pentru livrarea mesajelor.

Ferestrele sunt gestionate ca un arbore, cu fereastra Desktop la rădăcina (root) ierarhiei. Ferestrele de tip child (copii) sunt de obicei conținute de cele de tip părinte.

- *MSG (Message)*. Aplicațiile și sistemul de operare comunică cu handleri HWND folosind mesaje. O fereastră mesaj este combinația unei ținte HWND, un cod de mesaj, și alte detalii precum poziția cursorului.
- *Bucă de mesaje (Message Loop)*. Combinând apeluri către Windows USER pe calea potrivită, o aplicație Windows procesează mesaje. Bucă mesajelor de bază din Windows (inclusiv apelurile GetMessage, TranslateMessage, DispatchMessage) este nucleul oricărei aplicații Windows.

Dezvoltarea de funcționalități asociate Windows USER se găsește în fișierul C header *winuser.h*

În versiunile Windows pe 16 biți, Windows USER a fost implementat ca un fișier numit *user.exe*, deși extensia fișierului era improprie, fiind de fapt fișier *DLL(Dynamic Link Library)*.

În versiunile Windows pe 32 biți, versiunea pe 32 de biți a Windows USER se numește *user32.dll*, în timp ce compatibilitatea cu aplicațiile pe 16 biți prin prezența unei versiuni modificate de *user.exe*.

În versiunile Windows pe 64 biți, implementarea pe 64 de biți a Windows USER e numită *user32.dll*, în timp ce versiune modificată pe 32 biți(de asemenea numită tot *user32.dll*) e prezentă în directorul *SysWOW64*.” [1][2]

Windows API

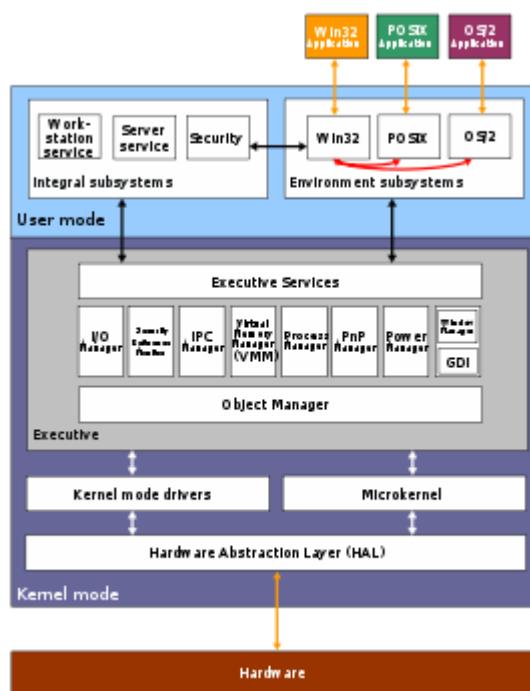
Este denumit și **WinAPI**, acronim de la *Windows Application Programming Interface*; înainte era cunoscut formal ca **Win32 API**, însă numele Windows API reflectă mai bine rădăcinile din versiunile Windows pe 16 biți și compatibilitatea cu versiunile pe 64 biți.

Este setul de API-uri bază, de la Microsoft, disponibil în sistemele de operare Windows. Aproape toate programele Windows interacționează Windows API; pe sistemele Windows NT un număr mic de aplicații folosesc *NativeAPI*(predominant utilizat la boot-ul sistemului).[1]

Suportul pentru dezvoltatori este disponibil în *Microsoft Windows SDK*, care oferă documentația și uneltele necesare pentru a dezvolta software bazat pe Windows API și alte interfețe Windows asociate.

Funcționalitatea oferită de Windows API poate fi grupată în 8 categorii:

- **Base Services.** Oferă acces la resursele fundamentale disponibile pe un sistem Windows. Includ aici sunt lucrurile precum sisteme de fișiere, dispozitive, procese și fire de execuție(thread-uri), și handler de erori. Aceste funcții se află în fișierele *kernel.exe*, *krnl286.exe* sau *krnl386.exe* pe versiunile Windows pe 16 biți, și *kernel32.dll* la Windows pe 32 biți.[2]
- **Advanced Services.** Oferă acces la funcționalități adiționale kernel-ului. Includ aici Windows Registry, stingerea/repornirea sistemului(sau revocarea), pornirea/oprirea/crearea unui serviciu Windows, gestionarea conturilor de utilizator. Aceste funcții se află în *advapi32.dll* pe o versiune Windows pe 32 biți.[3]
- **Graphics Device Interface.** Oferă funcționalități pentru ieșirea conținutului grafic pe monitoare, imprimante și alte dispozitive de ieșire. Se află în *gdi.exe* pe versiunile Windows pe 16 biți, și *gdi32.dll* pe versiunile Windows pe 32 de biți în user-mode. Suportul GDI Kernel-mode este oferit de *win32k.sys* care comunică direct cu driverul video/grafic.[4][5]
- **User Interface.** Oferă funcționalități pentru a crea și gestiona ferestre și controale de bază, precum butoane, scrollbars(bări pentru derularea paginii), primește input/date de intrare de la mouse și tastatură, și alte funcționalități asociate cu partea GUI din Windows. Această funcționalitate se află în *user.exe* pe versiunile pe 16 biți și în *user32.dll* pe versiunile pe 32 biți. De la Windows XP, controalele de bază se află în *comctl32.dll*, împreună cu controalele comune(Common Control Library).[6]
- **Common Dialog Box Library.** Oferă aplicațiilor casetele de dialog standard pentru deschiderea și salvarea fișierelor, alegerea fonturilor și culorilor, etc. Biblioteca se află într-un fișier numit *commdlg.dll* pe versiunile Windows pe 16 biți, și *comdlg32.dll* pe versiunile pe 32 biți. Este grupată sub categoria *User Interface* a API.[7]
- **Common Control Library** permite aplicațiilor acces la niște controale avansate oferite de sistemul de operare. Acestea includ elemente precum bări de stare, bări de progres, bări de unelte și tab-uri. Biblioteca se găsește într-un fișier DLL numit *commctrl.dll* pe versiunile Windows pe 16 biți, și *comctl32.dll* pe versiunile Windows pe 32 biți. Este grupată sub categoria *User Interface* a API.[8]

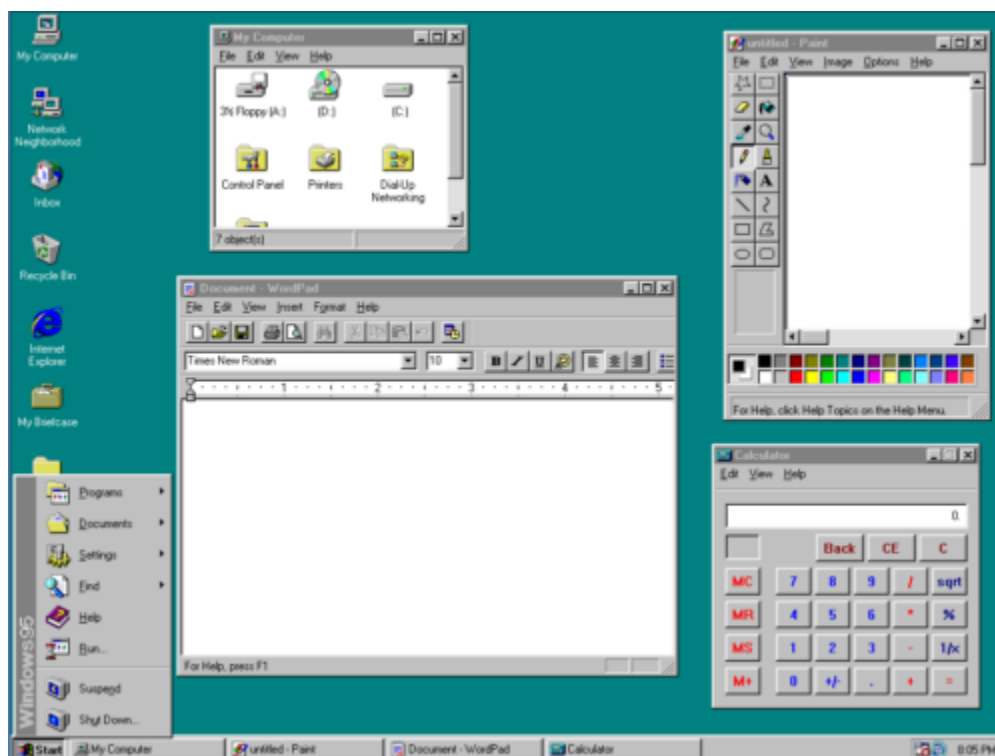
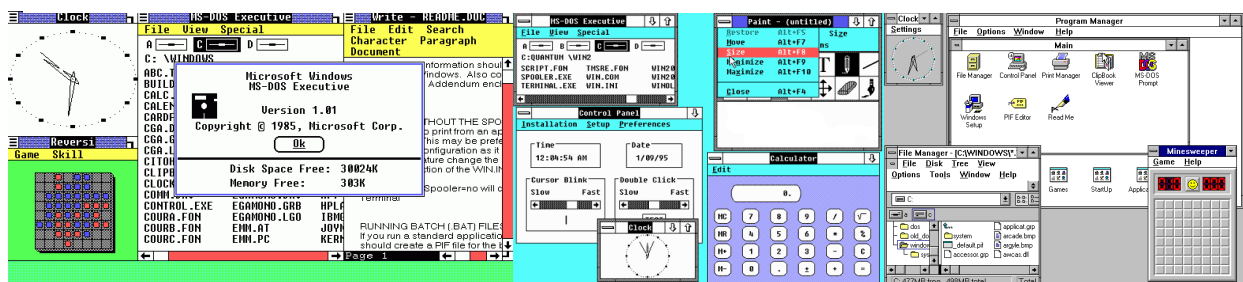


Arhitectura Windows NT

- **Windows Shell.** Componentă a Windows API, permite accesul aplicațiilor la funcționalități oferite de shell-ul sistemului de operare, la fel și schimbarea sau îmbunătățirea lor. Componenta se află în fișierul *shell.dll* pe sisteme Windows pe 16 biți, și în *shell32.dll* pe sisteme Windows pe 32 biți. Este o subcategorie a *User Interface* din categoria *API*. [9][10]
- **Network Services** oferă acces la capabilități de networking ale sistemului de operare. Subcomponentele ei sunt NetBIOS, Winsock, NetDDE, RPC(Remote Procedure Call) și multe altele. [11]

Windows API se ocupă în principal cu interacțiunea dintre sistemul de operare și o aplicație. Pentru comunicația între diversele aplicații Windows Microsoft a dezvoltat o serie de tehnologii pe lângă Windows API. Au început cu DynamicData Exchange (DDE) care a fost urmat de Object Linking and Embedding (OLE) și mai tarziu de Component Object Model (COM).

Windows 1.0, Windows 2.0, Windows 3.1x



Windows 95

Stilul Windows XP



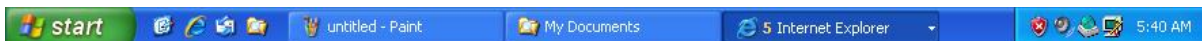
Temele Windows XP sunt customizări ale interfeței grafice pe Windows XP.

Luna este numele de cod pentru stilul vizual implicit pe această versiune de Windows, ilustrat în screenshot-ul de mai sus. Stilul este disponibil în variantele: blue(default, ca-n poza de mai sus), Olive Green și Silver.

Alte teme oficiale mai sunt Zune și Royale(Energy Blue), Royale fiind stil implicit pentru ediția Media Center a Windows XP.

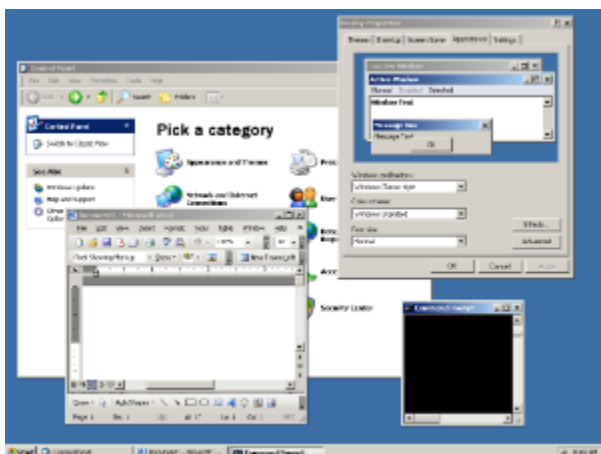
Stilul clasic(Windows 2000) este încă disponibil.

Principalele noutăți aduse de GUI-ul Windows XP sunt: gruparea activităților din taskbar, un meniu Start îmbunătățit(include și o secțiune cu programele cele mai folosite; se poate observa în figura de mai jos).



Windows XP taskbar

Stilul clasic în Windows XP

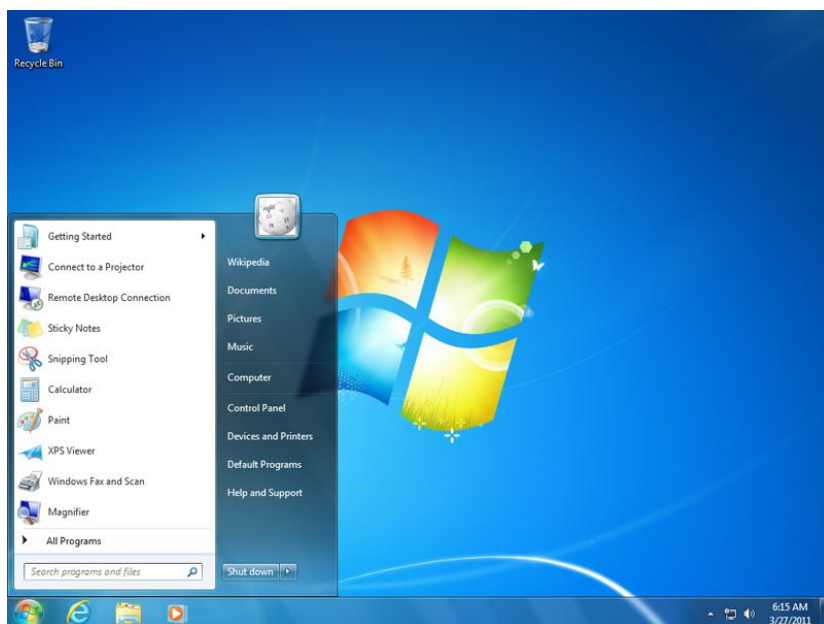


Microsoft pentru a putea rula. Luna este activat automat pe PC-urile cu mai mult de 64 MB RAM.

Luna se referă la un stil vizual particular, nu la toate caracteristicile noi ale interfeței în XP. Unii utilizatori aplică un „patch” fișierului *uxtheme.dll* (care restricționează abilitatea de a folosi stiluri vizuale create de oricine altcineva) pentru a folosi și alte teme.

Windows XP analizează impactul efectelor vizuale asupra performanței și utilizează acest lucru pentru a determina dacă este cazul să le activeze, asta pentru a preveni noua funcționalitate de la a consuma în exces resurse adiționale pentru mai multă procesare. Utilizatorii pot oricând să customizeze aceste setări. Unele efecte, precum *alpha compositing* (transparență și decolorare), sunt tratate automat de plăcile video mai noi. Totuși, dacă placa video nu e capabilă de *alpha blending* în hardware, performanța poate fi substanțial scăzută, și Microsoft recomandă ca această caracteristică să fie oprită manual. Windows XP a adăugat abilitatea de a folosi „stiluri vizuale”(Visual Styles) pentru a schimba aspectul interfeței. Stilurile vizuale trebuie să fie semnate criptografic de

Windows Aero



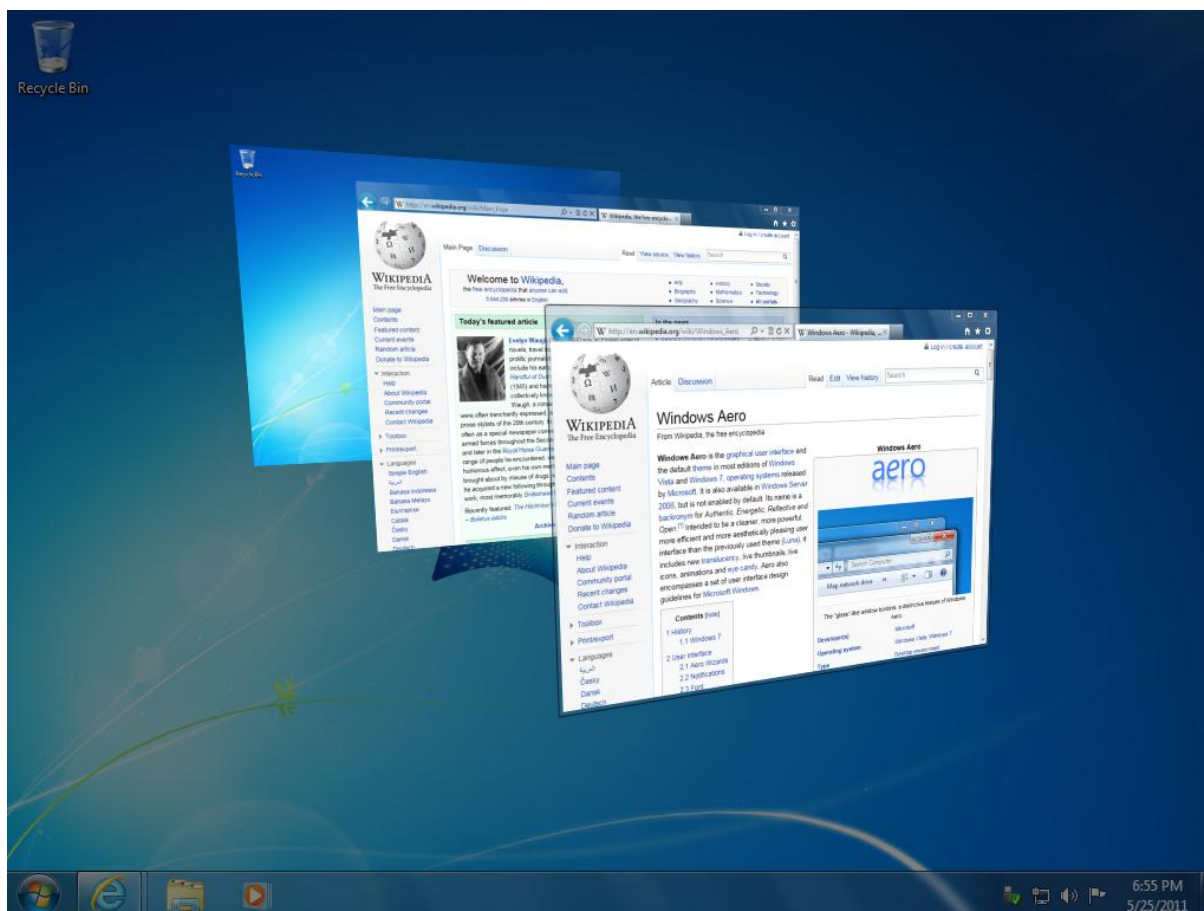
Windows 7 Desktop

Windows Aero este o interfață grafică și tema implicită a mai multor ediții ale Windows Vista și Windows 7, sistemele de operare realizate de Microsoft. Este disponibilă și în Windows Server 2008, dar nu este activată în mod implicit. Numele este un acronim pentru *Authentic, Energetic, Reflective and Open (Autentic, Energetic, Reflexiv și Deschis)*.

Destinată pentru a fi o interfață mai curată, mai puternică, mai eficientă și mult mai plăcută din punct de vedere estetic față de tema precedentă folosită (Luna), ea include transluciditate, miniaturi live, pictograme live și animații (chiar și 3D). Aero cuprinde de asemenea un set de interfețe create și programate de Microsoft Windows.



Border fereastră în Windows 7 cu Windows Aero activat.



Windows Flip 3D



Miniaturi(thumbnails) live în taskbar pentru aplicațiile rulate într-un moment anume.

Windows Aero, interfață accelerată hardware(de GPU), este controlat de **Desktop Window Manager**.

Desktop Window Manager

Desktop Window Manager (DWM, înainte numit Desktop Compositing Engine sau DCE) este managerul de ferestre în Windows Vista și 7, și chiar și-n Windows 8, care activează interfața grafică Windows Aero și tema vizuală cu același nume. DWM-ul necesită plăci video care suportă DirectX 9.0 și Shader Model 2.0. Nu este activat automat în Windows Server 2008 și 2008 R2. În edițiile Starter și Home Basic din Windows Vista și 7 DWM(implicit Aero) este parțial activat.

DWM lucrează în moduri diferite dacă este vorba ori de DWM-ul din Windows 7, ori de cel din Windows Vista, dacă driverele video sunt compatibile WDDM(Windows Display Driver Model) 1.0 sau 1.1. Sub Windows 7 cu drivere WDDM 1.1, DWM doar scrie buffer-ul programului în memoria VRAM(Video RAM), chiar dacă este un program GDI, asta pentru că Windows 7 suportă(limitat) accelerare hardware pentru GDI și astfel nu are nevoie să țină o copie a buffer-ului în memoria RAM a sistemului în care CPU-ul să o (supra)scrie.[2]

Deoarece compositorul are acces la grafica oricărei aplicații, permite cu ușurință efectelor vizuale ce leagă împreună lucruri vizuale de la aplicații multiple, precum transparența. DWM-ul utilizează DirectX 9 pentru a executa funcția de compositing și rendering în GPU(procesorul plăcii video), eliberând CPU-ul de sarcina gestionării bufferelor off-screen către display. Totuși, nu afectează desenarea aplicațiilor către bufferele off-screen; în funcție de tehnologiile folosite, încă poate exista legătură cu CPU-ul. Tehnicile de rendering ale DWM-agnostic precum GDI sunt redirecționate către buffere prin randarea UI ca bitmap. Tehnicile de rendering DWM-aware precum WPF fac direct structura internă a datelor disponibilă în format DWM-compatible. Conținutul ferestrei în buffere este apoi convertit în texturi DirectX.

Desktop-ul în sine este o suprafață 3D full-screen, cu ferestre reprezentate ca o plasă constând din două triunghiuri adiacente (și mutual-inversate), care sunt transformate pentru a reprezenta un dreptunghi 2D. Textura, reprezentând UI chrome, este mapat în acest dreptunghiuri. Tranzițiile ferestrei sunt implementate ca transformări ale plasei, utilizând programe shader.[3]

DWM mapează doar obiectul Desktop primar ca o suprafață 3D; alte obiecte Desktop, inclusiv și desktop-urile virtuale sau cele securizate folosite de User Account Control nu sunt 3D.[4][5]

Cum toate aplicațiile sunt randate către un buffer off-screen, ele pot fi citite în afara buffer-ului embedded în alte aplicații. Cum buffer-ul off-screen este actualizat constant de aplicație, randarea embedded va fi o reprezentare dinamică a ferestrei aplicației și nu o randare statică. Așa lucrează *live thumbnail-urile* și *Windows Flip* în Windows Vista. DWM-ul expune un API public care permite aplicațiilor să acceseze aceste reprezentări de miniaturi(thumbnails). [6] Dimensiunea thumbnail-ului nu este fixată; aplicațiile pot cere thumbnail-uri de orice dimensiune mai mică decât fereastra originală, la aceeași dimensiune, sau chiar mai mare – și DWM-ul le va scala corespunzător înainte de returnarea rezultatului.

Windows Flip 3D nu utilizează miniaturi API publice, de vreme ce nu permit acces direct la texturi Direct3D. [7] În schimb, Flip 3D este implementat direct în motorul(engine) DWM.

Desktop Window Manager utilizează *Media Integration Layer*(MIL), compositorul negestionat partajat cu *Windows Presentation Foundation*(WPF, nume de cod Avalon, subsistem .NET Framework 3.0 pentru rendering de interfețe grafice), pentru a reprezenta ferestrele ca noduri de compoziție într-un copac/arbore de compoziție, reprezentând un desktop și toate ferestrele găzduite în acesta, care sunt apoi randate de MIL de la spatele scenei în față. [8] Cum toate ferestrele contribuie la imaginea finală, culoarea unui pixel rezultat poate fi decis de mai mult de o fereastră. Acest lucru este folosit pentru a implementa efecte precum transparența per pixel. DWM permite shaderi customizați pentru a fi invocați să controleze cum pixelii din aplicații multiple sunt utilizați pentru a crea un pixel rezultat.

DWM include la pachet programe Pixel Shader 2.0 care calculează culoarea unui pixel într-o fereastră prin medierea culorii pixelului determinat de fereastra din spate și pixelii vecini. Acești shaderi sunt folosiți de DWM pentru a obține efect de blur în bordurile ferestrelor gestionate de DWM, și opțional pentru zonele unde aplicația cere acest lucru.[3][6][8]

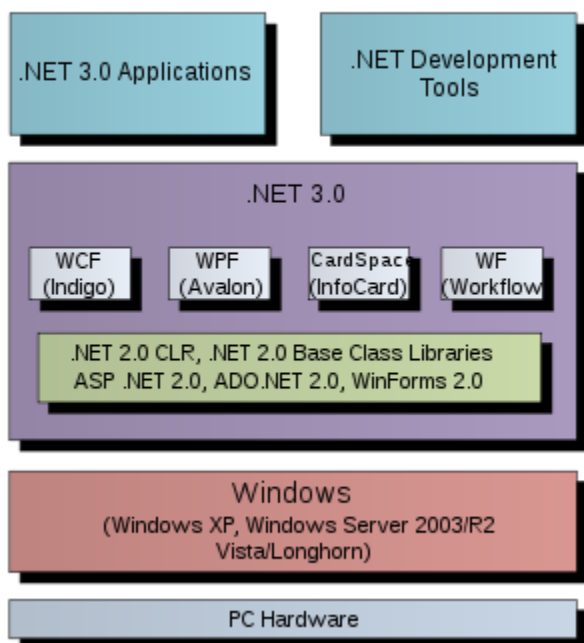
Cum MIL oferă un sistem cu mod restrâns de grafică prin caching-ul arborelui de compoziție, sarcina de redesenare și refresh a ecranului când ferestrele sunt mișcate este manevrat de DWM și MIL, eliberând aplicația de responsabilitate. Background-ul datei este deja în arborele de compoziție și în bufferele off-screen, ce este folosit

direct pentru a randă background-ul, fără ca aplicațiile din background să re-randeze înseși trimițând mesaje WM_PAINT, cum a fost cazul cu sistemele de operare mai vechi de Vista.

DWM utilizează grafică în buffer dublu pentru a preveni pâlpâirea și ruperea în timpul mișcării ferestrei cu mouse-ul. Motorul de compositing utilizează optimizări precum *culling* pentru a îmbunătăți performanța, de asemenea nedesenând zone ce n-au fost schimbate. Deoarece compositorul este și multi-monitor, DWM suportă asta nativ. În timpul rulării aplicațiilor full-screen, precum jocurile, DWM nu execută compositing de ferestre și astfel performanța nu va scădea apreciabil.[9]

Windows Presentation Foundation

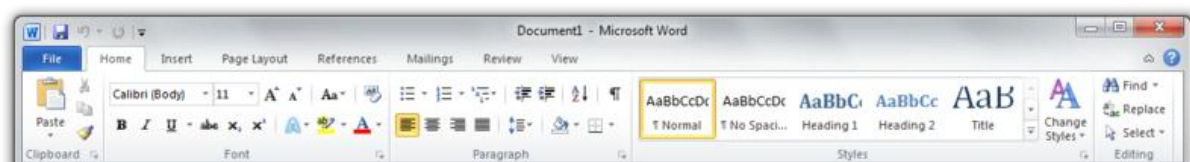
.NET 3.0 Stack



WPF este un subsistem grafic de randare a interfețelor cu utilizatorul în aplicațiile Windows.

Cunoscut inițial ca *Avalon*, a fost lansat cu .NET Framework 3.0. În loc să se bazeze pe GDI, utilizează DirectX. Oferă un model de programare consistent pentru a construi aplicații, și mai oferă o separare între interfața cu utilizatorul și logica de business. Se aseamănă cu modele XML obiect-orientate, ca implementate în XUL și SVG.

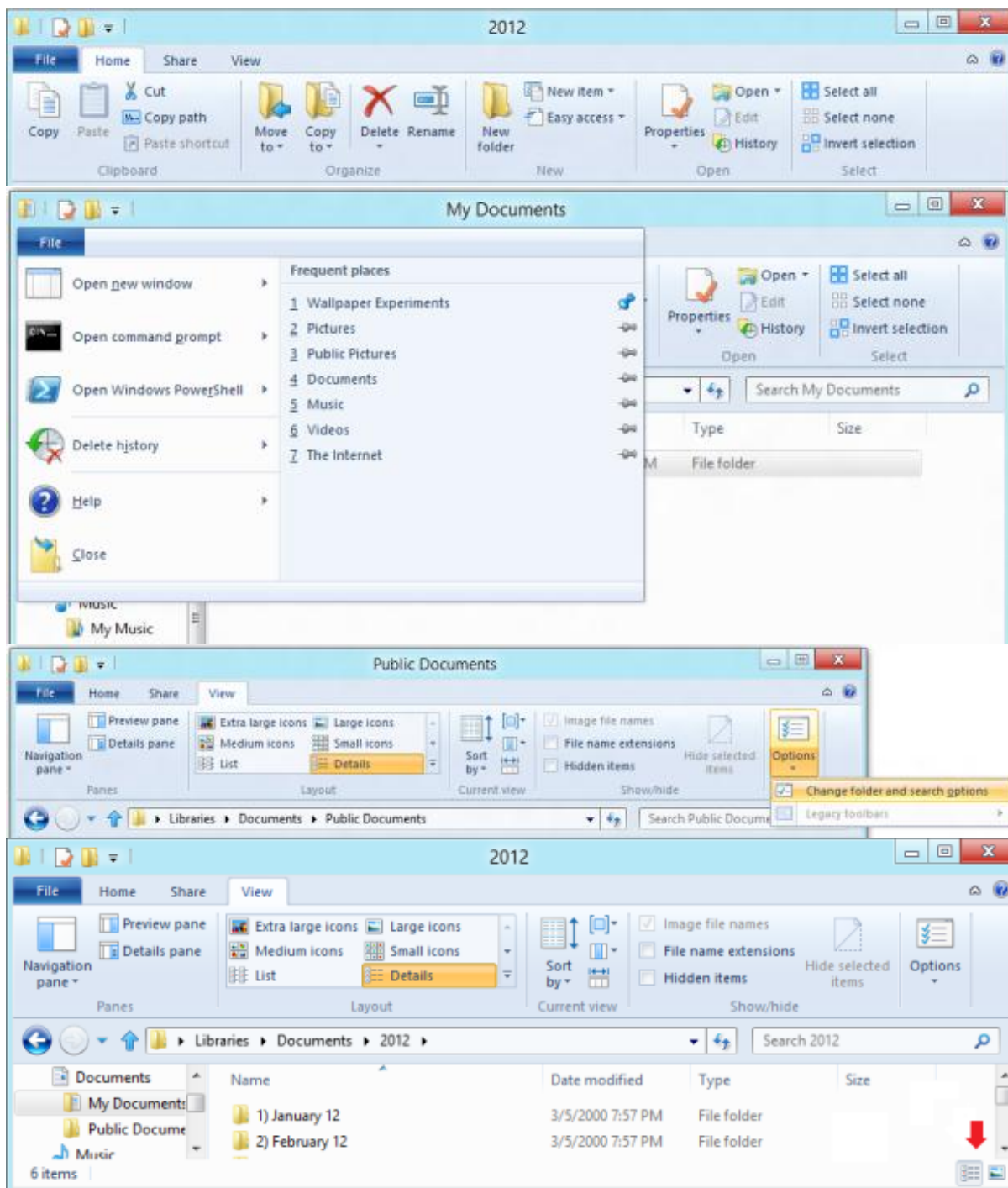
Interfața Ribbon



Interfața Ribbon în Microsoft Office 2010.

Este o interfață cu un set de toolbar-uri sunt plasate pe pe tab-uri într-o bară de tab-uri. GUI-ul Ribbon oferă interfața unei aplicații cu o lungă bară de unelte(toolbar) umplut cu reprezentări grafice ale unor elemente de control care sunt grupate după diferite funcționalități. Ribbon-ul poate de asemenea să conțină tab-uri care expun diferite seturi de elemente de control, eliminând nevoia de multe alte toolbar-uri bazate pe pictograme (icons). Unele dintre aceste tab-uri sunt contextuale și apar doar când un anumit tip de obiect e selectat, oferind unelte specifice pentru elemente precum tabele sau imagini.

A apărut inițial cu Microsoft Office 2007, iar reacțiile foarte pozitive (utilizarea mai eficientă) au determinat ulterior integrarea acestuia în meniurile tot mai multor aplicații Windows. Astfel începând cu Windows 7 s-a integrat Ribbon UI și-n utilitare precum Paint și Wordpad. În Windows 8 aceasta a fost integrată chiar și în Windows Explorer.



Metro (limbaj de design) & Windows 8



Interfața Metro în Windows 8

Este un limbaj de design creat de Microsoft pentru utilizarea acestuia pe platforma Windows Phone și Windows Media Center. Ulterior, cum tehnologia tinde să se îndrepte cât mai mult spre mobilitate, Microsoft a decis să unifice interfața sistemului de operare Windows pe toate dispozitivele, făcând în același timp sistemul de operare capabil să ruleze atât pe Desktop/laptop, cât și pe smartphone-uri, tablete bazate pe procesoare ARM (printr-o versiune RT). Este practic vorba de același sistem de operare, același kernel, aceeași interfață grafică, adaptat în funcție de nevoi și utilizare.

Scopul acestei interfețe este de a mări viteza utilizării aplicațiilor, punând conținutul aplicației pe prim plan, astfel aceasta devine interfață principală.

Meniul Start este înlocuit cu un Start Screen, în care se găsesc miniaturi live ale programelor principale din sistem, însă și celelalte programe pot fi găsite destul de ușor prin scrierea unor bucăți din numele lor, căutarea făcându-se imediat sau printr-un meniu contextual din care se alege afișarea tuturor programelor, chiar și a celor clasice, necompatibile cu Metro UI.

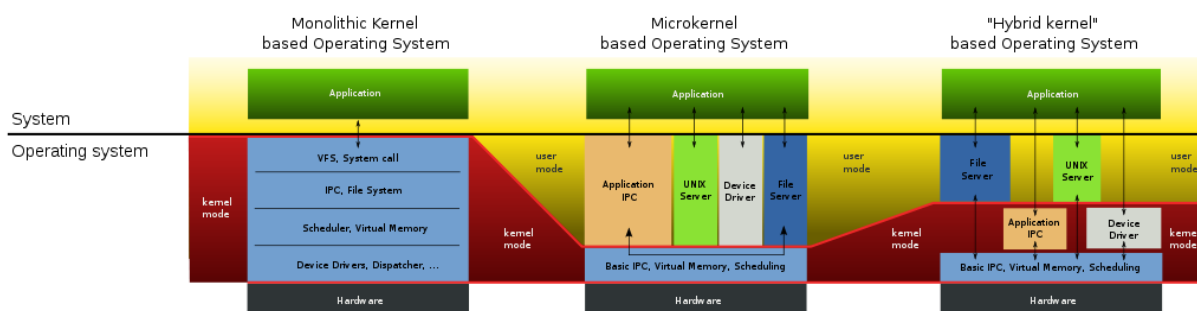
Butonul de Start a fost eliminat de pe taskbar-ul Desktop-ului și a fost înlocuit de un Charm Bar.

Această interfață este optimizată pentru ecranele tactile.



4. Comparație interfețe grafice

- În Linux interfața grafică este opțională, foarte customizabilă, pe când în Windows este parte integrantă din sistemul de operare, cu toate că inițial era doar extensie GUI pentru MS-DOS, iar din acest punct de vedere Linux este mai stabil. Avantajul Linux este că poate fi folosit spre deosebire de Windows și fără o interfață grafică. Poate fi folosit doar dintr-o interfață text(shell). Avantajul sistemului de operare Windows este că mai toate activitățile, de la simple la complexe, se pot desfășura într-un mediu grafic, iar instalarea aplicațiilor decurge foarte ușor, utilizatorul având doar sarcina de a apăsa niște butoane precum Next, Browse, Finish, și eventual de selectat anumite caracteristici de instalat ale programului respectiv. În Linux însă, partea grafică nu e la fel de dezvoltată ca cea din Windows, fiind nevoie de multe ori apelarea unui Terminal pentru instalarea unui program sau pentru configurarea anumitor setări de sistem, precum și instalarea unui driver. În Windows utilizarea unui prompt DOS(linie de comandă) nu este absolut necesară. Din acest motiv Windows-ul este preferat de utilizatorii casnici, fiind ușor de instalat, manevrat, oferind acces mult mai ușor la toate aplicațiile instalate, ba chiar și suportul din partea producătorilor de drivere și jocuri bazate pe DirectX este mult mai bun, dar și la Linux e în continuă creștere.
- Ambele sisteme de operare sunt multitasking și multiuser, mai ales datorită existenței GUIs.
- Denumirea de Linux nu se referă la un sistem de operare cu interfață grafică, ci doar la nucleu/kernel. Există distribuțiile bazate pe Linux, având o anumită interfață sau anumite programe preinstalate. Se poate alege din numeroase categorii, principale fiind GNOME, KDE, XFCE, astfel s-a ajuns la un număr mai mare de distribuții. Linux este un kernel monolithic, pe când kernel-ul familiei Windows NT este hybrid.



- Linux și distribuțiile bazate pe Linux, inclusiv majoritatea aplicațiilor, chiar și interfețele grafice reprezintă software gratuit și open-source(asta însemnând că orice programator poate modifica după bunul plac orice bucată dintr-o aplicație sau chiar întreaga aplicație), pe când Windows e în totalitate comercial și closed-source.
- Ambele sisteme de operare oferă performanțe superioare, la ambele dezvoltatorii software au putut realiza interfețe grafice accelerate hardware, reducând consumul CPU-ului, prin utilizarea GPU-ului la desenarea și animarea interfeței grafice, însă cu un cost cam la fel de crescut de memorie RAM și VRAM. Ambele sisteme de operare au actualmente o interfață vizuală bogată, atractivă și experiența vizuală este comparabilă.

5. BIBLIOGRAFIE

- *Operating Systems: Design and Implementation*, A. Tanenbaum, A. Woodhull
- *Modern Operating Systems*, A. Tanenbaum
- Resurse Web

1. <http://ieeexplore.ieee.org/>
2. <http://www.linfo.org/gui.html>
3. <http://www.catb.org/~esr/writings/taouu/html/ch02s05.html>
4. <http://www.parc.com/about/history/>
5. <http://xfce.org>
6. <http://www.gnome.org>
7. <http://www.freedesktop.org>
8. <http://gnome-look.org>
9. <http://www.kde.org>
10. <http://kde-look.org>
11. <http://technet.microsoft.com/en-us/sysinternals/bb897447.aspx>
12. <http://msdn2.microsoft.com/en-us/library/ms809754.aspx>
13. http://msdn.microsoft.com/library/default.asp?url=/library/en-us/winprog/winprog/overview_of_the_windows_api.asp
14. http://msdn.microsoft.com/library/default.asp?url=/library/en-us/winprog/winprog/windows_shell.asp
15. http://msdn.microsoft.com/library/default.asp?url=/library/en-us/winprog/winprog/user_interface.asp
16. <http://msdn.microsoft.com/en-us/library/hh270423%28v=vs.85%29.aspx>
17.
 - I. ^{^ a b c} Greg Schechter. "DWM's use of DirectX, GPU and hardware acceleration". *Greg Schechter's Blog*. MSDN Blogs.
http://blogs.msdn.com/greg_schechter/archive/2006/03/19/555087.aspx.
 - II. [^] Greg Schechter. "Responding to Comments from "DWM's use of DirectX, GPU and hardware acceleration"". *Greg Schechter's Blog*. MSDN Blogs.
http://blogs.msdn.com/greg_schechter/archive/2006/03/25/561110.aspx.
 - III. [^] Chris Jackson. "Desktop Window Manager only runs on the primary desktop". *Chris Jackson's Semantic Consonance*. MSDN Blogs.
<http://blogs.msdn.com/cjacks/archive/2006/11/09/a-desktop-of-your-own.aspx>.
 - IV. ^{^ a b c} Greg Schechter. "Under the hood of Desktop Window Manager". *Greg Schechter's Blog*. MSDN Blogs.
http://blogs.msdn.com/greg_schechter/archive/2006/03/05/544314.aspx.
 - V. ^{^ a b c} Greg Schechter. "How underlying WPF concepts and technology are being used in the DWM". *Greg Schechter's Blog*. MSDN Blogs.
http://blogs.msdn.com/greg_schechter/archive/2006/06/09/623566.aspx.
 - VI. ^{^ a b c} Greg Schechter. "Redirecting GDI, DirectX, and WPF applications".
http://blogs.msdn.com/greg_schechter/archive/2006/05/02/588934.aspx. Retrieved 2007-10-14.
 - VII. [^] "MSDN - DirectX SDK - DirectX3D 10 - D3D10 FEATURE LEVEL". Microsoft.
[http://msdn.microsoft.com/en-us/library/ee415631\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ee415631(VS.85).aspx).
18. <http://msdn.microsoft.com/en-us/library/aa969540.aspx>
19. <http://dwmexe.com/>

20. <http://msdn.microsoft.com/en-us/library/ms754130.aspx>. WPF Library
21. <http://blogs.msdn.com/b/wpfsdk/>. Windows Presentation Foundation
22.
 - Jensen Harris (2005–2008). "An Office User Interface Blog". [Microsoft Developer Network](http://blogs.msdn.com/jensenh/pages/table-of-contents.aspx). <http://blogs.msdn.com/jensenh/pages/table-of-contents.aspx>. Retrieved 2010-03-25.— Extensive discussion of the UI design by Microsoft's Group Program Manager of the Office 2007 User Experience team.
 - "MIX08 Microsoft Office 2007: The Story of the Ribbon". [Microsoft MIX 08](http://www.youtube.com/watch?v=TI9kD693ie4). 2008. <http://www.youtube.com/watch?v=TI9kD693ie4>. Retrieved 2010-06-17.- Prototype sketches and design process.
23.
 - [Metro Design Language of Windows Phone 7](#), Microsoft
 - [User Experience Design Guidelines for Windows Phone](#), Microsoft
 - [What is metro design & what are its principles](#), Elevatelocal

Paragrafe ale altor autori

Capitolul 1 (traducerea și sinteza unor paragrafe)

[Jeremy Reimer. "A History of the GUI" Ars Technica](#) [1]

["GUI definition". Linux Information Project](#). [2]

Capitolul 2 (traducerea și sinteza unor paragrafe)

1. <http://www.x.org/>
2. <http://www.linfo.org/x.html>
3. <http://www.art.net/~hopkins/Don/unix-haters/x-windows/disaster.html>
4. <http://lists.debian.org/debian-user/1996/11/msg00637.html>
5. [Robert W. Scheifler and James Gettys: X Window System: Core and extension protocols: X version 11, releases 6 and 6.1, Digital Press 1996](#)
6. <http://xwinman.org/>
7. <http://www.gnome.org/>
8. <http://lists.debian.org/debian-user/1997/08/msg02286.html>
9. http://linuxtoday.com/news_story.php3?ltsn=2000-09-05-001-21-OP-LF-KE
10. <http://www.gnu.org/licenses/why-not-lgpl.html>
11. <http://www.kde.org/>

Capitolul 3 (anumite paragrafe însemnate sunt traduse)

Windows USER

1. [^] [Petzold, Charles \(1998\). *Programming Windows Fifth Edition*, Microsoft Press, ISBN 978-1-57231-995-0](#) .
2. [^] [Chen, Raymond \(2007\). *The Old New Thing*, Addison-Wesley Professional, ISBN 978-0-321-44030-3](#)
3. [^] [Windows Ribbon Framework \[1\]](#)

Windows API

1. [^] [Microsoft TechNet](#) (November 2006). *Inside Native Applications*.
2. [^] ^a ^b [Microsoft Developer Network](#) (July 2005). *Overview of the Windows API*.
3. [^] [Microsoft Developer Network](#) (July 2005). *Base Services*.
4. [^] [Microsoft Developer Network](#) (July 2005). *Graphics Device Interface*.
5. [^] ["G". Microsoft Developer Network. http://msdn.microsoft.com/en-us/library/ms789535.aspx.](#)
6. [^] [Microsoft Developer Network](#) (July 2005). *User Interface*.
7. [^] [Microsoft Developer Network](#) (2005). *Common Dialog Box Library*.
8. [^] [Microsoft Developer Network](#) (July 2005). *Common Control Library*.
9. [^] [Microsoft Developer Network](#) (July 2005). *Windows Shell*.
10. [^] [Microsoft Developer Network](#) (2005). *Shell Programmer's Guide*.
11. [^] [Microsoft Developer Network](#) (July 2005). *Network Services*.

Desktop Window Manager

1. [^] <http://blogs.msdn.com/e7/archive/2009/04/25/engineering-windows-7-for-graphics-performance.aspx>
2. [^] ^a ^b ^c [Greg Schechter. "DWM's use of DirectX, GPU and hardware acceleration". *Greg Schechter's Blog*. MSDN Blogs. http://blogs.msdn.com/greg_schechter/archive/2006/03/19/555087.aspx.](#) [^] [Greg Schechter. "Responding to Comments from "DWM's use of DirectX, GPU and hardware acceleration"". *Greg Schechter's Blog*. MSDN Blogs. http://blogs.msdn.com/greg_schechter/archive/2006/03/25/561110.aspx.](#) [^] [Chris Jackson. "Desktop Window Manager only runs on the primary desktop". *Chris Jackson's Semantic Consonance*. MSDN Blogs. http://blogs.msdn.com/cjacks/archive/2006/11/09/a-desktop-of-your-own.aspx.](#)
3. [^] ^a ^b ^c [Greg Schechter. "Under the hood of Desktop Window Manager". *Greg Schechter's Blog*. MSDN Blogs. http://blogs.msdn.com/greg_schechter/archive/2006/03/05/544314.aspx.](#)
4. [^] [\[1\]](#)
5. [^] ^a ^b ^c [Greg Schechter. "How underlying WPF concepts and technology are being used in the DWM". *Greg Schechter's Blog*. MSDN Blogs. http://blogs.msdn.com/greg_schechter/archive/2006/06/09/623566.aspx.](#)
6. [^] ^a ^b ^c [Greg Schechter. "Redirecting GDI, DirectX, and WPF applications". http://blogs.msdn.com/greg_schechter/archive/2006/05/02/588934.aspx.](#)
9. [^] ^a ^b ^c [Greg Schechter. "Redirecting GDI, DirectX, and WPF applications". http://blogs.msdn.com/greg_schechter/archive/2006/05/02/588934.aspx.](#)

Capitolul 4 (doar concluziile personale)