Structura sistemelor de Operare Windows si Linux

- Zăpuc Gabriela, Grupa 433A

1. Structurile de bază ale fiecărui sistem de operare în parte: concepte generale, structura nucleului
2. Nivelul de abstractizare al hard-ului
3. Interpretoarele de comenzi din Linux

- *Ilie Marian Danut, Grupa 433A*

4. Comparaţie interfeţe
    4.1 GNOME
    4.2 KDE
    4.3 Interfaţa în mod text
5. Biblioteca apelurilor de sistem pentru Linux

- *Vlad Adina Madalina, Grupa 433A*

6. Win32 API şi registru de informaţii pentru Windows
    6.1 Win32 API
    6.2 Interactia intre programe
7. Subsistemul POSIX şi interfaţa de proces Win32
    7.1 POSIX
    7.2 Interfaţa de proces Win32

Zăpuc Gabriela, Grupa 433A

1. Structurile de bază ale fiecărui sistem de operare în parte: concepte generale, structura nucleului
2. Nivelul de abstractizare al hard-ului
3. Interpretoarele de comenzi din Linux

1. Structurile de bază ale fiecărui sistem de operare în parte: concepte generale, structura nucleului

Inainte de a vorbi de structurile de baza ale fiecarui sistem de operare in parte, trebuie inteles in primul rand conceptul de sistem de operare si care este rolul acestuia intr-un calculator. Astfel, un sistem de operare este un program conceput pentru a putea lucra alte programe pe un calculator, fiind unul din cele mai importante programe. Acest program este considerat “coloana vertebrala” a unui calculator, fiind responsabil in gestionarea ambelor resure hardware si software.

Sistemele de operare au responsabilitatea a tot ceea ce se intampla in calculator, in ceea ce priveste controlul si alocarea de memorie in recunoasterea intrarii pentru dispozitive externe si in transmiterea iesirii pentru display-ul calculatorului. Sistemele de operare au rol important, de asemenea, ingestionarea fisierelor de pe hard-urile calculatoarelor si perifericele de control, cum ar fi de exemplu scaner-ele sau imprimantele.

Sistemul de operare **Linux** este unul din cele mai cunoscute sisteme de operare, un sistem de operare sigur in conformitate cu kernel. Kernel-ului este un program care constituie nucleul central al unui sistem de operare calculator. Acesta are control complet asupra tot ceea ce are loc in sistem. Un nucleu poate fi contrastat cu un shell (cum ar fi bash, csh sau ksh în sistemele de operare Unix-like), care este parte exterioară a unui sistem de operare şi un program care interacţionează cu comenzile utilizatorului. Kernel-ului în sine nu interacţionează direct cu utilizatorul, ci interacţionează cu shell-ul şi alte programe, precum şi cu dispozitivele hardware de pe sistem, inclusiv procesor (numit, de asemenea, unitatea centrală de prelucrare sau CPU), memorie şi hard disc.

Sistemul de operare Linux este o clona a sistemului de operare UNIX, Linux fiind alcatuit din mai multe sisteme de operare cum ar fi Fedora, Gentao, RedHat. Kernel-ul este nucleul sistemului de operare Linux. Sistemele de operare Linux au cam acelesi avantaje si dezavantaje, datorita faptului ca folosesc acelasi nucleu Kernel.

Sistemul de operare Linux are avantaje in ceea ce priveste securitatea sistemului, instalarea software-ului, stabilitatea sistemului si compatibilitate hardware.

Cel mai semnificativ avantaj al sistemului de operare Linux il reprezinta faptul ca acesta are abilitatea de a executa, in continuare, chair si dupa dupa"defectarea" unui program, adica daca o aplicatie se blocheaza, aceasta nu afecteaza Kernel-ul sau alte programe(procese). Atunci cand aplicatia s-a blocat trebuie sa ne asiguram ca aceasta nu o sa isi mai revina (adica ca aplicatia s-a blocat complet). Aceasta capacitate Linux in cazul de acidentare, sa nu afecteze kernel-ul, ne demonstreaza faptul ca sistemul de operare Linux este un sistem de operare stabil, ceea ce inseamna foarte mult pentru majoritatea utilizatorilor(a oamnenilor).

Un alt avantaj, dupa cum am spus si anterior, il reprezinta securitatea. Acest sistem de operare, Linux, a fost proiectat cu gandul de a face un sistem de operare sigur(securizat). Codul sursa Linux este ceea ce face diferenta dintre Linux si alte sisteme de operare, facand din Linux un sistem de operare sigur(codul se afla in domeniul public, rezolvandu-se, astfel, rapid majoritatea problemelor de securitate). Utilizatorii au posibilitatea de a vedea modificarile ce urmeaza sa fie facute in cod, singura data cand un utilizator este administratorul sistemului de operare este atunci cand ceva este modificat in sistem. Rularea cu privilegii de utilizator are posibilitatea(avantajul) de a opri toate virusurile si instalarile redate de spy-ware si add-ware. Prevenirea virusurilor se realizeaza si prin faptul ca niciun sistem Linux nu este la fel, deoarece Linux este un sistem de operare personalizat, adica fiecare Linux instalat are un design unic.

Linux are insa scapari cu referire la suportul softwere, la suportul hardwew si la user knowlage.

In cazul suportului softwere, care reprezinta si unul din cele mai mari dezavantaje ne demonstreaza ca sistemul de operare Linux este un sistem bun, insa ii ia destul timp pentru a intelege cum merg lucrurile in sistemul de operare. Greutatea pentru utilizatori de a folosi un sistem de operare necunoscut.

Multi spun ca Linux este un sistem de operare mai bun decat Windows. In ceea ce priveste pretul, stabilitatea este adevarat si decizia este foarte clara. Ambele sunt sisteme de operare. Linux de cele mai multe ori este oferit gratis, in timp ce Windows a crescut in pret de-a lungul anilor.

Luand in vedere stabilitatea sistemului de operare Windows, acesta este cunoscut sub numele de "ecranul albastru" din cauza blocarii sistemului, pe cand Linux in mod normal nu se blocheaza, doar daca seterile au fost facute incorect.

O alta diferenta Windows/Linux se refera la factorul de repornire al sistemului de operare. Un motiv pentru care Windows trebuie restartat se face, deoarece Windows lucreaza in Windows kernel. Linux nu trebuie repornit, deoarece modul sistemului de operare ramane in tact si nu afecteaza cererile kernel-ului. Capacitatea de a rula o perioada mai lunga reduce timpi morti

economisandu-se astfel bani. Faptul ca Windows ruleaza de ani de zile ii confera un avantaj semnificativ, peste Linux.

Kernelul Windows-ului are acces complet la hardware şi la resursele de sistem ale calculatorului şi rulează cod într-o zonă de memorie protejată. Aceasta controlează accesul la programarea., prioritizarea thread-urilor, gestiunea memoriei şi interacţiunea cu hardware-ul. Modul kernel opreşte serviciile modului utilizator şi aplicaţiile să acceseze zonele critice ale sistemului de operare la care acestea nu ar trebui să aibă acces. Procesele modului utilizator trebuie să intrebe modul kernel pentru a efectua aceste operaţiuni în numele lor.

2. Nivelul de abstractizare al hard-ului

Nivelul de abstractizare, prescurtat HAL(Hardware Abstraction Layer) reprezinta o modalitate de a ascundere a detaliilor de implementare ale unui anumit set de functionalitati particulare. La inceput, 'Linux' reprezenta numele nucleului. Numele de nucleu se refera la o aplicatie de sistem de jos nivel ("low-level"->nivel scazut) care formeaza nivelul de abstractizare a componentelor hardware , controleaza accesul la hard si la sistemulde fisier, permite lucrul multitasking, accesul distribuit si la reteaua locala, constrangerile de securitate informatica.

HAL este un nivel software aplicat la hardware-ul calculatorului, de obicei calculator desktop, caruia ii perimite sa localizeze si sa utilizeze dispozitive hardware cum ar fi imprimanta, scanner etc. HAL este un spatiu de utilizare software ce mentine o lista de cu proprietati bine definite pentru fiecare dispozitiv in parte.

Una dintre principalele sale (cum ar fi placa de baza, unităţi, şi plăci video, de exemplu.) funcţii este de a ascunde diferenţele dintre sistemul informatic hardware de la o mare parte din nucleul sistemului de operare. In acest fel, codurile se executa in kernel-mode si nu trebuie sa fie modificate pentru a rula pe hardware diferit. Cu privire la un calculator personal Microsoft Windows, HAL poate fi considerata ca fiind driverul placii de baza, lasand instrucţiunile din toate limbajele de calculator de nivel superior "sa vorbeasca" cu componente de nivel inferior, cum ar fi hardware-ul sistemului.

Portabilitatea a Windows NT kernel-mode code permite accesul la procesoare proiectate diferit, cu arhitecturi diferite ale unitatii de management al memoriei. De asemenea, permite sistemului sa aiba de controlul unor sisteme cu diferite arhitecturi de intrare/ iesire. Linux O / S are capacitatea de a introduce un HAL în timp ce se executa.

Sistemul de operare Windows NT are un HAL in spatiul kernel-ului, intre hardware si kernel-ului, drivere, servicii executive. Acest lucru permite portabilitatea a Windows NT kernel-mode code sa varieze pentru diferite procesoare , cu arhitecturi unitare de management de memorie diferite si o varietate de sisteme cu arhitecturi diferite de I/O bus; majoritatea codurilor se executa fara schimbarea pe aceste sisteme, atunci când sunt compilate pentru set de instructiuni pentru aceste sisteme. De exemplu, SGI Intel x86 bazate pe statiile de lucru nu au fost IBM, statii de lucru PC-uri compatibile, dar, datorită HAL, Windows NT a fost capabil să ruleze pe ele. Windows Vista si mai târziu (Windows Server 2008 şi mai târziu pentru servere), detecteaza automat care nivel de abstractizare ar trebui folosit pentru momentul de boot.

Hal.dll reprezinta Windows Hardware Abstraction Layer. HAL implementeaza o serie de funcţii care sunt puse în aplicare în diferite feluri de catre platforme hardware diferite, care in acest context, se refera mai ales la chipset. Alte componente în sistemul de operare pot apela apoi aceste functii în acelasi fel pe toate platformele, fara a tine seama de actuala implementare.

3. ***Interpretoare de comenzi***

In sistemul de operare Linux exista mai multe interpretoare de comenzi(numite shell-uri) disponibile : bash(Bourne Again Shell), tesh(C Shell->csh), zsh, ksh (Korn Shell).
Un interpretor de comenzi permite unui utilizator să execute comenzi prin tastarea lor manual la un terminal sau în mod automat în programe numite scripturi de shell. Shell este singura cale prin care utilizatorul poate comunica cu Kernel-ul cu privire la: programele care trebuie exexutate, cine anume sa le execute, ce sa faca iesirea, comunicarea a ceea ce doreste utilizatorul catre Kernel si alte servici utilizator. Shell primeste anumite comenzi de la utilizaor, le decodifica dupa care dorintele acestuia sunt comunicate la Kernel, deci toate comunicatiile dintre nucleul Kernel si utilizator sunt transmise prin shell.

BASH = **B**ourne **A**gain **Sh**ell

Bash este un intrepretor de comenzi (shell)scris ca un inlocuitor gratuit pentru standard Bourne Shell (/ bin / sh) realizat de Steve Bourne pentru sistemele UNIX. Acesta are toate caracteristicile originale Bourne Shell, plus adăugiri care fac mai uşor programul si utilizat de la linia de comandă.
Deoarece este software-ul liber, a fost adoptat ca shell-ul implicit pe majoritatea sistemelor Linux.

Diferente Bash vs. DOS:

- In Linux/UNIX comenzile si numele fisierelor sunt „case sensitive”, adica daca scriem „LINUX” in loc sa scriem „linux” este o greseala.
- In DOS, forward-slash „/” este comanda argument delimitatoare, pe cand backslash „\” este un director de separare. In Linux/UNIX „/” este directorul separator and „\” debarasator de caractere.
- In familia DOS se foloseste „opt punct trei” conventie filename, in sensul ca toate fisierele urmeaza un format care permite pana la 8 caractere in filename, urmata de o optiune extinsa(„punct”), de pana la 3 caractere lungime(de exemplu FILENAME.TXT). In UNIX/Linux nu exista asemenea extensi de fisiere. Perioadele pot fi plasate in orice parte a numelui fisierului si „extensiile” pot fi interpretate diferit de catre toate programele, sau nu pot fi deloc interpretate.

Shell C (csh)

C Shell este un shell dezvoltat de Bill Joy de la Universitatea din California, la Berkeley. Shell C este un procesor de comanda care lucreaza intr-o fereastra de text tipica, permitandu-I utilizatorului sa tasteze comezi care porduc actiuni. Shell C poat citi, de asemenea, comenzile dintr-un fisier, numit script. Ca toate shell-urile Unix, aceasta suporta fisierele wildcarding, conducte,documente, substituirea de comenzi, variabile si structuri de control pentru starea de testare şi de repetare. Ceea ce diferenţiază shell C, în special în anii 1980, au fost caracteristicile sale interactive şi stilul general. Caracteristicile sale noi, l-au mai uşor şi mai rapid de folosit. Stilul general al limbii este mult asemanator cu C, şi a fost văzut ca fiind mai uşor de citit. Pe multe sisteme, cum ar fi Mac OS X şi Linux, Red Hat CSH este de fapt TCSH, o versiune îmbunătăţită a CSH. Un fişier care conţine executabil tcsh are legături la ambele ca \ "csh \" si \ "tcsh \", astfel încât să fie numele se referă la aceeaşi versiune imbunatatita a shell-C.

Ksh (Korn Shell)

Linux foloseşte, în general, bash, Bourne Again shell. Inainte de a trece de la acesta la Korn Shell, se ia in considerare ce am putea vrea sa schimbam la shell. Binar Korn este mai mica decat binar bash, si multe funcţii (cum ar fi echo şi getopts) sunt construite mai degraba in shell decat executabile separate, ceea ce înseamnă că Korn utilizează mai puţină memorie pentru a rula mai repede.

Korn Shell este derivat din sh, si are anumite propritati ale chs-ului, astfel avem o posibilitate de utilizare a variabilelor de structurare asemanator csh. In plus, Korn este complet compatibil cu bash. Aceasta inseamna ca daca nu am folosi niciodata functionalitatile Korn, putem utilize in continuare comenzile pe care deja le utilizam. In plus , Korn ofera caracteristici utile, cum ar fi construirea expresiilor aritmetice, a variabilelor, functiilor combinate de disciplina si coprocese, pe care le va acoperi intr-o clipa. Korn Shell are deasemenea si un grad mare de portabilitate

Z Shell (zsh)

Z Shell (zsh) este un shell Unix, care poate fi folosit ca un shell interactive, cat şi ca un interpretor de comenzi puternic pentru shell scripting. Zsh poate fi gandit ca o extensie a Bourne shell, cu un numar mare de imbunatatiri, inclusiv unele caracteristici ale bash, ksh, şi tcsh.

Cateva aracteristici:

- programabila in linie de comanda, completare care poate ajuta utilizator sa scrie ambele opţiuni şi argumente pentru comenzile cele mai utilizate, cu suportul out-of-the-box pentru mai multe sute de comenzi;
- Expandarea de fişier extinsă permite specificarea fisierului fara a fi nevoie de a rula un program extern, cum ar fi gasit;
- Imbunatatirea variabileleor/array de manipulare;
- Editarea comenzilor multi-line intr-un singur buffer;
- Corectie gramaticala;

Ilie Marian Danut(433A)

4. Comparatie Interfete

O interfata este un instrument care permite comunicarea intre un sistem de operare si un operator. Acest instrument poate fi de natura hardware sau software. In primul rand vom realiza o clasificare a interfetelor ca mai apoi sa putem observa care sunt avantajele si dezavantajele diferitelor sisteme de operare in ceea ce priveste interfata cu utilizatorul.

Tipuri de interfete cu utilizatorul:

Interfete hardware: De exemplu, tastatura unui mic calculator de buzunar.

Interfete software: sisteme de programe care initiaza si intretin un dialog cu utilizatorul calculatorului, pentru utilizarea si/ sau configurarea acestuia. Ele reprezinta totalitatea interfetelor cu utilizatorul incluse in sistemul de operare. Exemple:

Monitoruleste un program stocat in memoria ROM interna, care se lanseaza automat cand calculatorul este pornit, si ii permite utilizatorului sa efectueze operatii simple asupra sistemului de calcul, cum ar fi: vizualizarea si alterarea continutului memoriei sau inspectarea si modificarea registrelor procesorului.

Interfete in linie de comandasunt reprezentate de un program numit interpretor de comenzi care afiseaza o fereastra in care primeste comanda si o executa.

Interfetele grafice sunt cele mai populare interfete si reprezinta un set de obiecte grafice. Practic sunt niste suprafete dreptunghiulare prin care utilizatorul comunica cu sistemul de operare.

Avantaje si dezavantaje

Se prezinta avantajele si dezavantajele interfetei grafice si a interfetei in linie de comanda:

Avantaje:

Interfata in linie de comanda: este flexibila in utilizare, comenzile sunt explicite si clare, iar parametrii sunt bine definiti. Comunicarea sa cu sistemul de operare este rapida.

Interfata grafica: este usor de folosit de aceea poate fi utilizata de neprofesionisti. Se pot creea si utiliza aplicatii complexe.

Dezavantaje:

Interfata in linie de comanda: este greu de utilizat de neprofesionisti deoarece comenzile si efectele lor trebuiesc cunoscute foarte bine.

Interfata grafica: nu sunt accesibile unele operatii de configurarea sistemului din meniuri si ferestrele interfetei grafice. Sunt folosite resurse multe si este mai putin flexibila decat cealalta.

4.1 Gnome

Gnome este un mediu desktop pus la dispozitie gratuit si se foloseste pentru sistemele compatibile UNIX. Este un sistem simplu, se bazeaza pe un ghid de design strict care asigura ca aplicatiile functioneaza la fel. Pentru ca functionalitatea sa fie mare s-au eiliminat sau comprimate multe optiuni care s-au dovedit nefolositoare. Orice modificare facuta in setari se aplica imediat si nu mai este nevoie sa se apese un buton de salvare.

Datorita mentalitatii de functionalitate cu minimum de efort nucleul sistemului are incluse foarte putine programe: Nautilus este un navigator de fisiere de sistem, Metacity este un manager de ferestre, Epiphani este un browser web etc.

Gnome se poate configura in intregime: pozitionarea continutului, aspectul ferestrelor, meniului, dialogurilor, tema. Sistemul dispune de o foarte mare accesibilitate, oferind suport pentru limbi multiple, cititor de ecran, o ‘lupa’, control vocal, configuratoare specific de tastatura etc.

4.2 KDE

KDE=K Desktop Environement este tot un mediu desktop pus la dispozitie gratuit pentru sistemele compatibile UNIX. El este facut pentru ca sa ofere o interfata care se utilizeaza si adapteaza usor nevoilor personale ale utilizatorului. El este un standard pe distributiile : Mandriva, SUSE, Knoppix, Slackwave, Kubuntu.

KDE poate rula pe Linux BSD, Solaris, HP-UX, AIX si alte sisteme compatibile UNIX. Majoritatea softurilor KDE utilizeazaframework-ul Qt care ruleaza atat pe majoritatea sistemelor UNIX cat si pe MAC OS X si Windows.

4.3 Interfata in mod text

Interfata in mod text este un mediu alfanumeric. Pentru a da instructiuni sistemului de operare, utilizatorul introduce comenzi de la tastatura. Sisteme de operare ale PC-urilor au interfete cu linii de comanda, care permit introducerea comenzilor de la tastatura intr-o linie de comanda ce contine un prompter. Prompterul ii arata utilizatorului ca sistemul accepta comenzi sau instructiuni. Pentru a fi preluata comanda scrisa, utilizatorul trebuie sa apese Enter si astfel se prelucreaza acea comanda.

Exemple de interfete in mod text: MS-DOS, UNIX.

MS-DOS-ul este un sistem de operare produs de Microsoft pentru platform x86. Este un sistem de operare in mediu alfanumeric. El are o serie de biblioteci de instructiuni care stau la baza acceptarii si prelucrarii instructiunilor scrise de catre utilizator.

MS-DOS este unul dintre cele mai populare sisteme de operare deoarece este foarte simplu si ca exista o varietate foarte mare de programe care pot fi scrise pentru el. Cei care l-au creat s-au inspirat din cele doua sisteme de operare care erau populare in acele vremuri: CP/M si UNIX.

Sistemul de operare MS-DOS are urmatoarele caracteristici:

- Este un sistem de operare la care nu poate lucra decat un singur utilizator si la care nu ruleaza mai multe programe in acelasi timp;
- Poate detecta erori prin procedee avansate;
- Atribuie un nume pentru fiecare dispozitiv de intrare/iesire;
- Poate face sa ruleze acelasi program pentru mai multe adrese;
- Fisierele sunt intr-o structura eficienta;
- Retine data si timpul;

5. Biblioteca apelurilor de sistem pentru Linux

Un sistem de operare pune la dispozitia programatorului servicii de acces la resursele hardware si software care sunt gestionate de sistem. Aceste servicii sunt cunoscute ca apeluri de sistem.

Exemple de servicii: lucrul cu discurile, tastatura, dispozitivul de afisare, fisiere, directoare etc.

Deoarece operatiile care pot fi facute asupra resurselor sunt doar operatii simple care au putine facilitati, exista in bibliotecile specifice limbajelor de programare functii complexe care se ocupa de gestionarea resurselor respective. Aceste functii se numesc functii de biblioteca.

In urmatoarele randuri se prezinta functiile sistem puse la dispozitie pentru lucrul cu fisiere de catre sistemele de operare MS-DOS si UNIX.

Functiile acestea vor fi scrise in limbajul de programare C deoarece se cunosc caracteristicile si facilitatile sale.

Apelul *sistem open*- deschiderea unui fisier:

```
#include <sys/types.h>

#include <sys/stat.h>

#include <fcntl.h>

int open (const char *pathname, int oflag, …/*, mode_t mode*/ );
```

Daca apelul se executa fara eroare el returneaza o valoare ne-negativa.Daca la executie apare o eroare se returneaza -1.

‘pathname’ - nume de cale al fisierului.

‘oflag’-indica optiunile de deschidere.

Apelul *sistem creat*- crearea unui nou fisier:

```
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
int creat (const char *pathname, mode_t mode);
```

Apelul *sistem close*- inchiderea unui fisier:

```
#include <unistd.h>
int close(int filedes);
```

Apelul returneaza 0 in caz de succes si -1 in caz de eroare.

Apelul *sistem lseek*- prelucrare sau deplasament curent:

```
#include <sys/types.h>
#include <unistd.h>
off_t lseek(int filedes, off_t offset, int whence);
```

Apelul *sistem seek:*

Este apelul care returneaza noua valoare a deplasamentului iar in cazul in care apare o eroare se returneaza -1.

Apelul *sistem read*- citirea datelor dintr-un fisier:

```
#include <unistd.h>
```

ssize_t read(int filedes, void *buf, size_t nbytes);

Apelul returneaza numarul de octeti cititi din fisier si -1 in caz de eroare.

‚filedes'- descriptorul de fisier din care se face citirea;

‚buf'- adresa de memorie unde se salveaza ce s-a citit;

‚nbytes'- numarul de octeti care se doresc a fi cititi;

Apelul *sistem write*- scrierea datelor:

#include <unistd.h>

size_t write(int filedes, const void *buf, size_t nbytes);

Apelul returneaza numarul de octeti scrisi sau -1 in caz de eroare.

La apelul sistem write pot aparea erori:

- Se umple dispozitivul in care se scrie;
- Unele implementari UNIX nu permit sa se depaseasca o anumita dimensiune a fisierului.

Apelul *sistem times:*

Acest apel poate returna: valoarea timpului de sistem, timpul utilizat in regim utilizator, timpul utilizat in mod supervisor.

Apelul *sistem stat:*

Apelul sistem stat returneaza 0 in caz de success si -1 in caz de eroare.

'stat' – returneaza informatii despre un fisier indicat;

'fstat' – pentru fisiere deschise;

'lstat' – se returneaza ca si la 'stat' daca fisierul este legatura simbolica se returneaza informatii despre acest fisier.

Apelul *sistem umask*- controleaza valoarea mastii de acces:

#include <sys/types.h>

```
#include <sys/stat.h>

mode_t umask(mode_t cmask);
```

Apelul *sistem fork()*- proces fiu:

Apelul sistem fork() creeaza un proces fiu si mosteneste tabela de descarcare a fisierelor.

Apelul *sistem exit():*

Acest apel incheie procesul curent.

Functii de stergere si creare a unui director:

```
int mkdir(const char *pathname, mode_t mode)  //creare

int rmdir(const char *pathname)  //stergere
```

Vlad Adina Madalina(433A)

6. Win32 API si registrul de informatii pentru Windows

6.1Win 32 API

Windows API este o interfata folosita pentru crearea de aplicatii in Microsoft Windows . Cunoscuta , in general ca Win32 API (dar suport si pentru Windows 64-biti), prin intermediul ei programatorul are acces diret la mare parte a functiilor de nivel jos ale OS . Interfata de programare API permite programatorului sa creeze o interfata grafica pentru propriile aplicatii , sa acceseze memoria , dispozitive , sa inglobeze poze , sunete , video sau sa primeaca/trimita mesaje catre ferestre sau controale(Button, Text, CheckBox).

Functionarea unei aplicatii Win 32 presupune prelucrarea si traducerea mesajelor intr-un While iar apoi trimiterea lor catre fereastra activa.

6.2 Interactia intre programe

Interfata API poate fi descrisa in cadrul unei comunicari implicit intre sistemul de operare si o cerere si explicit intre mai multe aplicatii din cadrul Windows OS. Pentru ca aceasta comunicare intre aplicatii sa aiba loc au fost dezvoltate mai multe tehnologii :

- DDE (Dynamic data exchange) a fost folosit in 1987 odata cu aparitia Windows 2000 , ca o solutie pentru comunicarea intre aplicatii , sau pentru controlarea unei aplicatii de catre o alta.
- OLE (Object Linking and Embedding) dezvoltat in 1990 ca urmasul lui DDE , permite unei aplicatii de editare sa exporte o parte dintr-un document unei alte aplicatii si sa o importe ulterior cu modificarile aferente.

- COM(Component Object Model) permite obiectelor create intr-un mediu sa fie folosite in alte medii de programare
- Controale AtiveX
- NET Framework include o librarie imensa care asigura sincronizarea intre limbaje (un program poate folosi linii de cod din alte limbaje de programare).

Se pot defini 8 categorii prin care se poate reda functionalitatea Windows API

1. Servicii de baza

Asigura accesul la resurse ca : sisteme de fisiere, dispozitive, procese si fire, si tratarea erorilor, functii pe care le gasim in kernel.exe, krnl286.exe sau fisiere krnl386.exe pe 16-bit Ferestre, si Kernel32.dll pe 32-bit.

2. Servicii avansate

O completare a kernel-ului include Windows registry, shutdown / pauza sistem, pornire / oprire dispozitiv/interfata grafica dar si faciliteaza afisarea de continut grafic pe monitoare si dispozitive de iesire .

Se pate gasi in gdi.exe pe 16-bit si in gdi32.dll pe32-biti .

3. Interfata dispozitive grafice

Face posibila afisarea de continut grafic la monitoare, imprimante si alte dispozitive de iesire.

Se gasestein gdi.exe sau sigdi32.dll in modul utilizator.

Suportul GDI se bazeaza pe Win32k.sys (care comunica direct cu driverul grafic.)

4. Interfata utilizator

Face posibila administrarea/crearea ferestrelor precum si a butoanelor, barelor de scroll, administrarea/primirea informatiei de la mouse si administrarea interfatei grafice.

Se gaseste in user.sau in user32.dll .

5. Biblioteca de ferestre de dialog uzuala

Ofera aplicatiilor ferestrele de dialog pentru deschiderea si salvarea de fisiere, alegerea culorii si fontului etc. Biblioteca se regaseste intr-un fisier numit commdlg.dll pe Windows pe 16 biti si in comdlg32.dll pe Windows pe 32 biti.

6. Biblioteca de control uzuala

Ofera aplicatiilor acces lalucruri avansate precum bare de stare, toolbar-uri ,tab-uri , etc.

Se gaseste incommctrl.dll sau comctl32.dll .

7. Windows shell

Permite accesul/posibilitatea de a edita functiile oferite de Shell-ul OS-ului .

Se gaseste in shell.dll sau in shell32.dll.

8. Servicii retea

Ofera acces la NetBIOS, NetDDE, RPC etc.

7. Subsistemul POSIX si interfata de proces Win32

7.1 Subsistemul POSIX

POSIX (Portable Operating System Interface) este o familie de standarde a IEEE (Institute of Electrical and Electronics Engineers) pentru asigurarea compatibilitatii intre sistemele de operare. Acest standard a fost elaborat de sute

de experti din industrie, universitati si guvern si a fost publicat intr-o serie de documente numite seria POSIX 1003.

Initial POSIX reprezenta standardul IEEE 1003.1-1988, iar acum se refera la o familie de standarde asemanatoare. Pentru standardul initial IEEE 1003.1-1988 s-a ales termenul de POSIX1, care specifica interfete de programare a aplicatiilor (API) la nivelul sursa si se facea referire la portabilitatea codului sursa.

Cea mai noua versiune de POSIX.1 a fost dezvoltata de Austin Group in 2004.

Standardul POSIX 1003 cuprinde urmatoarele elemente

- 1003.1 functii de biblioteca, apeluri de sistem
- 1003.2 shell-ul si utilitarele
- 1003.3 metode de testare
- 1003.4 timp real
- 1003.5 limbajul Ada pentru Unix
- 1003.6 securitatea
- 1003.7 administratia sistemului
- 1003.8 accesul la fisiere
- 1003.9 limbajul FORTRAN pentru Unix
- 1003.10 super-calculatoare
- 1003.12 interfetele independente de protocol
- 1003.13 real-time profiles
- 1003.15 interfetele pentru supercalculatoare
- 1003.16 limbajul C
- 1003.17 servicii de directoare
- 1003.18 POSIX standardized profile
- 1003.19 FORTRAN 90

Un standard POSIX nu se refera la apeluri de sistem, ci se refera la interfete de programare a aplicatiilor.

7.2 Interfata de proces Win32

Windows API este o interfata destianta programarii aplicatiilor pentru sistemul de operare Windows. Aceasta a aparut in mai multe versiuni: Win16 pentru versiunile pe 16 biti ale sistemelor de operare Microsoft Windows, Win32 pentru sistemele de operare pe 32 de biti si Win64 pentru sistemele de operare pe 64 de biti.

Prin Win32 API, programatorul acceseaza o mare parte a functiilor de baza ale sistemului de operare, aceste functii fiind impartite in trei categorii:

- functii USER aflate in legatura cu obiectele GUI (Graphic User Interface) cum ar fi meniurile si butoanele
- functii GDI ce efectueaza operatii de desenare pe dispozitive precum monitoare si imprimante
- functii KERNEL ce monitorizeaza procese, threaduri, memoria partajata si fisierele.

BIBLIOGRAFIE

1. http://jalobean.itim-cj.ro/Cursuri/ArhCalc/Materiale/carte/cap4.htm
2. http://www.scribd.com/doc/14143826/Sisteme-de-operare
3. http://ro.wikipedia.org
4. http://en.wikipedia.org
5. http://stst.elia.pub.ro
6. http://en.wikipedia.org/wiki/Windows_API
7. http://win32apiforum.com/
8. http://stst.elia.pub.ro
9. **Andrew S. Tanenbaum** , "**Sisteme de Operare Moderne**"
10. http://en.wikipedia.org/wiki/Microsoft_POSIX_subsystem
11. http://www.scribd.com/doc/25325199/%E2%80%A2-a-Posix-Standard-Ieee-1003-1c-API
12. http://labs.cs.upt.ro/labs/so/html/so1.html
13. Ioan Jurca – Programarea de sistem in UNIX, Editura de Vest, Timisoara, 2004