

2012

Structura sistemelor de operare

aspecte generale si particulare pentru
Windows si Linux

Studenti:

Băbăruș George 431A

Voicu Adrian-Nicolae 431A

ETTI

01.06.2012

CUPRINS:

Partea I: Structura sistemelor de operare - Băbăruș George grupa 431A

1.1 Structura de baza a fiecarui sistem de operare in parte: concepte generale si particulare

1.1.1 Concepte generale

1.1.2 Serviciile oferite de sistemele de operare

1.2 Structura particulara a sistemelor de operare

1.2.1 Structura sistemului de operare MS-DOS

1.2.2 Structura sistemului de operare Windows XP

1.2.3 Structura sistemului de operare UNIX

1.2.4 Sistemul de operare linux

Partea II: Win32 API si registrul de informatii pentru Windows — Voicu Adrian-Nicolae 431A

2.1 Win32 API

2.1.1 Introducere

2.1.2 Windows API

2.1.3 Versiuni

-Win16

-Win32

-Win32s

-Win64

2.1.4 Aplicabilitatea in .NET

2.2 Registrul de informatii pentru Windows

2.1 Structura

Partea I: Structura sistemelor de operare

Babarus George 431A

1.1 Structura de baza a fiecarui sistem de operare in parte: concepte generale si particulare

1.1.1 Concepte generale

Structura sistemului de calcul se refera componentelor functionale si structurale ale sistemelor de operare care permit realizarea operatiilor si gestiunea operatiilor de prelucrare de date tot mai necesare astazi.

Din cauza necesitatii prelucrarii complexe a datelor structura sistemelor de operare este din ce in ce mai complexa pentru a satisface nevoile utilizatorului final. Desi structura este din ce in ce mai complexa, aceasta se tine partitionata in parti mici in asa fel incat sa se mentina specializarea componentelor. Astfel fiecare componenta are o functie bine definita si o metoda specifica de rezolvare a cerintelor.

Structura uzuala a unui sistem de operare:

- Manager-ul de proces
- Intrumente de gestionare a memoriei principale
- Sistem de gestiune a fisierelor
- Sistem de manipulare a intrarilor si iesirilor sistemului
- Sistemul de memorie secundara
- Gestionarea sistemelor distribuite
- Sistem de protectie
- Interpretor de comenzi

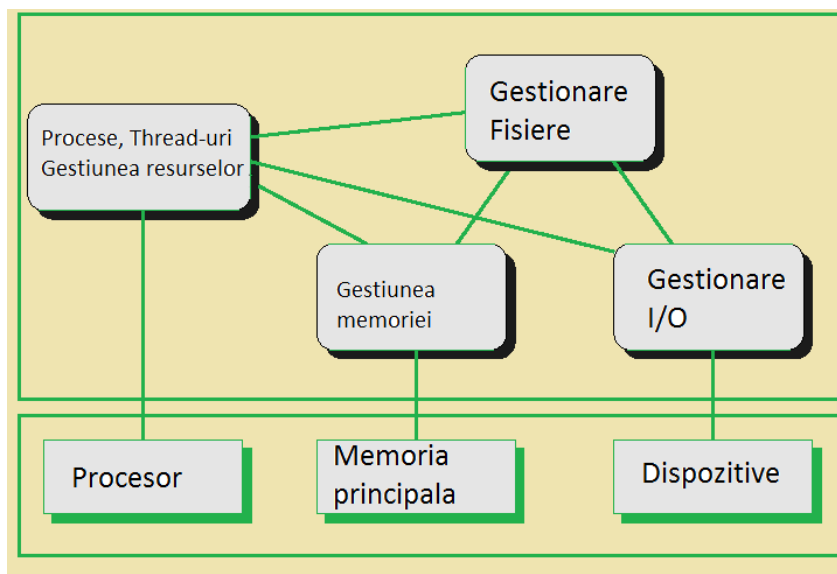


Figura 1.:Sistemul de operare

1) Manager-ul de proces

Un proces este un program aflat in executie. Un proces are nevoie de resursele sistemului de calcul pentru a-si indeplini rolul: Procesor, memorie, fisiere, intrari si iesiri din sistem. Resursele sunt alocate procesului de catre sistemul de operare in momentul inceperii executiei acestuia, si de asemenea sunt eliberate tot de catre sistemul de operare la sfarsitul executiei (inchiderea procesului). Deci sistemul de operare are urmatoarele indatoriri conform acestei functii: sa creeze si sa inchida procesul, sa il suspende si sa il reia atunci cand este necesar acest lucru (in cazul in care un alt proces este mai prioritar), si de asemenea tot rolul sistemului de operare este sincronizarea si comunicarea intre mai multe procese.

2) Intrumente de gestionare a memoriei principale

Memoria este o insiruire de numeroase locatii de memorie si desi dimensiunea este in crestere accesul la memorie si lucrul cu aceasta trebuie sa se desfasoare repede, pentru a permite procesorului o rulare cat mai rapida. Drept urmare a celor spuse mai sus este nevoie de folosirea unui dispozitiv special si a unei proceduri speciale de gestiune a memoriei. Sistemul de operare trebuie sa aiba grija de urmatoarele aspecte in partea de lucru cu memoria: trebuie sa stie ce parte a memoriei este folosita si de cine, sa decida care proces sa se apeleze atunci cand este incarcata o anumita parte a memoriei, sa aloce si sa stearga spatii de memorie necesare proceselor.

3) Gestiunea fisierelor

Sunt componentele sistemului de operare cu care lucreaza cel mai mult

utilizatorul final si permite calculatoarelor sa stocheze informatia pe diferite dispozitive. Un fisier este o reprezentare logica si o abstractizare a informatiei stocate

fizic. Este o colectie de informatii relationate intr-un anumit fel de creatorul fisierului, ele fiind catalogate in programe si date.

Sistemul de operare este responsabil de urmatoarele functii in cadrul gestionarii fisierelor: crearea si stergerea fisierelor, crearea si stergerea directoarelor, sa ofere functii de gestionare a fisierelor, sa salveze datele in memoria nevolatila.

4) Sistem de manipulare a intrarilor si iesirilor sistemului

Rolul sistemului de operare este alocarea resurselor in comunicarea procesorului cu alte dispozitive conectate la sistem prin porturile special prevazute in acest sens. Tot sistemul de operare manipuleaza si intreruperile solicitate in comunicarea cu alte dispozitive

5) Manipularea memoria secundare

Memoria secundara este o memorie mai lenta insa necesara sistemului datorita capacitatii mari de stocare si de asemenea datorita faptului ca este nevolatila. Sistemul de operare are responsabilitatea alocarii memoriei, gestionarii spatiului liber si a gaurilor de spatiu neocupat si stergerea memorie.

6) Gestionarea sistemelor distribuite

Un sistem distribuit este o colectie de procesoare care nu impart memorie sau acelasi ceas, ci au fiecare pe al lui. Procesoarele in aceste sisteme sunt conectate printr-o retea de comunicatie si comunicatia are loc prin protocoale. Un sistem distribuit ofera utilizatorului acces la o gama larga de resurse si aici intervine rolul sistemului de operare.

7) Sistemul de protectie

Protectia se refera la mecanismul de control a accesului programelor, proceselor, si utilizatorilor atat la sistem cate si la resursele acestuia.

Mecanismul de protectie trebuie sa distinga intre utilizările autorizate si neautorizate la sistem. Rolul sistemului de operare este de a cataloga gradul de acces la anumite cereri de acces la resurse si de a permite sau nu accesul la acestea.

8) Sistemul de interpretare a comenzilor

Multe dintre comenzile date sistemului de operare au drept scop gestionarea tuturor responsabilitatilor subliniate mai sus: crearea proceselor, stergerea, gestionarea I/O, gestiunea memoriei, acces la fisiere, protectie, gestiunea memoriei principale.

Acest program citește instrucțiunile și le execută în timp real; Denumirea sa este interpretor de linie de comandă sau Shell pe Unix.

1.1.2 Serviciile oferite de sistemele de operare

Principalele servicii oferite de un sistem de operare sunt enumerate mai jos:

- Executarea programelor – sistemul de operare trebuie să fie capabil să preia un program din memorie și să îl ruleze
- Operații I/O – programele nu sunt în general capabile să execute operații de intrare ieșire, din această cauză un serviciu important pe care sistemul de operare trebuie să îl pună la dispoziție este asigurarea unui mijloc prin care să se realizeze operații I/O.
- Comunicarea între procese – este nevoie de un nivel superior pentru a facilita comunicarea între procese.
- Mecanism de detectie a erorilor – trebuie să asigure detectia și corectia erorilor aparute în procesor, în memorie și în sistemele I/O pentru a permite utilizarea programelor.
- Alocarea de resurse
- Protecție
- Apeluri de sistem – asigură trecerea de la un proces la altul sau executarea unor proceduri specifice cum sunt cele de întrerupere. Asigură de asemenea trecerea parametrilor de la un proces la altul.
- Mecanisme de transmitere a mesajelor.

Apelurile de sistem: aceste apeluri pun la dispoziție interfețe între procese și sistemul de operare. Aceste apeluri se pot realiza în general prin limbajul de asamblare sau pentru unele sisteme apelurile de sistem se pot realiza chiar și din limbajul de nivel înalt, cum ar fi C.

Există trei metode de a iniția apeluri care să permită trecerea parametrilor de la un proces la altul sau direct către sistemul de operare: transferul prin intermediul registrelor, salvarea parametrilor într-o tabelă a cărei adrese se salvează într-un registru, inserarea parametrilor în stivă și extragerea ulterioară.

Aceste apeluri de sistem sunt de mai multe tipuri:

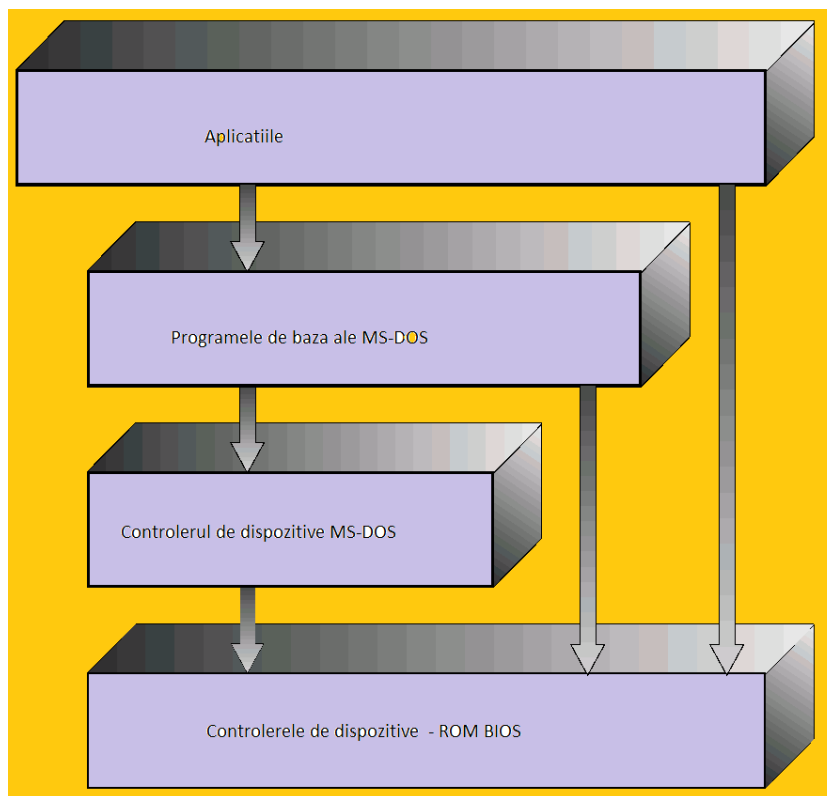
- Procese de control: preia, execută, închide, creează procese, alocă memorie, așteaptă evenimente.
- De gestiune a fișierelor: creează/sterge fișiere, deschide/inchide, citește/scrie fișiere
- Gestiune a dispozitivelor: trimite/recepționează date către dispozitive
- Comunicare : creează sau închide conexiuni de comunicație, trimite și recepționează mesaje

1.2 Structura particulara a sistemelor de operare

Structura sistemelor de operare moderne trebuie bine proiectata datorita dimensiunilor si complexitatii mari a acestora. Si cea mai buna abordare este din punctul meu de vedere divizarea si specializarea pe componente care sa aiba functii specifice si pe care sa „stie” bine sa le indeplineasca. Aceste subunitati specifice sunt definite mai sus si reprezinta criteriul de performanta in cadrul unui sistem de operare modern.

1.2.1 Structura sistemului de operare MS-DOS

MS-DOS este un sistem de operare dezvoltat pe platforma DOS. Acest sistem de operare este dezvoltat pentru sistemele de calcul cu procesoare compatibile X86 si este un sistem de operare utilizat in prezent la pornirea sistemelor de calcul pentru a permite accesul la resurse.



Are o structura simpla care ii permite sa aiba acces rapid la resurse si sa ofere rapiditate utilizatorului. Facilitatile principale ale acestui sistem de operare ofera utilizatorului un interpretor de linie de comanda, prin intermediul caruia ii este permis utilizatorului sa execute functiile specifice unui sistem de operare: gestionare de fisiere, executarea aplicatiilor etc. MS-DOS este integrat si astazi pe majoritatea sistemelor de calcul si permite pornirea rapida a acestuia. De asemenea este integrat in toate sistemele de operare Windows si permite interpretarea liniilor de comanda in aceste sisteme.

1.2.2 Structura sistemului de operare Windows XP

Windows XP aduce o imbunatatire considerabila in editiile sitemelor de operare windows prin performantele si stabilitatea oferite utilizatorului mult peste editiile anterioare ale windows. Prezinta o buna gestionare a memoriei si interfata schimbata ceea ce atrage si un succes mai mare fata de editiile anterioare.

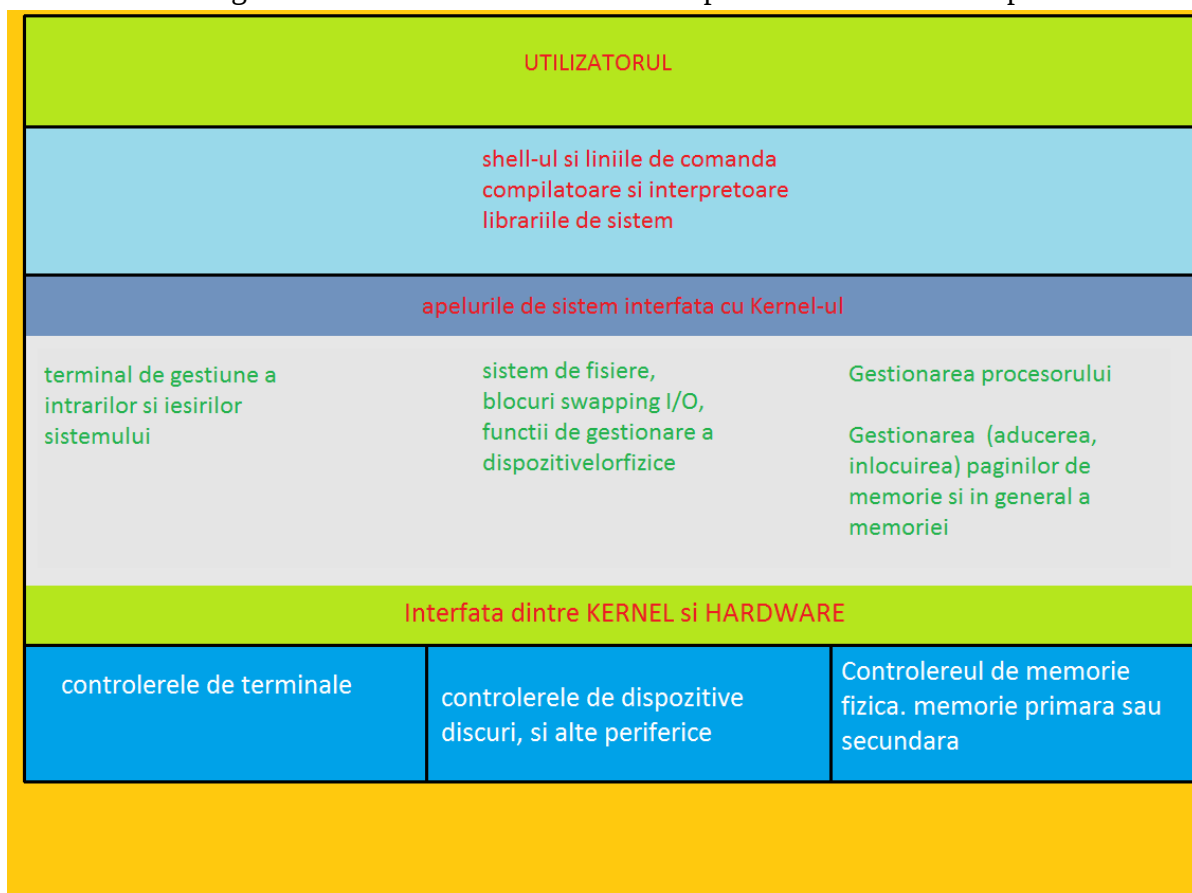
O alta facilitate importanta este permiterea gestionarii mai rapide a resurselor de memorie mai mari prin mecanisme specifice.

1.2.3 Structura sistemului de operare UNIX

Sistemele de operare UNIX sunt stucturate in doua parti separate si de o importanta mare:

- Programele de sistem - folosesc kernel-ul pentru a pune la dispozitie functii importante cum ar fii compilarea, si gestionarea fisierelor
- Kernel-ul – reprezinta tot ceea ce exista sub nivelul apelurilor de sistem dar deasupra nivelului hardware. Pe el se bazeaza apelurile de sistem si ofera facilitati specifice gestionarii partii hardware si comunicarea acesteia cu partea software de niveluri mai inalte.

Pune la dispozitie altor parti ale sistemului de calcul sistem de gestionare al CPU-ului, sistem de gestionare a memoriei si alte finctii specifice sistemului de operare.



KERNEL –ul este un nucleu monolitic care are posibilitatea de incarcare a sistemelor suplimentare. Aceasta este o trasatura specifica tuturor sistemelor de operare unix.

Pe baza nucleului UNIX sunt realizate in prezent o gama larga de sisteme de operare cum ar fii: Solaris (produs de firma SUN Microsystems, realizat pe arhitectura SPARC, X86, X86-64), HP-UX produs de firma HP si ruleaza pe platforma IA-64, AIX- produs de firma IBM, IRIX produs de firmaSGI, MacOS X produs de firma Apple pentru calculatoarele Macintosh ruleaza pe arhitectura X86 si PowerPc, Linux initiat de Linus Torvalds, BSD.

Structura stratificata a sistemelor de calcul este un lucru bun deoarece tratarea problemelor de implementare se poate rezolva pe niveluri si rezolvarea problemelor la un nivel poate oferi functionalitati nivelelor superioare.

1.2.4 Sistemul de operare linux

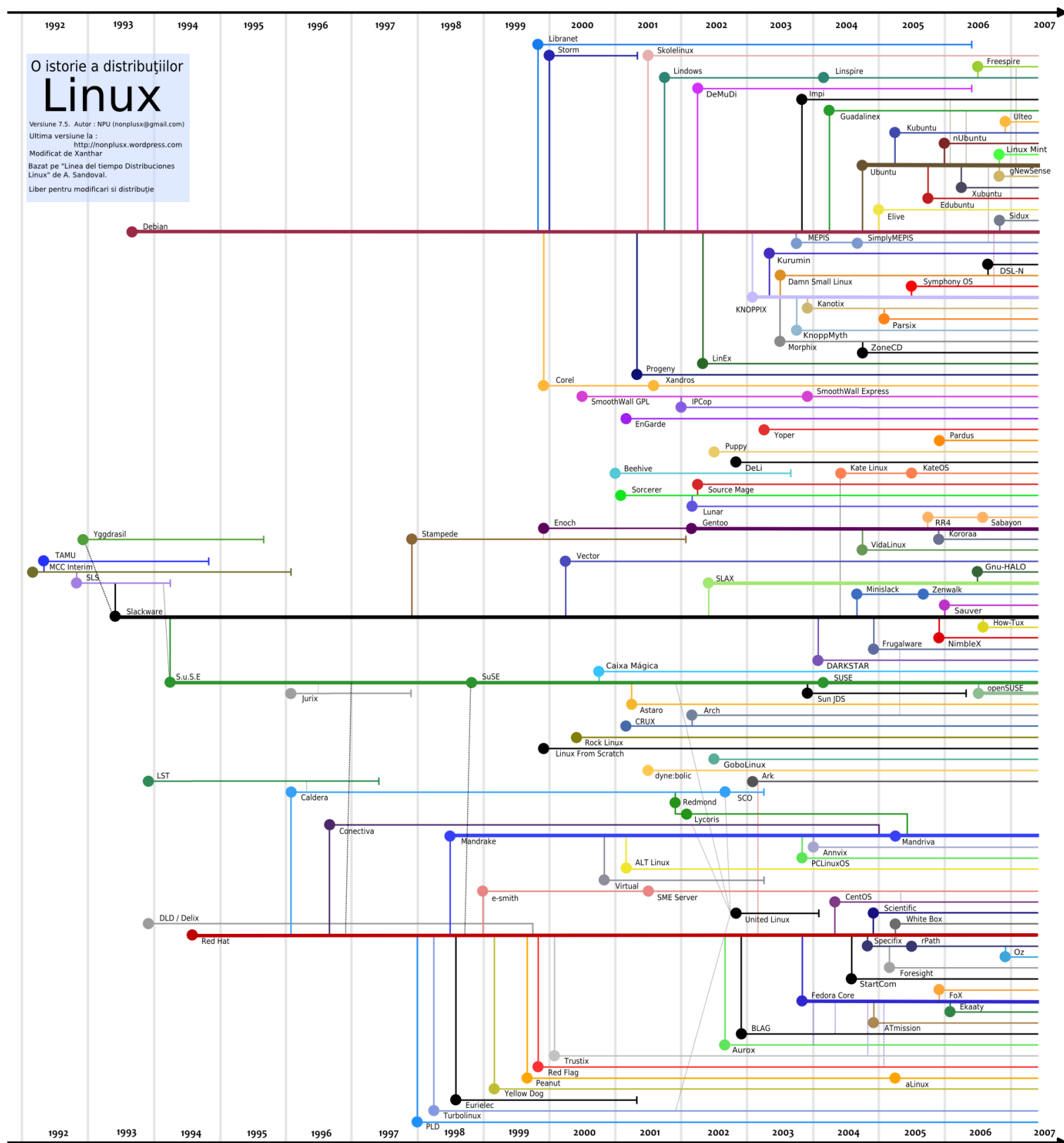
Sistemul de operare linux este un sistem de operare open source bazat pe structura si functionarea nucleului UNIX si este disponibil intr-o gama larga de distributiipentru asta am afisat urmatoarea poza care ilustreaza o cronologie a celor mai importante distributii

Linux este caracterizat de stratificarea specifica UNIX si ofera utilizatorilor stabilitate si performanta ridicata, drept pentru care este foarte popular in utilizarea pentru servere.

Linux are o structura stratificata care presupune divizarea in mai multe niveluri, fiecare nivel fiind bazat pe nivelul inferior lui. Nivelul cel mai de jos in acest context este hardware-ul iar cel mai de sus este interfata grafica.

Stratificarea pe niveluri permite abstractizarea informatiei si incapsularea acesteia pe categorii de nivele pentru care este dedicata. De asemenea aceasta stratificare permite si modularizarea procedurilor si serviciilor oferite de fiecare nivel in parte.

source image wikipedia.org



Referinte bibliografice:

- Andrew S. Tanenbaum - Modern Operating Systems - 2nd Edition
- Amjad Mahmood, Operating Systems
- Györödi Robert, Sisteme de operare
- Wikipedia.org
- Microsoft.com

PARTEA II: Win32 API si registrul de informatii pentru Windows

Voicu Adrian, grupa 431A

1.Win32 API

1.1 Introducere

Win32 API reprezinta nucleul Microsoft ce contine un set de instructiuni destinate programarii aplicatiilor disponibile in sistemul de operare Microsoft Windows. Altfel spus este o interfata programabila(API). Win32 API are o denumire echivalenta si anume: Windows API care nu este oficiala, insa denumirea este una universala permitand utilizatorului sa se gandeasca la suportul atat pentru Windows pe 32 biti, cat si pentru Windows pe 64 biti.

Folosind win32 API programatorul are permisiunea de a accesa o mare parte din functiile nivelului de baza ce apartin sistemului de operare, dandu-i acestuia posibilitatea de a crea aplicatii intr-un mod cat mai flexibil.

1.2 Windows API

Prin API (un acronim din engleza pentru Application Program Interface) intelegem un set de functii Windows predefinite folosite pentru a controla aspectul si comportamentul fiecarui element Windows.

Fiecare actiune a utilizatorului cauzeaza actiunea unor functii API ce informeaza Windows-ul ce se intampla. Cu o privire de ansamblu putem spune ca este ceva gen codul nativ al Windows-ului.

Un API este o interfata software la software, nu este o interfata utilizator. Folosind API-uri, aplicatiile comunica intre ele fara alte cunostinte sau interventii din partea utilizatorului. Utilizatorul vede decat o singura interfata in spatele careia se afla mai multe aplicatii care lucreaza impreuna utilizand API-uri.

Astazi Windows este cu siguranta cel mai utilizat sistem de operare pentru calculatoarele personale. Winapi este interfata codului sursa care este folosita pentru a crea aplicatii windows. Aceste aplicatii sunt create folosind platforma SDK(Software development kit) care contine resurse ca: unelte, librarii, documente, exemple, toate acestea ajuta programatorul sa realizeze aplicatii windows. Windows API a fost creat pentru limbajele de

programare C si C++, dar si pentru cea mai noua varianta a C-ului si anume C#. Este cea mai simpla si directa metoda de a crea aplicatii win.

Windows API are la baza patru componente:

- serviciile de baza care faciliteaza accesul la resursele fundamentale ale Windows-ului.

- GDI(Graphics Device Interface) este o interfata ce lucreaza cu grafica. Este folosita pentru a interactiona cu dispozitivele grafice.

- interfata utilizator furnizeaza functionalitati pentru a crea ferestre si contoale.

- serviciile network ofera acces la internet.

In trecut dezvoltatorii de aplicatii aveau de ales intre C si Assembler ca limbaje de programare, astazi exista o platforma ce inglobeaza mai multe astfel de limbaje. Un exemplu este platforma de la Microsoft - Microsoft Visual Studio. Aceasta platforma permite folosirea unui numar semnificativ de limbaje de programare : C#, C++, J#,VB. In zilele noastre programatorii folosesc limbajele de ultima generatie sau cu cele mai noi imbunatatii ale versiunilor precedente pentru a crea aplicatii de tip business. Daca se doreste a crea aplicatii de tip grafic sau chiar si jocuri, unde sunt necesare cantitati mari de resurse, ca limbaj de programare se foloseste C(cel mai utilizat) dar, unii programatori mai folosesc si mediul DirectX.

Am observat ca platformele ce furnizeaza posibilitatea de a alege intre diferite limbaje de programare este foarte populara si in acelasi timp foarte folosita. Exista cateva librarii care ajuta programatorii sa creeze aplicatii ce pot fi rulate pe toate sistemele de operare: Windows, Linux, BSD si MAC OS. Ca exemplu de librarii am sa amintesc pe cele mai cunoscute: Qt, Swing sau wxWidgets.

Limbajele C si C++ pot folosi Windows API direct, in timp ce alte limbaje folosesc librarii intermediare. Aceste librarii sunt si ele create in C sau C++ pentru a permite celorlalte limbaje de programare sa foloseasca Windows API indirect.

1.3Versiuni

Aproape orice noua versiune de la Microsoft Windows si-a introdus propriile modificari pentru Windows API .Totusi, acronimul API a ramas neschimbat in timpul tranzitiilor de la o versiune la alta a sistemului de operare Windows. In cele din urma Microsoft a modificat numele curent al familiei Win32 API in Windows API pentru a oferi denumirii o flexibilitate mai mare in folosire.

Familia Win32 API a cunoscut patru versiuni:

-Win16. Este API-ul pentru prima versiune a pe 16 biti a Microsoft Windows. Functiile Win16 API se gasesc in fisierele de baza ale sistemului de operare: kernel.exe, user.exe si gdi.exe. Facand abstractie de extensia fisierelor acestea chiar sunt dll-uri.

-Win32. Este versiunea moderna pe 32 biti . Dll-urile de baza pentru aceasta versiune sunt: kernel.dll, user.dll si gdi.dll.

-Win32s. Este o extensie pentru familia de Windows 3.1x de la Microsoft Windows care implementeaza un subset al Win32 API pentru aceste sisteme. S vine de la subset.

-Win64. Este varianta de API implementata pe platformele de 64 biti ale arhitecturii Windows. La aceasta versiune toti pointerii sunt pe 64 de biti in mod implicit, de aceea este necesar sa se verifice compatibilitatea codului sursa .

1.4 Aplicabilitatea in .NET

In .NET putem apela Win32 API folosind platforma Interop Services care se gaseste in namespace-ul System.Runtime.InteropServices. Windows API se bazeaza pe trei dll-uri principale:

- user.dll folosit in lucrul cu interfata utilizator

- kernel.dll contine managementul memoriei si lucrul cu fisierele

- gdi.dll este folosit in elementele de grafica

Interop Services din cadrul .NET lucreaza cu dll-uri externe si anume cu cele mentionate mai sus dar nu numai. System.Runtime.InteropServices folosit in aplicatie ofera un plus de functionalitate prin posibilitatea de a apela aplicatii din exterior.

Exemplu de program simplu:

```
using System;
using System.Runtime.InteropServices;
namespace Win32Application
{
    public class Win32
    {
        [DllImport("User32.dll")]
        public static extern Int32 FindWindow(String lpClassName,String lpWindowName);
    }
}
```

Linia de cod [DllImport("User32.dll")] importa in program fisierul user32.dll dupa care poti sa beneficiezi de functiile lui. Urmatoarea linie de cod foloseste Find Window, una din functionalitatile fisierului user32.dll. Aceasta functie localizeaza fereastra de nivel inalt in ordine Z care are numele clasei si titlul date ca parametrii ai functiei. Aplicatia ce poate folosi aceasta functie poate specifica o clasa particulara, un titlu particular, amandoua sau niciuna.

Trebuie mentionat faptul ca inainte de utilizarea oricarei functii externe in .NET trebuie declarata functia in program. In linia de cod de mai sus se incearca sa se foloseasca

functia FindWindow din cadrul user32.dll.while folosind cuvantul cheie extern care indica faptul ca functia este declarata ca fiind una externa.

Cateva functii importante ale dll-ului user32.dll:

- FindWindow
- EnumChildWindows
- EnumWindows
- GetDesktopWindow
- SetForegroundWindow
- SendMessage
- GetWindowText
- GetWindowTextLength
- GetWindow

Am mentionat mai sus functiile principale si celor mai utilizate din user32.dll, iar pentru a avea o viziune clara asupra utilizarii acestor functii o sa evidentiez printr-un exemplu:

```
using System;
using System.Text;
using System.Runtime.InteropServices;
namespace Win32Application
{
    public class Win32
    {
        [DllImport("User32.dll")]
        public static extern Int32 FindWindow(String lpClassName,String lpWindowName);
        [DllImport("User32.dll")]
        public static extern Int32 SetForegroundWindow(int hWnd);
        [DllImport("User32.dll")]
        public static extern Boolean EnumChildWindows(int hWndParent,Delegate
        lpEnumFunc,int lParam);
        [DllImport("User32.dll")]
        public static extern Int32 GetWindowText(int hWnd,StringBuilder s,int nMaxCount);
        [DllImport("User32.dll")]
        public static extern Int32 GetWindowTextLength(int hWnd);
        [DllImport("user32.dll", EntryPoint="GetDesktopWindow")]
        public static extern int GetDesktopWindow();
    }
}
```

2.Registrul de informatii pentru Windows

2.1 Structura

Registrul este o baza de date ierarhizata care contine date critice pentru functionarea sistemului de operare Windows precum si functionarea aplicatiilor si serviciilor care ruleaza

sub Windows. Datele sunt structurate in trei formate. Fiecare nod al arborelui se numeste cheie("key" in engleza). Fiecare cheie contine la randul sau subchei care se numesc valori("values" in engleza).

Cateodata, prezenta unei chei este suficient pentru o aplicatie, alte ori aplicatia are nevoie sa deschida o cheie si sa utilizeze valorile asociate ei. Numele cheilor nu sunt case sensitive si nu pot contine caracterul "\". Numele valorilor si a datelor pot contine caracterul "\".Numele fiecărei subchei este unic.

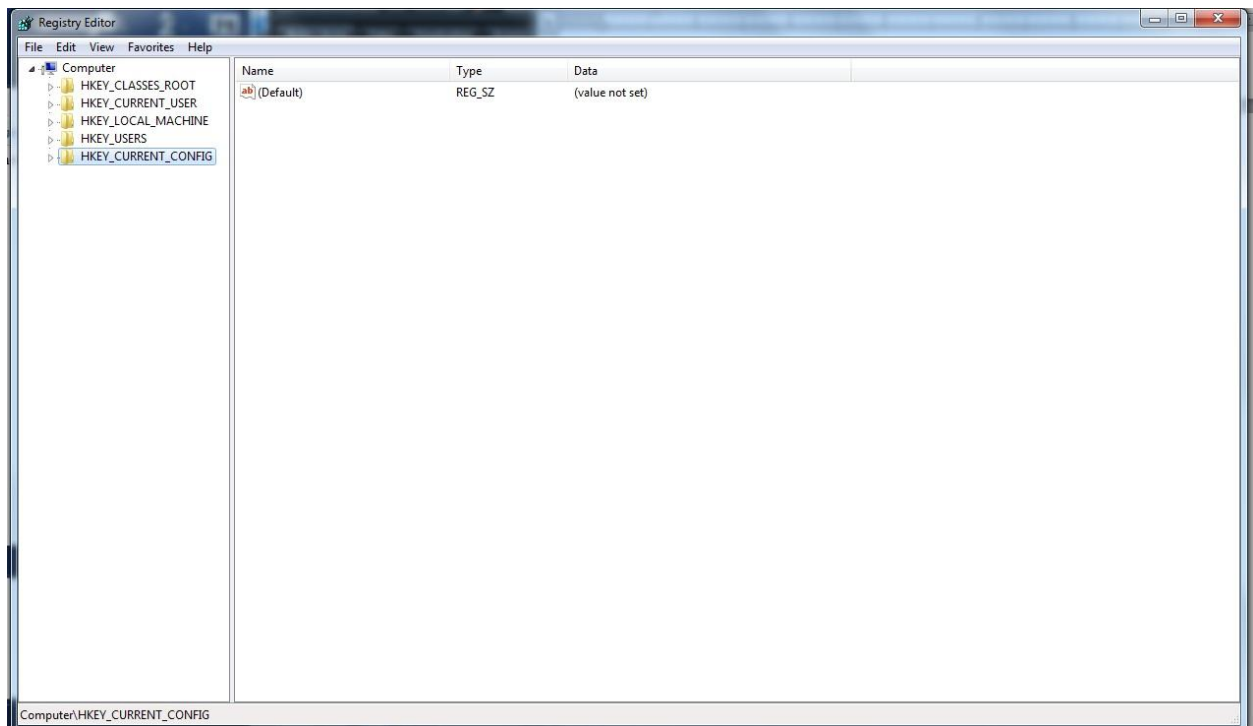


Figura 1

Figura 1 ilustreaza structura registrelor cheie folosind un editor numit regedit.exe.

Dupa cum se poate observa si in imagine Computer are un set de arbori care, fiecare reprezinta cate o cheie. Principalii arbori prezenti in imagine sunt:

- HKEY_CLASSES_ROOT
- HKEY_CURRENT_USER
- HKEY_LOCAL_MACHINE
- HKEY_USERS
- HKEY_CURRENT_CONFIG

HKEY_LOCAL_MACHINE are urmatoarele subchei:

- HARDWARE

-SAM

- SECURITY

-SOFTWARE

-SYSTEM

Fiecare subcheie de mai sus are la randui ei alte subchei. De exemplu cheia HARDWARE are urmatoarele subchei:

-DESCRIPTION

-DESCRIPTION

-RESOURCEMAP

Aruncand o privire prin aceste registre se observa ca nu gasim doua register care sa fie la fel. Deoarece registrul stocheaza date despre hardware-ul si software-ul care sunt instalate sau disponibile in calculatorul tau, valorile multor intrari de registre sunt specifice calculatorului si configuratiei sub care lucreaza. In plus, unele continuturi de registre se creeaza de fiecare data cand sistemul de operare este pornit sau log on. Drept rezultat, multe schimbari ale registrelor sunt pierdute atunci cand utilizam butonul de restart, sau cand folosim log off apoi log on iar.

Datorita diversitatii programelor, este foarte greu sa prezici cum un program anume poate sa interpreteze datele din registre. Este important de retinut ca registrele apartin programelor, si nu utilizatorilor. De aceea, daca utilizatorul nu este familiar cu programul sau cu intrarea registrului, este indicat sa nu se modifice continutul acestora sau sa se efectueze alte operatii asupra lor cum ar fi delete.

Referinte Bibliografice:

-<http://zetcode.com/gui/winapi/introduction/>

-http://en.wikipedia.org/wiki/Windows_API

-<http://msdn.microsoft.com>

-<http://technet.microsoft.com>