

VALERIU MANUEL IONESCU
GRIGORE-ADRIAN IORDĂCHESCU

COMUNICAȚII, VIRTUALIZARE ȘI PROCESARE MULTIMEDIA



EDITURA UNIVERSITĂȚII DIN PITEȘTI

PITEȘTI 2015

UNIVERSITATEA DIN PITEȘTI
FACULTATEA DE ELECTRONICĂ, COMUNICAȚII ȘI
CALCULATOARE
DEPARTAMENTUL DE ELECTRONICĂ, CALCULATOARE
ȘI INGINERIE ELECTRICĂ

COMUNICAȚII, VIRTUALIZARE ȘI PROCESARE
MULTIMEDIA

VALERIU MANUEL IONESCU
GRIGORE-ADRIAN IORDĂCHESCU

Editura Universității din Pitești
Pitești 2015

Cuvânt înainte

Cartea “COMUNICAȚII, VIRTUALIZARE ȘI PROCESARE MULTIMEDIA” se adresează atât studenților cât și celor pasionați de comunicațiile multimedia în general. Înțelegerea aplicațiilor necesită noțiuni de baza privind rețelele de calculatoare, sistemele de operare și elemente de birotica.

Datele în format multimedia au fost elementul care au impulsionat evoluția comunicațiilor. Dacă pentru comunicații în mod text un venerabil modem de 56kbps era suficient, paginile web moderne inserează tot mai multe imagini, filme și alte componente interactive care necesită un trafic de nivel ridicat.

Autorii au încercat o abordare nouă a domeniului comunicațiilor multimedia, unitară, ce vizează toate etapele de procesare a conținutului multimedia, de la achiziționarea acestuia, până la transmiterea sau stocarea conținutului, în contextul tehnologiilor de comunicații moderne. Se prezintă situații și probleme pe care un utilizator de comunicații multimedia le poate întâmpina și soluții aplicative pentru rezolvarea acestora.

Orice telefon mobil recent este capabil în prezent să achiziționeze imagini având o rezoluție mare și o calitate a imaginii bună, apropiată de aceea a unei camere de fotografiat dedicate. Totuși sunt necesare mici retușări înainte ca imaginile să poată fi prezentate pe internet pentru a fi clare și de o dimensiune potrivită. Capitolele 1 – 4 tratează în detaliu aceste aspecte, de la achiziția unei imagini (folosind o cameră digitală sau un scanner), procesarea acesteia, până la astfel încât să le poată include pe o pagină a unui site web.

Odată achiziționate și prelucrate imaginile, este posibil o persoană să dăcă construirea propriul site web în care să publice imagini sau videoclipuri. Suplimentar, dacă dorește să crească securitatea conexiunilor utilizatorilor la acest site, cartea prezintă în capitolele 5 – 6 modul în care un site securizat (HTTPS) poate fi instalat și configurat și modul în care se pot controla caracteristicile de trafic la accesarea acestui site.

Fișierele video stocate pe unități de stocare externă prezintă problema modului în care vor fi accesate. Este posibil ca dispozitivul pe care dorim să redăm filmul să nu aibă suficient spațiu de stocare pentru a îl descărca înainte de a îl reda. Cartea prezintă în capitolul 7 metode prin care aceste videoclipuri pot fi transmise în timp real prin rețea către unul sau mai multe calculatoare conectate.

În final, capitolele 8 – 10, este tratat subiectul virtualizării resurselor în internet, adică transformarea resurselor hardware locale în resurse software, accesate printr-o rețea de calculatoare, în scopul reducerii costurilor și a creșterii accesibilității acestor resurse. Subiectul virtualizării este unul complex și are mai multe aspecte, printre care enumerăm virtualizarea stocării, a sistemului de operare, a rețelor, etc.

Să prezentăm câteva exemple.

Fișierele video sunt tot mai căutate pe internet pentru că pot captura mult mai multe informații decât o imagine. Dezavantajul este că ocupă și un spațiu corespunzător mai mare. Pentru a stoca multe filme sunt necesare multe dispozitive de stocare. Dispozitivele de stocare pot ocupa un spațiu semnificativ, iar dacă numărul lor crește excesiv ele nu mai pot fi conectate simultan la un singur sistem de calcul obișnuit ci necesita sisteme/servele dedicate. Dar dacă utilizatorii doresc să aibă acces la aceste resurse ca și când ar fi conectate la sistemul local? Soluția o oferă virtualizarea stocării.

Instalarea și configurarea unei rețele înseamnă de obicei instalarea de fire care leagă echipamente de rețea și configurarea acestor echipamente. Dar dacă dorim să modificăm această configurație de rețea fără să scoatem un fir? Soluția o oferă în acest caz virtualizarea rețelelor.

Anterior am arătat că va fi prezentat modul în care putem instala un server web securizat sau un echipament care modelează traficul în rețea. Cum rezolvăm însă situația în care nu avem o mașină dedicată pe care să testăm configurațiile dorite înainte de a instala serverul final? Soluția o oferă virtualizarea sistemelor de operare.

Iată câteva din situațiile la care această carte încearcă să dea răspunsul, ghidând cititorul în procesarea conținutului multimedia, configurarea propriei rețele, testarea locală a acesteia cu sisteme fizice și sisteme virtuale și în final conectarea ei la internet.

Sl. dr. ing. Ionescu Valeriu Manuel

Cuprins

Cuvânt înainte	3
Cuprins	5
Capitolul 1. Fundamentele prelucrării de imagini. Introducere înGNU Image Manipulation Program (GIMP).....	11
Obiectivul lucrării	11
Breviar teoretic	11
Meniul	11
Personalizarea interfeței.....	13
Importarea unei imagini. Spațiul culorilor	14
Crearea unei noi imagini	15
Analiza culorilor unei imagini.....	16
Salvarea și redimensionarea unei imagini	17
Formate de imagini	19
A. Formatul .bmp	19
B. Formatul .jpg.....	21
C. Formatul .png.....	21
Calibrarea rezoluției monitorului.....	23
Calibrarea culorilor	23
Desfășurarea lucrării.....	25
Capitolul 2. Prelucrarea avansată de imagini cu GIMP	27
Obiectivul lucrării	27
Breviar teoretic	27
Uneltele din Toolbox.....	27
Uneltele de selecție	27
Specificarea culorilor.....	31
Uneltele pentru pictură și desen.....	31
Unealta Text.....	35
Unealta Traseu	36

Imagini vectoriale.....	38
Straturi și Moduri de superpoziție	40
Desfășurarea lucrării.....	42
Capitolul 3. Funcții speciale și tehnici de prelucrare automată cu GIMP	43
Obiectivul lucrării	43
Breviar teoretic	43
Unelte de culoare.....	43
Balanța de culori (Color Balance).....	43
Nuanță și Saturație (Hue-Saturation).....	44
Niveluri (Levels).....	45
Reglaje automate (Auto).....	47
Filtre speciale	47
Reducerea și îmbunătățirea clarității.....	47
Detectia marginilor și convoluția cu matrici	48
Efecte artistice și speciale	50
Aplicații web.....	51
Animații gif	53
Script-Fu	53
Scrierea unui script	54
Un script de animație.....	56
Desfășurarea lucrării.....	58
Exerciții recapitulative	60
Capitolul 4. Securizarea serverelor web. Certificate self-signed. Virtual Private Network (VPN).	64
Obiectivul lucrării	64
Breviar teoretic	64
Servere web	64
Rolul certificatelor și al certificatelor auto-semnate	67
VPN (Virtual Private Network)	68
PPTP (Point to Point tunneling protocol).....	69
L2TP/IPsec (Layer 2 Tunneling Protocol)	69
IPSec (IP Security).....	69
SSL VPN (Secure Socket Layer VPN)	69

Aplicații	70
Setarea unei conexiuni VPN	70
Setare server VPN	70
Setare client VPN	71
Instalarea unui server web Apache.....	73
Generarea și utilizarea unui certificat cu semnătură proprie în Apache	74
Rolul folosirii unui certificat self-signed.....	74
Configurarea serverului Apache în Windows pentru a folosi un certificat self-signed.	74
Desfășurarea lucrării.....	77
Capitolul 5. Modelarea traficului în Linux	78
Obiectivul lucrării	78
Breviar teoretic	78
Componentele latenței în rețele de calculatoare	78
Etaple parcurse de un pachet în traversarea unui echipament de rețea cu funcții de calitate a serviciului	80
Traffic Control (tc).....	81
Discipline de plasare în coada de pachete fără clase.....	81
Discipline de plasare în coada de pachete cu clase	84
Aplicații	85
Network Emulation (netem)	85
Desfășurarea lucrării.....	87
Capitolul 6. Transmiterea fluxurilor multimedia. Port forwarding	89
Obiectivul lucrării	89
Breviar teoretic	89
Transmission Control Protocol	89
User Datagram Protocol	90
Tipuri de fluxuri.....	90
Port forwarding.....	90
iptables.....	91
Aplicații	92
Flux UDP.....	93
Configurare sursă flux UDP	94
Configurare destinație flux UDP.....	94

Flux TCP	96
Port forwarding folosind iptables	97
Specificarea unei game de adrese în filtrele iptables	98
Crearea unui server SSH.....	99
Crearea unui tunel SSH	101
Port forwarding pentru porturile locale	101
Port forwarding pentru porturile remote	102
Desfășurarea lucrării.....	102
Capitolul 7. Virtualizarea sistemelor de operare	104
Obiectivul lucrării	104
Breviar teoretic	104
Aplicații	107
Instalarea și configurarea unui sistem de virtualizare de tip 1: Microsoft Hyper-V	108
Pașii necesari pentru instalarea sistemului de operare Hyper-V Server	109
Configurare sistem Client pentru controlul de la distanță al Hyper-V Server	110
Crearea unei mașini virtuale	112
Instalarea și configurarea unui sistem de virtualizare de tip 2: VirtualBox	117
Instalare sistem de operare Linux în VirtualBox	119
Desfășurarea lucrării.....	121
Capitolul 8. Virtualizarea echipamentelor de rețea.....	122
Obiectivul lucrării	122
Breviar teoretic	122
Configurarea conectării la rețea a sistemului de operare virtual	122
NOT ATTACHED (adaptorul nu este atașat la rețea)	122
NETWORK ADDRESS TRANSLATION (NAT).....	122
BRIDGED NETWORKING	123
INTERNAL NETWORKING	123
HOST-ONLY NETWORKING.....	124
Aplicații	124
Conectarea unei mașini virtuale la rețea în modul Network Address Translation (NAT)	124
Conectarea unei mașini virtuale la rețea în modul Bridge.....	125
Desfășurarea lucrării.....	126
Capitolul 9. Securizarea rețelelor locale. VLAN. Interfețe virtuale și subinterfețe.....	128

Obiectivul lucrării	128
Breviar teoretic	128
Subinterfețe	128
Interfețe virtuale.....	129
Virtual Local Area Network (VLAN).....	129
Aplicații	132
Hardware	132
Software.....	132
Configurarea interfețelor virtuale în Linux	132
Configurarea interfețelor virtuale în Windows.....	134
Desfășurarea lucrării.....	136
Capitolul 10. Virtualizare stocare. Configurare Storage Area Network folosind protocolul iSCSI	137
Obiectivul lucrării	137
Breviar teoretic	137
Aplicații	138
Actualizarea surselor sistemului Linux.....	138
Instalare target iSCSI	138
Configurarea targetului iSCSI	138
Configurarea stocării.....	139
Crearea țintei iSCSI.....	139
Repornire target iSCSI	140
Testarea țintei iSCSI	140
Securizarea unei partiții iSCSI	142
Configurare inițiator Linux	144
Formatare partiție în Linux	145
Desfășurarea lucrării.....	146
Anexa 1 – Codul integral al filtrului de animație „Whirl”	147
Anexa 2 – Configurarea unui ruter Linux.....	149
Descărcarea și pornirea sistemului de operare	149
Sistemul de fișiere Linux	150
Execuția comenzilor	151
Fișiere importante în Linux pentru configurarea adaptorului de rețea.....	152
Configurarea setărilor de rețea.....	152

Configurarea manuală a unui ruter cu sistemul de operare Linux	153
Bibliografie	156

Capitolul 1. Fundamentele prelucrării de imagini. Introducere în GNU Image Manipulation Program (GIMP)

Obiectivul lucrării

Lucrarea își propune familiarizarea studentului cu mediul OpenSource de prelucrare a imaginilor digitale GNU Image Manipulation Program (GIMP). La sfârșitul acestei lucrări, studentul va fi capabil să importe și să salveze imagini digitale, să le reducă dimensiunile pe disc, precum și să facă o analiză completă a imaginii importate din punct de vedere a distribuției culorilor.

Breviar teoretic

Prelucrarea (sau editarea) imaginilor este procesul prin care sunt aduse modificări asupra imaginilor, fie că e vorba de fotografii tradiționale (pe film fotografic), fie de imagini digitale. Cum secolul XXI este dominat de fotografia digitală, stăpânirea tehnicilor de prelucrare numerică a imaginilor este esențială atât pentru designeri web, cât și pentru ingineri sau cercetători. Imaginile digitale apar peste tot în viața de zi cu zi, de la pagini web la documente Word sau prezentări Power Point, de la ziare până la pagini Facebook.

Există zeci de programe destinate prelucrării de imagini, dintre care cele mai cunoscute sunt Adobe Photoshop, Corel PHOTO-PAINT, GIMP (și GIMPshop), Microsoft Paint, paint.net, Picasa, Instagram, etc. În această lucrare de laborator vom folosi GIMP, un software de prelucrare a imaginilor ce oferă performanțe similare cu cele ale celor mai bune editoare (Adobe Photoshop) având marele avantaj de a fi OpenSource.

Meniul

Interfața GIMP (la origine o abreviere a numelui General Image Manipulation Program) constă dintr-un meniu (Figura 1), o fereastră mobilă cu unelte grafice (Figura 2.b) și una sau multe ferestre de dialoguri ancorabile (Figura 2.a). Meniul, reprezentat în Figura 1, are următoarele principale funcționalități:

File – Crează noi imagini (*New*), Deschide imaginile sau proiectele GIMP salvate anterior (*Open*), Salvează un nou proiect GIMP (*Save*), Salvează proiectul într-un format bitmap (*Export*), Inchide proiectul (*Close*), Iese din program (*Quit*).

Edit – Anulează ultima modificare (*Undo*), Reface ultima modificare (*Redo*), Copiază regiunea selectată, chiar dacă face parte dintr-un strat ascuns (*Copy*), Copiază doar partea vizibilă corespunzătoare regiunii selectate (*Copy visible*), Lipește regiunea copiată (*Paste*), Lipește regiunea copiată într-un proiect sau un strat nou (*Paste as*), Setează preferințele generale ale programului GIMP (*Preferences*).

Select – Selectează toată imaginea (*All*), Anulează selecția curentă (*None*), Selectează partea complementară a selecției curente (*Invert*), Selectează pe baza similitudinii culorilor (*By Color*), Micșorează selecția curentă (*Shrink*), Mărește selecția curentă (*Grow*), Transformă selecția într-un traseu vectorial (*To Path*).

View – Poate apropia sau depărta imaginea (*Zoom*), Selectează filtre pentru diferite corecții de afișaj (*Display Filters*), Alege ce elemente vizuale ajutătoare să fie prezente în timpul lucrului dintr-o listă de opțiuni selectabile.

Image - Fixează spațiul culorilor (*Mode*), Rotește sau reflectă imaginea (*Transform*), Ajustează dimensiunea imaginii (*Scale Image*), Lipește straturile vizibile într-unul singur (*Merge visible layers*), Aplatizează imaginea prin alipirea tuturor straturilor și eliminarea canalului alpha (*Flatten image*), Afișează proprietățile imaginii (*Image properties*).

Layer – Crează un nou strat (*New layer*), Adaugă un nou strat identic cu cel selectat (*Duplicate layer*), Sterge un strat (*Delete layer*), Modifică ordinea straturilor (*Stack*), Adaugă și modifică o mască de strat (*Mask*), Modifică informațiile despre canalul alpha pentru stratul selectat (*Transparency*), Rotește sau reflectă stratul (*Transform*), Modifică frontiera stratului (*Layer boundary size*), Redimensionează conținutul stratului (*Scale layer*).

Colors - Deplasează echilibrul culorilor (*Color Balance*), Fixează nuanța predominantă a imaginii și nivelul de saturație al culorilor (*Hue-Saturation*), Transformă într-o singură nuanță (*Colorize*), Reglează luminozitatea și contrastul (*Brightness – Contrast*), Binarizează cu un prag (*Threshold*), Modifică nivelurile de intensitate pentru fiecare canal (*Levels*), Modifică distribuția intensității pentru fiecare canal în parte (*Curves*), Reduce numărul de culori (*Posterize*), Transformă imaginea în nuanțe de gri (*Desaturate*), Inversează fiecare culoare cu valoarea complementului ei (*Invert*), Reglează automat nivelurile de luminozitate sau culoarea ale fiecărui pixel în conformitate cu o anumită condiție impusă histogramei imaginii (*Auto*).

Tools – Afișează toate uneltele disponibile în Toolbox (Figura 2.b) sub formă de comenzi.

Filters – Aplică filtre matematice speciale pe imagine, precum: estompare (*Blur*), detecția marginilor (*Edge-Detect*), operatori morfologici (*Dilate* și *Erode*), convoluția cu o matrice (*Convolution Matrix*), etc. Tot din meniul *Filters* putem activa și consola *Script-Fu*.

Windows – Crează un nou Toolbox (*New toolbox*) sau activează Toolboxul existent (*Toolbox*), poate adăuga un dialog ancorabil (*Dockable Dialogs*) și poate fixa toate ferestrele mobile pe fereastra principală (prin selecția opțiunii *Single Window Mode*).

Help – Oferă informații despre versiunea curentă (*About*) și instrucțiuni despre folosirea GIMP-ului (*Help* și *User Manual*) cât și acces la pagini web importante (*GIMP Online*).

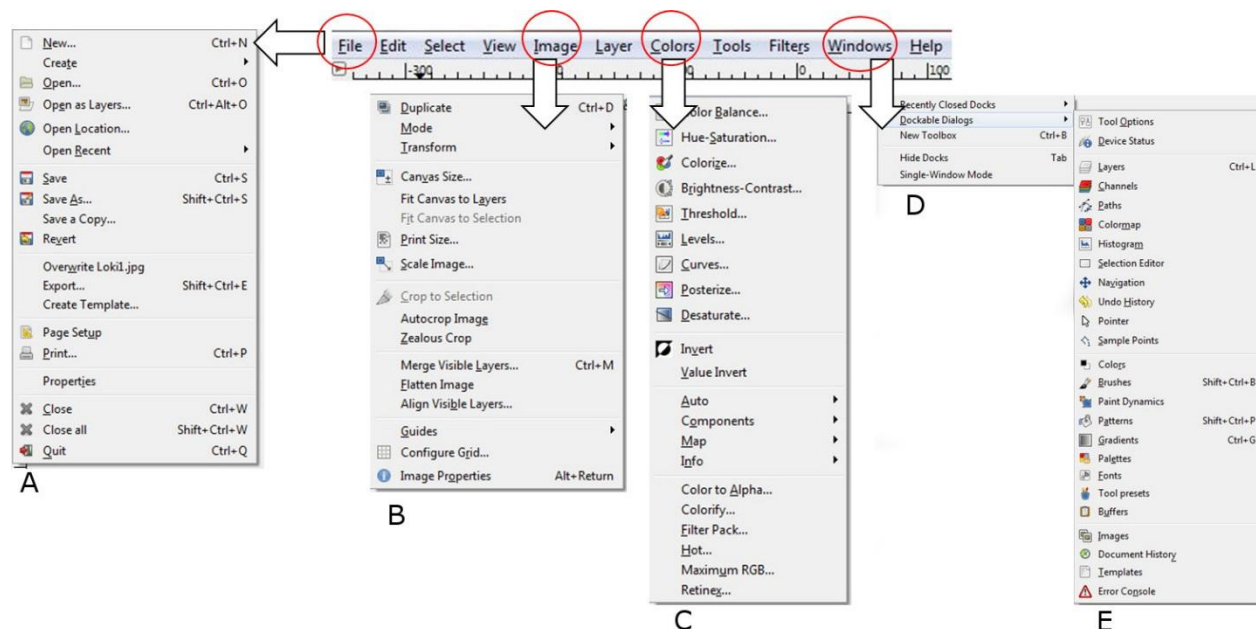


Figura 1 Meniul GIMP cu explicitarea următoarelor submeniuri: A. File, B. Image, C. Colors, D. Windows, E. Windows | Dockable Dialogs

Personalizarea interfeței

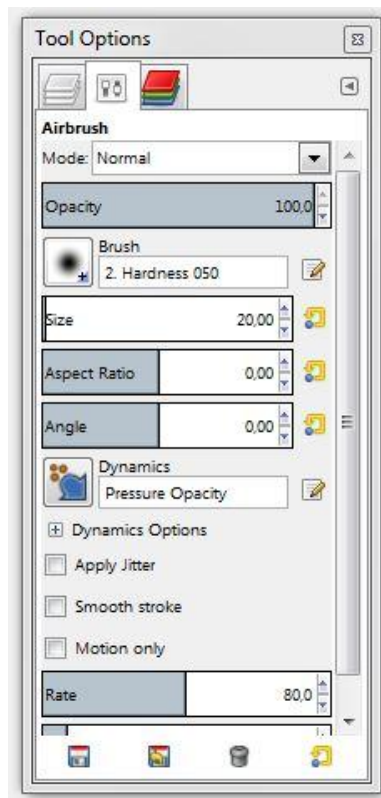
GIMP-ul oferă o mare libertate utilizatorului în a seta ce ferestre de dialog dorește să aibă tot timpul la dispoziție, precum și dispunerea lor în pagină. Toolbox este fereastra în care sunt prezentate principalele unelte grafice pe care GIMP-ul le pune la dispoziție (Figura 2.b). Dacă această fereastră mobilă nu este prezentă când deschidem pentru prima dată GIMP-ul, putem crea una nouă folosind comanda „*Windows / New toolbox*”. Alături de această fereastră mai putem observa o a doua căsuță de dialog mobilă (Figura 2a), în care putem adăuga așa zisele dialoguri ancorabile (comanda „*Windows / Dockable Dialogs*”). Putem avea oricât de multe ferestre de dialoguri ancorabile de tipul celei din Figura 2a. Putem transfera un dialog ancorabil dintr-o fereastră în alta prin metoda drag and drop. Putem translata un dialog ancorabil chiar și în partea de jos a ferestrei Toolbox prin aceeași metodă drag and drop.

Dacă dorim să legăm toate ferestrele mobile de fereastra principală (în care vom edita imaginile) putem face asta selectând opțiunea „*Windows / Single window mode*” din meniu.

Exerciții

A. Schimbați interfața GIMP încât să conțină:

- Fereastra mobilă Toolbox care să integreze în partea sa de jos dialogul ancorabil „*Tool Options*”
- O a doua fereastră mobilă în care să se găsească dialogurile ancorabile „*Channels*”, „*Layers*”, „*Paths*” și „*Undo history*”.



a.



b.

Figura 2 Interfața GIMP: a.Fereaștră mobilă cu trei dialoguri ancorabile; b. Fereaștră Toolbox

Importarea unei imagini. Spațiul culorilor

O imagine poate fi importată de GIMP fie printr-un simplu drag-drop al acesteia pe suprafața unei aplicații GIMP neutilizate (în care nu avem deschisă nicio altă imagine), fie prin intermediul comenzii „*File | Open*”. Dacă importăm o imagine prin metoda drag-drop pe o

fereastră GIMP care conține deja cel puțin un layer (strat), noii imagini importate îi va fi alocat automat un nou layer, deasupra tuturor straturilor existente deja. Același efect poate fi realizat și prin comanda „*File / Open as Layers*”.

GIMP este capabil să importe atât imagini bitmap cât și imagini vectoriale. Manipulările grafice pe care le poate realiza pe o imagine importată vor fi însă doar de tip bitmap (pix-map). În funcție de tipul imaginii importate, GIMP alege unul din cele 3 moduri imagine (folosind comanda „*Image / Mode*”), fiecărui mod corespunzându-i un spațiu al culorilor specific:

- RGB: imaginile sunt reprezentate de 3 matrici de pixeli, câte o matrice (canal) pentru fiecare culoare de bază (R – red, G – green, B – blue). Opțional se poate adăuga o a patra matrice (canal), corespunzătoare canalului alpha (de transparență).
- Greyscale: acest mod corespunde imaginilor alb-negru. Fiecare imagine este reprezentată de o singură matrice de pixeli. Opțional putem adăuga un al doilea canal corespunzător informațiilor despre transparență (alpha).
- Indexed: acest mod (din ce în ce mai puțin folosit în prezent) corespunde imaginilor cu un număr mic de culori (în general 256), fiecare culoare fiind indexată în așa-zisa hartă a culorilor („colormap”). Fiecărui pixel al matricii-imagine îi va fi alocată una din culorile disponibile în acest colormap. De fiecare dată când schimbați o imagine RGB într-una Indexed, trebuie să specificați numărul maxim de culori corespunzător noii imaginii.

Exerciții

- B.** Deschideți în GIMP o imagine fie provenind de pe telefonul dumneavoastră mobil (via e-mail), fie provenind de pe un site web sau de la o captură de ecran.
- C.** Verificați modul de lucru. Schimbați modul de lucru la *Greyscale*. Ce se întâmplă cu imaginea? Ce s-a schimbat în dialogul ancorabil „*Channels*”?
- D.** Anulați modificarea precedentă. Schimbați modul de lucru la *Indexed*. Ce se întâmplă cu imaginea?

Crearea unei noi imagini

O imagine nouă poate fi creată cu ajutorul comenzii „*File / New*”. După setarea dimensiunilor imaginii se poate seta și spațiul culorilor cu ajutorul meniului derulant „*Advanced Parameters*”, în care mai poate fi reglată și rezoluția imaginii (Figura 3). Odată imaginea creată, putem acum folosi orice unealtă din Toolbox-ul (Figura 2.b) oferit de GIMP pentru a picta pe aceasta.

Exerciții

- E. Deschideți în GIMP o nouă imagine de dimensiune 640x480 de tip RGB. Căutați în Toolbox unealta „Blend” și formați un gradient de culori de la negru la roșu care să varieze de-a lungul întregii lungimi a imaginii (ca în Figura 4).

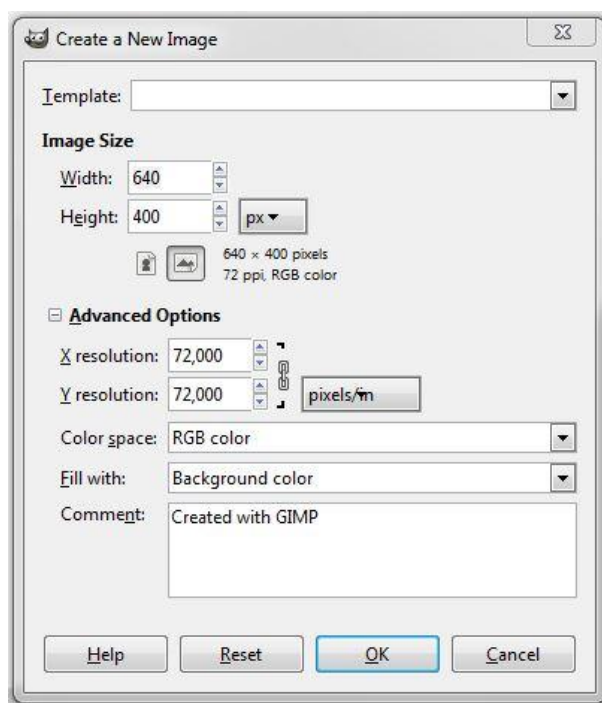


Figura 3 Crearea unei noi imagini în GIMP

Analiza culorilor unei imagini

Cele mai complete informații despre distribuția culorilor unei imagini pot fi aflate din histogramă (comanda „Colors / Info / Histogram”). Histograma arată, pentru fiecare canal al unei imagini (vezi 2.2) câți pixeli au o anumită intensitate. Pentru fiecare canal intensitatea luminoasă variază între 0 și 255 (datorită reprezentării fiecărui pixel din fiecare canal pe câte un octet). Să luăm ca exemplu histograma imaginii din Figura 4 corespunzătoare canalului Red (roșu). Această histogramă este afișată în Figura 5. Se observă din aceasta că nu există mulți pixeli care să aibă o intensitate a canalului roșu inferioară valorii 17 (din 255). În schimb se observă că există o distribuție cvasi-uniformă a numărului de pixeli cu luminozități cuprinse între 18 și 255.

Sub histogramă putem observa date cantitative despre distribuția intensităților luminoase pe fiecare canal, precum media și abaterea standard. Dacă faceți click pe histogramă în orice punct al ei veți putea afla numărul exact de pixeli (ordonată) care au intensitatea aleasă (abscisă).

O altă modalitate de a afla informații despre culorile unei imagini este dacă selectați din meniu comanda „Colors | Info | Colorcube Analysis”. În fereastra care se va deschide poate fi citit numărul total de culori diferite prezente în imagine.

Exerciții

- F. Aflați numărul total de culori diferite prezente în imaginea din Figura 4. Schimbați modul imaginii (spațiul culorilor) din RGB în „Indexed cu un număr maxim de 256 culori”. Ce se întâmplă cu numărul de culori diferite prezente în imagine?

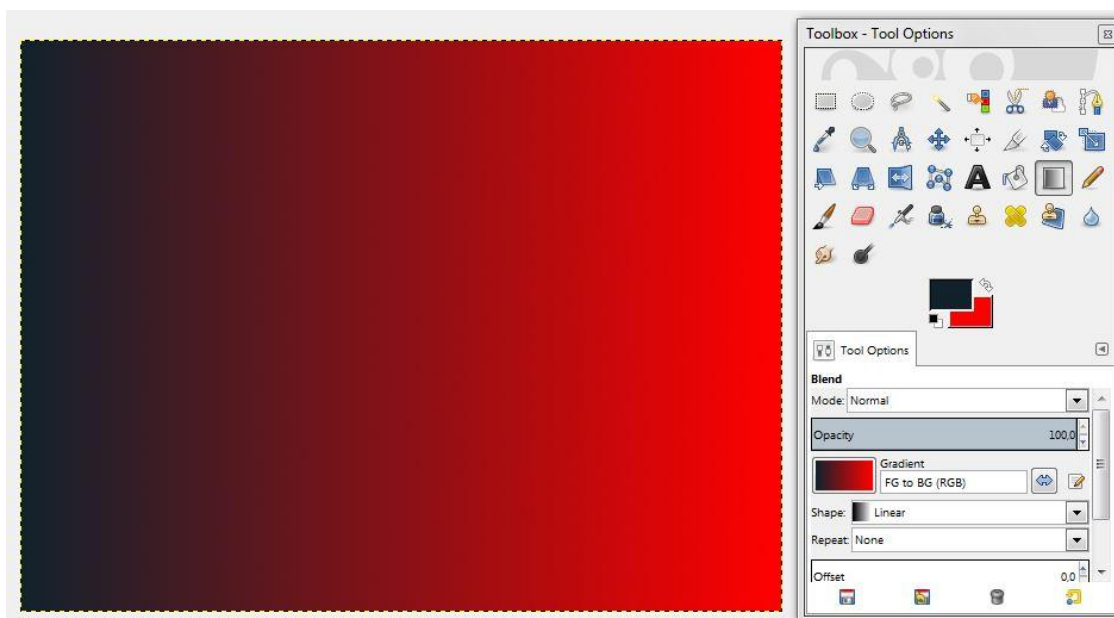


Figura 4 Producerea unui gradient de culori în GIMP folosind unealta „Blend” din Toolbox (selectată în figura de mai sus).

Salvarea și redimensionarea unei imagini

Pentru a salva un proiect GIMP se folosește comanda „File / Save” sau „File / Save as”. Folosind aceste comenzi fișierul rezultat va avea extensia „.xcf” și nu va putea fi citit de alte programe în afară de GIMP. Pentru a salva proiectul sub forma unei imagini bitmap se folosește comanda „File / Export”. GIMP oferă o gamă foarte largă de formate în care imaginea poate fi exportată (.jpg, .eps, .png, .tif, .gif, .bmp, .ico, etc.) convenind așadar oricărui tip de aplicație. Una dintre temele sesiunii viitoare de laborator va fi alegerea formatului de imagine în funcție de aplicația pe care o țintim. Deocamdata ținem doar să remarcăm că pentru a crea o imagine vectorială nu putem folosi comanda „File / Export”.

Un alt considerent important de care trebuie să ținem cont înainte de a salva o imagine este dimensiunea spațiului în care vrem să fixăm imaginea exportată. De exemplu dacă imaginea trebuie să ocupe doar un colț de 320 x 240px pe o pagină web, nu are sens să o înregistrăm la un format superior. Ideal ar fi să construim imaginea în funcție de dimensiunea spațiului final care îi este alocat. Aceasta se face folosind comanda „Image / Scale Image” precum în Figura 6.

Reducerea dimensiunii unei imagini poate fi folosită și cu scopul a reduce dimensiunea documentelor (.doc de exemplu) în care respectiva imagine este integrată.

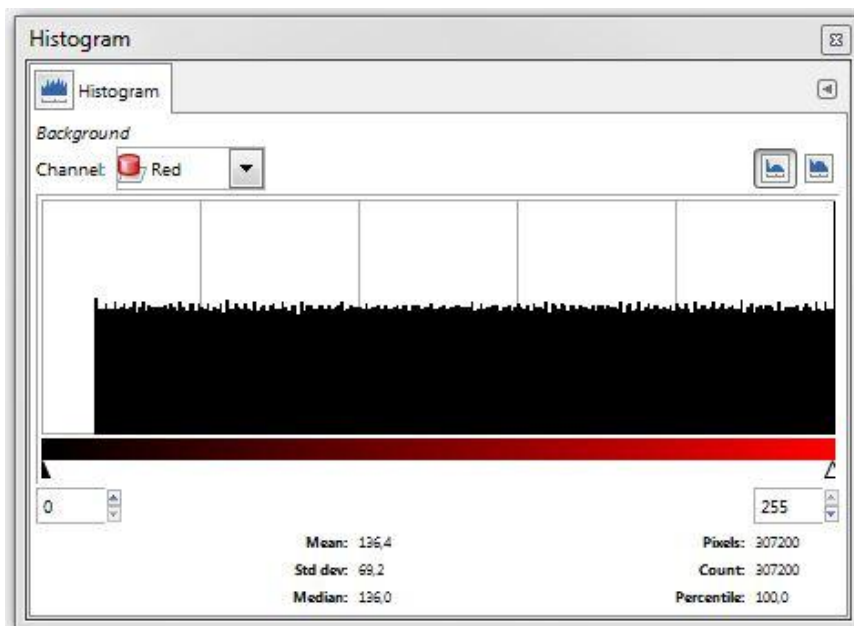


Figura 5 Histograma imaginii din Figura 4 corespunzătoare canalului roșu (Red)

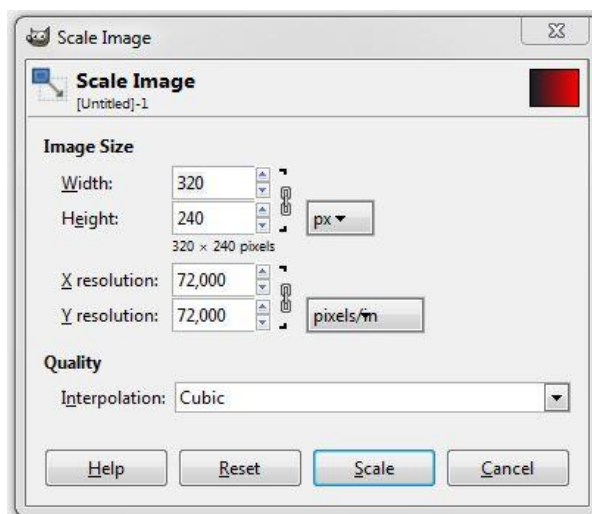


Figura 6 Scalarea imaginii din Figura 4 folosind comanda „Scale Image” din meniul „Image”

Pentru a păstra forma imaginii, raportul între lățimea și înălțimea imaginii trebuie să rămână constant. Aceasta se realizează având grijă ca în dialogul „Scale Image” valoarea pentru lățimea imaginii (Figura 6) să fie legată cu un lanț de valoarea înălțimii imaginii.

Exerciții

- G. Redimensionați imaginea din Figura 4 la o dimensiune de 320 x 240px și exportați noua imagine într-un fișier jpg.

Formate de imagini

A. Formatul .bmp

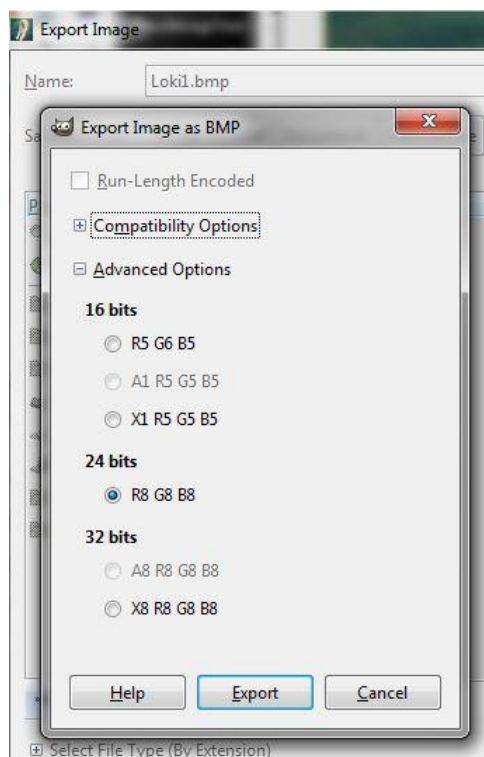
Această secțiune va face o scurtă descriere a celor mai răspândite formate de imagine: .bmp, .jpg și .png. Primul dintre ele, formatul .bmp (prescurtarea de la bitmap) este un format lipsit de pierderi. Din această cauză, fișierele de tip .bmp vor ocupa un spațiu relativ mare pe disc, dar în același timp, pot fi compactate cu un randament foarte bun în arhive de tip .zip sau .rar, datorită cantității mare de informație redundantă pe care o conțin. O altă caracteristică a acestui format este lipsa patentării, ceea ce îl face răspândit și acceptat de majoritatea sistemelor de operare. Ca și structură a fișierului, fișierele .bmp au mai multe headere de dimensiune prestabilită, precum și porțiuni de informație cu dimensiune variabilă, ce se modifică de la caz la caz. Primul header al fișierului are 14 octeți. Primii doi octeți din acest header sunt destinați identificării tipului de fișier. Majoritatea fișierelor de tip .bmp au o valoare fixată a acestor doi octeți (în hexazecimal) de [0x42] [0x4D]. Următorii 4 octeți desemnează dimensiunea fișierului .bmp (în octeți), iar următorii 4 octeți sunt rezervați pentru aplicația în care este creată imaginea. Valoarea ultimilor 4 octeți ai header-ului reprezintă offsetul (adresa) primului octet din matricea propriu-zisă de pixeli (cea care caracterizează fiecare pixel în parte din imagine).

Al doilea header al fișierului .bmp, așa-zisul header DIB (Device Independent Bitmap), este cel care fixează felul în care sunt ordonate informațiile în matricea de pixeli: lungimea și înălțimea matricii (exprimate în pixeli), numărul de biți alocați fiecărui pixel, rezoluția imaginii pe orizontală și pe verticală (numărul de pixeli pe unitatea de lungime) și numărul de culori din paleta de culori. Doar pentru imaginile indexate, header-ul DIB are posibilitatea de a prevedea folosirea unui algoritm de compresie fără pierderi (în general de tip RLE).

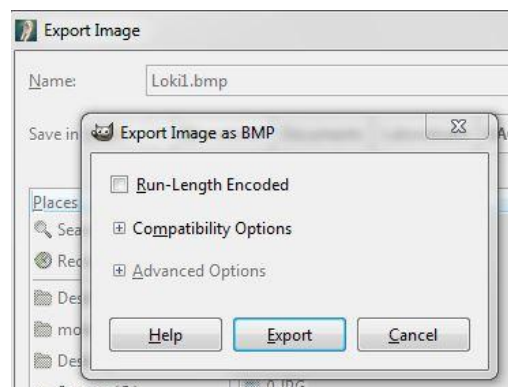
Imediat sub header-ul DIB urmează tabelul ce indexează culorile din paletă. Pentru imaginile indexate (Secțiunea 2.3) fiecare pixel din matricea imaginii este descris printr-un număr de biți (1, 4 sau 8) ce reprezintă indexul unei culori din acest tabel. În imaginile care nu sunt indexate, acest tabel are doar rolul de listare a tuturor culorilor ce se găsesc în imagine cu rol în optimizarea afișajului pe ecranele cu număr limitat de culori. Fiecare culoare din acest tabel ocupă 4 octeți de memorie, folosindu-se formatul RGBA32.

Imediat după header-ul DIB urmează în general matricea propriu-zisă de pixeli, în care pixelii din imagine sunt organizați conform informațiilor oferite de header-ul DIB. Spațiul de memorie ocupat de fiecare rând de pixeli din imagine este un multiplu de 4 octeți, echivalent cu 32 de biți. Dacă în urma multiplicării numărului de biți alocați fiecărui pixel cu dimensiunea pe orizontală a matricii de pixeli, dimensiunea astfel obținută pentru un rând nu este un multiplu de

32, atunci rândul va fi completat cu biți inutilizați până la umplerea și ultimului grup de 32 de biți. Dimensiunea ocupată în memorie de întreaga matrice de pixeli este egală cu dimensiunea (în octeți) a unui rând de pixeli, înmulțită cu numărul de rânduri ale imaginii (așa-zisa înălțime a matricii imagine).



a.



b.

Figura 7 Opțiunile oferite de GIMP la salvarea unei imagini în formatul .bmp: a. pentru imaginile ne-indexate, putem fixa numărul de biți per pixel; b. pentru imaginile indexate, putem opta pentru compresia imaginii folosind algoritmul RLE.

La salvarea cu GIMP a unei imagini neindexate, o fereastră de dialog ne permite selecționarea numărului de biți per pixel, precum în Figura 7a. Opțiunea selectată automat pentru cazul unei imagini lipsită de canalul alpha (canalul de transparență) este de 24 de biți ($R_8G_8B_8$ - 8 biți pentru canalul roșu, 8 biți pentru cel verde și 8 biți pentru cel albastru). În cazul în care imaginea ar fi avut și un canal alpha, atunci opțiunea selectată automat ar fi fost de 32 de biți ($A_8R_8G_8B_8$). Dacă selectăm în schimb opțiunea $X_8R_8G_8B_8$, am opta pentru neglijarea canalului alpha, iar primii 8 biți ar fi nefolosiți. Pentru imaginile indexate, nu putem fixa numărul de biți per pixel, dar putem opta în schimb pentru folosirea algoritmului de compresie fără pierderi RLE (Run-Length Encoding), precum în Figura 7b.

B. Formatul .jpg

Formatul .jpg sau .jpeg (Joint Photographic Expert Group) este un format de compresie cu pierderi. Datorită algoritmului de compresie folosit, poate atinge în mod tipic rate de compresie de 10:1 față de imaginea originală, fără pierderi evidente în calitatea imaginii. Cele mai răspândite două tipuri de fișiere care suportă acest format de compresie sunt „JPEG/Exif” (folosit în special de camerele digitale de fotografiat) și „JPEG/JFIF” (rezultat în urma manipulării imaginii într-un editor de imagini). Utilizarea acestui standard de compresie se pretează cel mai bine cazurilor în care avem o variație lină a tonului și a culorii unei fotografii sau picturi, și mai puțin acolo unde trecerile sunt bruște, precum atunci când fotografiem text sau desene în creion. Artefactele pe care standardul .jpg le introduce în aceste din urmă cazuri pot duce la scăderea vizibilă a calității unei fotografii.

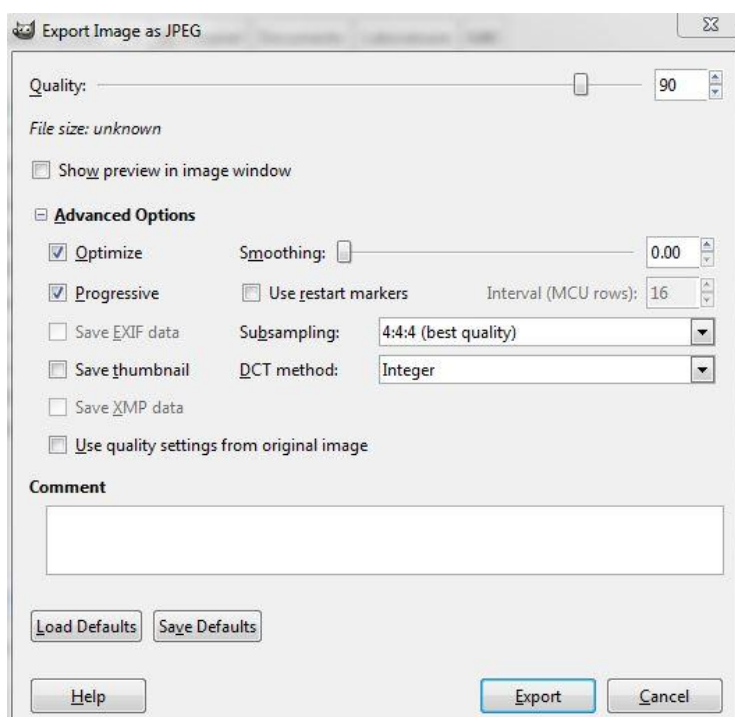
Algoritmul de compresie propriu-zis care stă la baza reducerii dimensiunii imaginii, pornește de la transformarea spațiului culorilor din RGB în $Y C_B C_R$. În noul spațiu al culorilor, canalul Y reprezintă luminozitatea fiecărui pixel, iar canalele C_B și C_R sunt cele două canale de culoare rămase, cărora li se reduce rezoluția cu un factor de 2 sau 3. Această reducere a rezoluției culorilor, care nu afectează canalul de luminozitate, are la bază caracteristica vederii umane, care este mai sensibilă la detaliile luminozității decât la detaliile culorilor. Imaginea este împărțită apoi în blocuri de 8x8 pixeli, cărora li se aplică transformata DCT (Discrete Cosine Transform) ce transpune imaginea în domeniul frecvenței spațiale. După obținerea matricii transformate (de dimensiune 8x8) fiecare element al acestei matrici va fi divizat la elementul corespondent situat într-o matrice de cuantizare. Urmează rotunjirea fiecărui element al matricii astfel rezultate la cel mai apropiat întreg. Această etapă este singura care introduce pierderi. După această etapă elementele din matrice corespunzătoare frecvențelor spațiale înalte devin în general zero. Și celelalte elemente din matrice (corespunzătoare frecvențelor joase) vor avea mult mai puține cifre semnificative și vor putea fi așadar reprezentate printr-un număr mai mic de biți decât înainte de cuantizare.

Meniul GIMP care precede salvarea unei imagini în format .jpg este afișat în Figura 8a. Factorul de calitate este cel mai important parametru, fiind cel în funcție de care sunt reglați coeficienții matricii de cuantizare. La alegerea unui factor de calitate scăzut, coeficienții matricii de cuantizare vor avea valori mari, ceea ce va duce (după folosirea lor la divizarea matricii transformate) la anularea multor frecvențe înalte.

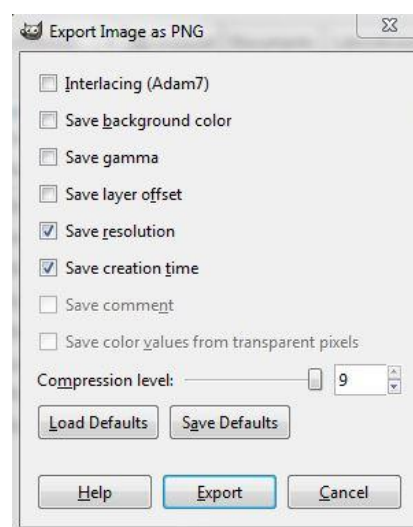
C. Formatul .png

Formatul .png (Portable Network Graphics) este un format de compresie fără pierderi, apărut ca un înlocuitor mai performant al formatului patentat .gif (la rândul lui bazat pe algoritmul de compresie LZW). La ora actuală .png este cel mai folosit tip de format fără pierderi de pe internet. Ca structură interioară, un fișier .png debutează cu o semnătură specifică pe 8 octeți. După această header-semnătură, urmează mai multe bucăți cu rol în definirea proprietăților

imaginii. Unele din aceste bucăți sunt critice (precum cele care definesc lungimea și înălțimea imaginii, numărul de biți alocați fiecărui pixel, informațiile despre paleta de culori, etc.) în timp ce altele sunt doar auxiliare (precum profilul ICC necesar la calibrarea culorilor) și nu sunt descifrate de toate aplicațiile ce folosesc imaginile .png. O parte din aceste câmpuri auxiliare (precum culoarea preferată a fundalului, data ultimei modificări, valoarea parametrului gamma, etc) pot fi setate în fereastra de dialog ce precede în GIMP salvarea unei imagini în formatul .png (Figura 8b).



a.



b.

Figura 8 Opțiunile oferite de GIMP la salvarea unei imagini în: a. formatul .jpg; b. formatul .png.

Prima opțiune din fereastra GIMP destinată salvării unei imagini în format .png, odată activată, permite salvarea intercalată a imaginii. În acest mod, ordonarea octeților de informație nu se face în aceeași secvență în care pixelii sunt aranjați în imagine, ci se face intercalat, în așa fel încât imaginea să poată fi reconstituită chiar și dacă transferul imaginii nu este complet. Selectarea acestei modalități de reprezentare a imaginii este indicată atunci când folosim imaginea pe o pagină web și nu putem fi siguri de viteza conexiunii clienților la server, însă are și dezavantajul unei compresii de calitate mai redusă față de cazul reprezentării normale a imaginii.

Standardul .png oferă multe opțiuni de utilizare a canalului alpha (de transparență). Acest canal poate fi adăugat atât la imaginile ne-indexate în culori, cât și la cele reprezentate doar în

nuanțe de gri. În plus, imaginile indexate pot conține palete de culori în care să fie incluse și informațiile legate de transparența culorilor indexate de paletă.

Exerciții

- H. Care este dimensiunea maximă pe care o poate avea un fișier .bmp, neglijând alte posibile limitări date de sistemul de operare sau capacitatea hardware a calculatorului, știind că porțiunea din header destinată notării acestei dimensiuni are 4 octeți?
- I. Ce dimensiune, în octeți, are o matrice de pixeli atunci când o reprezentăm folosind standardul .bmp, dacă informațiile date de header-ul DIB sunt următoarele: numărul de pixeli ai matricii pe orizontală (lungimea matricii) este egal cu 545, numărul de pixeli ai matricii pe verticală (înălțimea matricii) este egal cu 345 și numărul de biți per pixel este egal cu 16?
- J. Salvați aceeași imagine neindexată în format .bmp, folosind următoarele adâncimi de culoare: 16 biți per pixel, 24 de biți per pixel și 32 de biți per pixel. Cum variază dimensiunea imaginii salvate?

Calibrarea rezoluției monitorului

Pentru a ajuta calculatorul să facă o corespondență corectă între o lungime exprimată în pixeli și una exprimată în centimetri, este nevoie de o calibrare manuală a monitorului. Aceasta deoarece fiecare ecran are propria sa densitate specifică de pixeli pe cm^2 . Această calibrare manuală a monitorului se face din meniul Display accesibil după rularea comenzii „*Edit / Preferences*”. În dreptul opțiunii „*Monitor Resolution*” se selectează opțiunea „*Enter manually*” urmată de comanda „*Calibrate*”. În ecranul care se deschide, trebuie să utilizați o riglă sau o ruletă pentru măsurarea dimensiunilor axelor verticală și orizontală afișate și să treceți rezultatele în locurile corespunzătoare.

Exerciții

- K. Calibrați rezoluția monitorului.

Calibrarea culorilor

Multe dispozitive utilizate pentru design sau fotografie, cum ar fi camere digitale foto, scanere, display-uri, imprimante, etc, au propriile lor caracteristici de reproducere a culorilor. În cazul în care acestea nu sunt luate în seamă în timpul deschiderii și editării unei imagini, riscați să faceți ajustări negative în calitatea acestora.

În general, camerele fotografice sau scanerele care au profile ale culorilor diferite de standardul sRGB utilizat de majoritatea editoarelor fotografice vor integra în poză informația

necesară pentru interpretarea corectă a culorilor oferite de acestea. GIMP va afișa automat o fereastră de dialog de fiecare dată când întâlnește o astfel de imagine, cu scopul convertirii ei la standardul culorilor sRGB (standard ce corespunde unei temperaturi a culorilor de 6500K).

Mai dificil este când monitorul este cel care nu este calibrat și afișează culorile într-un mod diferit față de celelalte dispozitive de afișaj. În această situație, o imagine care apare perfect pe calculatorul utilizatorului după publicarea ei pe internet, va apărea diferit pe calculatoarele celorlalți vizitatori ai paginii web pe care imaginea este publicată. GIMP rezolvă această problemă prin intermediul așa-ziselor filtre de afișaj, accesibile prin comanda „*View / Display Filters*”. În dialogul care apare, filtrul „Color Management” trebuie selectat și trecut în panoul din partea dreaptă, precum în Figura a. Deschideți apoi comanda „*Edit / Preferences*” și selectați dialogul „Color Management” (10b). Selectați la modul de operare opțiunea „Color managed display” și introduceți un fișier de profil al monitorului în dreptul opțiunii „Monitor profile”. Dacă nu aveți un fișier de profil al monitorului, utilizați profilul creat de sistemul de operare prin selectarea opțiunii „Try to use the system monitor profile”.

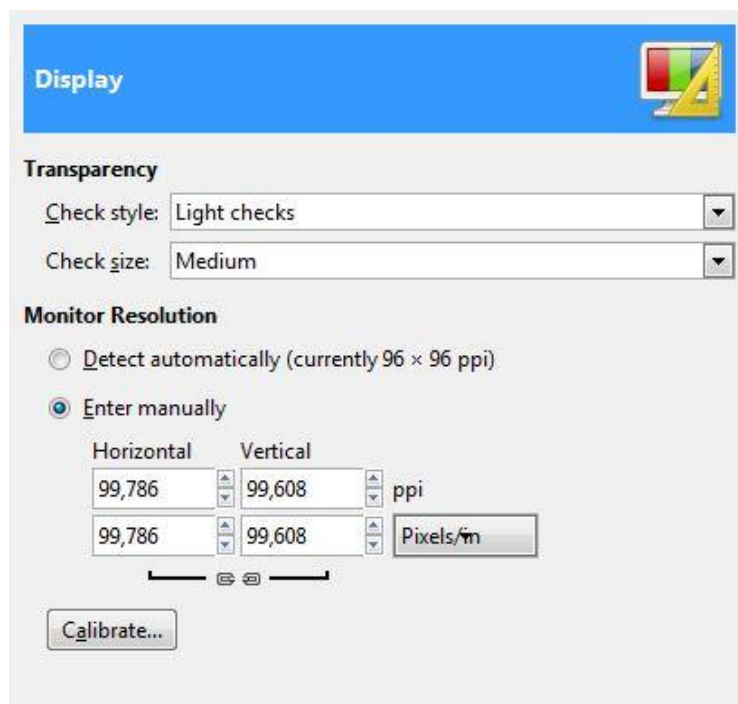


Figura 9. Calibrarea rezoluției

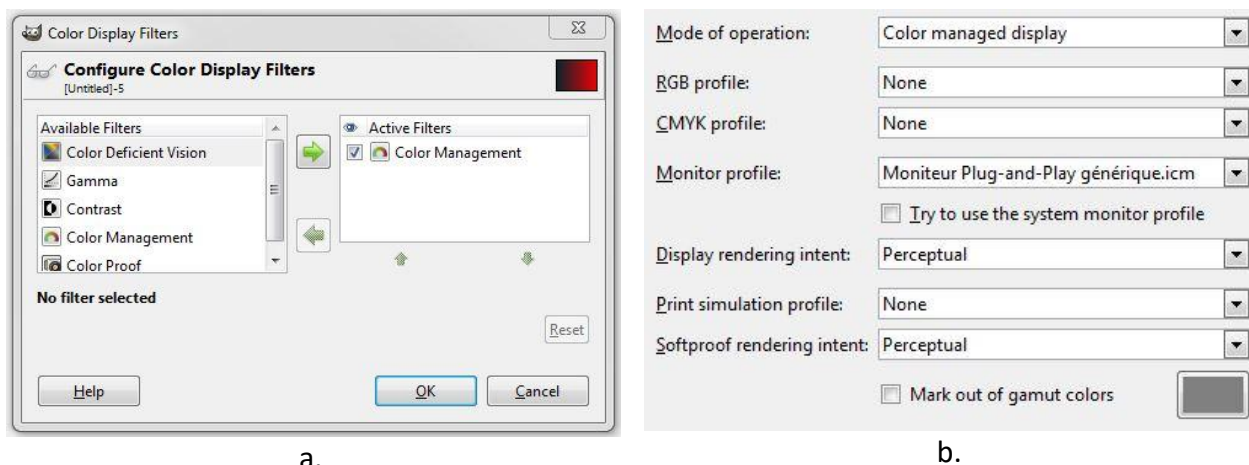


Figura 10. Configurarea profilului culorilor pentru monitor în GIMP: a. Comanda "Display Filters" din meniul View, b. Dialogul "Edit | Preferences | Color Management".

Desfășurarea lucrării

1. Familiarizați-vă cu interfața GIMP-ului și aranjați-o respectând cerințele exercițiului **A** din secțiunea 2.2.

2. Rezolvați exercițiile **B**, **C** și **D** (secțiunea 2.3). Salvați cele 3 imagini rezultate pe rând în formatele .jpg, .tif și .png. Ordonati cele 9 imagini salvate pe disc în funcție de dimensiunea lor.

3. Familiarizați-vă cu operațiile de generare, analiză a culorilor, redimensionare și salvare ale unei imagini prin rezolvarea exercițiilor **E**, **F** și **G** (secțiunile 2.4, 2.5 și 2.6).

4. Familiarizați-vă cu structura unei imagini bitmap rezolvând exercițiile **H**, **I**, **J** (din secțiunea 2.7).

5. Deschideți cu GIMP o imagine care conține cel puțin 10^5 culori. Salvați-o în formatele .jpg, .tif, .png, .eps, .gif și .bmp întâi fără nicio modificare, apoi reducându-i pe rând numărul de culori la 256, respectiv 10 culori și notați de fiecare dată dimensiunea imaginii (memoria ocupată pe disc). Treceți rezultatele într-un tabel și comentați efectul reducerii numărului de culori asupra dimensiunii imaginii, în funcție de format. Care format ocupă inițial cel mai puțin spațiu pe disc? Pentru ce format efectul reducerii numărului de culori asupra dimensiunii este cel mai pronunțat? Notă: de fiecare dată la salvare se aleg parametrii standard (default) pentru fiecare format.

6. Deschideți cu GIMP o imagine de cel puțin 500x500pixeli. Salvați-o în format .jpg variind factorul de calitate între 100 și 1 (prin divizare succesivă la 2). Notați de fiecare dată dimensiunea pe disc a imaginii și desenați un grafic al dependenței acesteia de factorul de calitate.

7. Deschideți cu GIMP o imagine de cel puțin 500x500pixeli. Salvați-o în format .png variind numărul de culori între 256 și 2 (prin divizare succesivă la 2). Notați de fiecare dată dimensiunea pe disc a imaginii și desenați un grafic al dependenței acesteia de numărul de culori. Desenați și graficul dependenței dimensiunii pe disc a imaginii în funcție de logaritmul în baza 2 din numărul de culori.

8. Deschideți cu GIMP o imagine de cel puțin 500x500pixeli. Salvați-o în format .jpg cu un factor de calitate de 60. Numiți-o „poza1.jpg”. Inchideți poza și apoi deschideți imaginea nou salvată. Trasați o linie la întâmplare pe imagine folosind unealta creion (Pencil Tool) din Toolbar. Pentru a trasa dreapta, se va alege o extremitate a dreptei cu un click. Apăsând Shift, un al doilea click de mouse va fixa locul unde se va termina dreapta. Se resalvează imaginea în format .jpg („poza2.jpg”) cu un factor de calitate de 60. Se închide imaginea și se deschide imaginea nou salvată. Se trasează o nouă dreaptă, diferită de prima și iarăși se salvează imaginea în același format. Operațiunea se repetă de 15 ori, până ajungem la „poza16.jpg”.

Ultima imagine („poza16.jpg”) salvată va conține 15 drepte trasate la întâmplare. Se compară zonele nesuprapuse de linii, dar aflate în imediata vecinătate a acestora cu zonele identice din prima imagine în format .jpg („poza1.jpg”). Se observă diferențe de calitate? Cum explicați rezultatul?

9. Deschideți cu GIMP o imagine de cel puțin 500x500pixeli, reprezentând o fotografie făcută după o pagină de carte plină de text. Salvați-o în format .jpg cu un factor de calitate de 60. Numiți-o „poza1.jpg”. Inchideți poza și apoi deschideți imaginea nou salvată. Decupați imaginea în așa fel încât noua dimensiune să aibă cu 3 pixeli mai puțin atât pe verticală, cât și pe orizontală. Repetați operațiunea de salvare în format .jpg („poza2.jpg”) folosind același factor de calitate. Repetați de cel puțin 10 ori închiderea și re-salvarea imaginii decupând mereu câte 3 pixeli din dimensiunile imaginii. Se compară calitatea imaginii după ultima decupare cu cea obținută înainte de prima decupare („poza1.jpg”).

10. Deschideți o imagine având cel puțin 1000x500 de pixeli. Salvați-o pe rând în formatele .jpg, .png, .gif și .bmp. Reduceți-i dimensiunea la jumătate atât pe verticală cât și pe orizontală. Suprafața imaginii se reduce așadar de 4 ori. Ce se întâmplă cu spațiul ocupat de imagine pe disc în cazul tuturor celor 4 formate?

Capitolul 2. Prelucrarea avansată de imagini cu GIMP

Obiectivul lucrării

Această lucrare își propune obișnuirea studenților cu folosirea uneltelor de selecție și desen ale GIMP-ului. La sfârșitul acestui laborator studenții vor fi capabili să adauge propriile artefacte grafice sau text pe o imagine, să modifice structura unei imagini existente, precum și să realizeze imagini vectoriale simple.

Breviar teoretic

Uneltele din Toolbox

Am văzut în primul laborator cum se poate crea un gradient de culori într-o nouă imagine. Unealta care ne-a permis să realizăm acel desen este doar una din cele câteva zeci de unelte disponibile în GIMP. În Figura 1 le putem observa pe cele mai importante dintre acestea, grupate pe categorii: unelte de selecție, pentru transformări geometrice, de desen, etc. Cunoașterea opțiunilor fiecăreia dintre aceste unelte poate face diferența între un începător și un profesionist al prelucrării de imagini.

Uneltele de selecție

Uneltele de selecție pot fi accesate direct din Toolbox unde ocupă primele 7 poziții, sau accesând submeniul „Tools | Selection Tools”. Odată selectată o zonă din imagine, aceasta va apărea înconjurată de un contur fin de-a lungul căruia segmente negre se deplasează deasupra unui marcaj alb. Să luăm exemplul primului instrument prezent în Toolbox, cel al selecției dreptunghiulare. Opțiunile asociate acestui instrument le putem vedea în Figura 8, cele mai importante fiind marcate cu roșu. Cele patru moduri de selecție („*Mode*”) posibile fixează felul în care noua selecție pe care o vom face se va raporta la zonele deja selectate din imagine:

- Modul înlocuire (*replace*): alegerea primului mod face ca noua selecție să înlocuiască total selecțiile anterioare
- Modul adunare (*add*): face ca noua selecție să se adauge la zonele deja selectate din imagine; se poate activa temporar dacă Țineți tasta *Shift* apăsată în momentul în care faceți o selecție

- Modul scădere (*subtract*): face ca zona corespunzătoare noii selecții să fie scăzută din zonele din imagine deja selectate, conducând la deselectarea ei; se poate activa temporar prin apăsarea tastei *Ctrl*
- Modul intersecție (*intersect*): face ca zona selectată să fie definită de suprafața de intersecție dintre regiunile din imagine deja selectate și noua regiune aleasă.

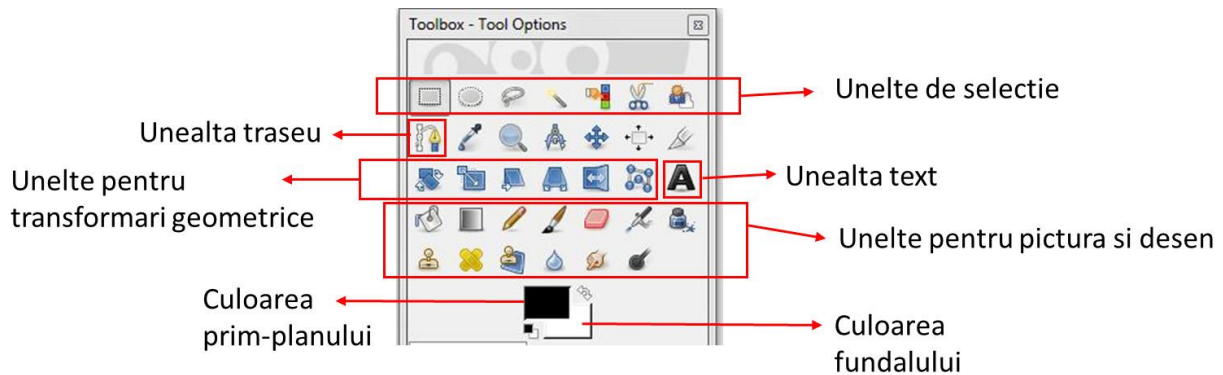


Figura 7 Uneltele din Toolboxul GIMP: de selecție, pentru transformări geometrice, penru desen, pentru crearea și editarea traseelor și pentru text

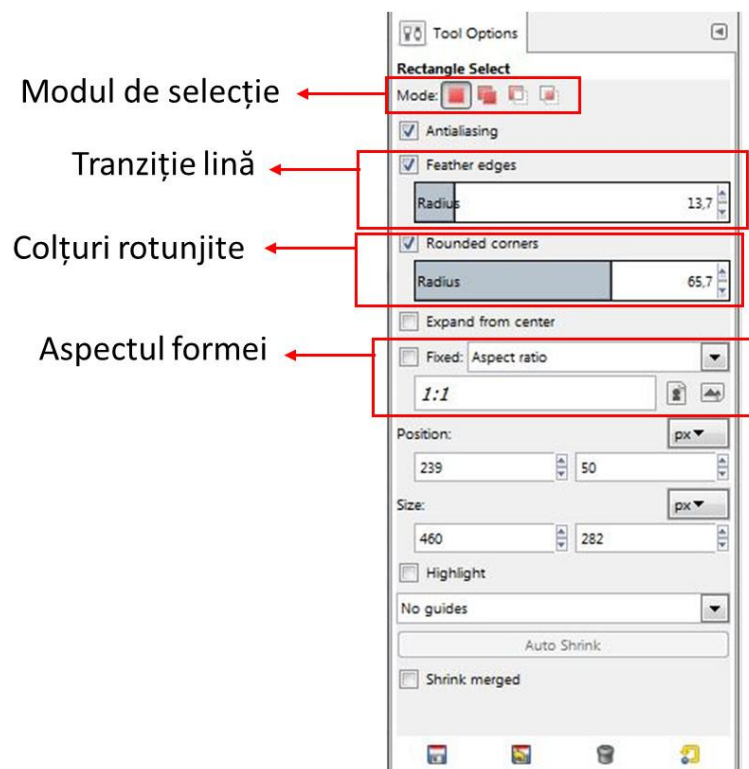


Figura 8 Proprietățile Unelei de Selecție dreptunghiulară

Alt parametru al uneltelor de selecție, „*Feather edges*” (Figura 8), ne permite să alegem dacă tranziția dintre zona selectată și restul imaginii să se facă brusc (caz în care această opțiune nu este bifată) sau lin (caz în care căsuța corespunzătoare trebuie bifată). Selecția așadar nu este

văzută de GIMP ca o matrice de valori boolene, în care zonele marcate cu „true” sunt alese, iar celelalte nu. Selecția poate fi văzută așadar ca o matrice de octeți, în care fiecărui pixel din imagine îi corespunde un octet a cărui valoare este proporțională cu intensitatea selecției (255 – acel pixel este selectat total; 0 – acel pixel este ignorat total). Pentru cazul în care ne dorim o tranziție lină a selecției, trebuie în plus să alegem o valoare și pentru lungimea („radius”) zonei de tranziție. O modalitate și mai eficientă de a face o selecție cu intensitate graduală este prin comanda din meniu „*Select / Feather...*”. Comanda „*Select / Sharpen*” are rolul invers, de a transforma tranziția dintr-una graduală într-o tranziție bruscă. Atenție! Toate modificările legate de tranziție nu se vor răsfrânge asupra unei selecții deja realizate, ci asupra tuturor selecțiilor viitoare. Din păcate reprezentarea selecțiilor printr-o frontieră nu ne spune nimic despre intensitatea cu care este selectat fiecare pixel în parte în interiorul regiunii selectate. Conturul este de fapt doar o linie de demarcație între regiunea în care fiecare pixel este preponderent selectat (având o intensitate de selecție superioară valorii de 128) și regiunea în care fiecare pixel este preponderent ignorat (având o intensitate de selecție inferioară lui 128). Pentru a vedea intensitatea de selecție pentru fiecare pixel în parte, putem activa modul *Quick Mask* (din meniul „*Select*”, comanda „*Toggle Quick Mask*”). Putem face activarea acestui mod de afișaj și prin apăsarea tastelor *Shift+Q*. După activare, regiunile complet ignorate din imagine vor fi marcate cu roșu iar regiunea complet selectată va fi lăsată neschimbată. Pixelii care au o intensitate de selecție intermediară vor fi marcați cu atât mai accentuat cu cât intensitatea lor de selecție este mai ridicată. Fiecare comandă GIMP va avea o semnificație diferită după activarea modului de afișaj „*Quick Mask*”, de aceea pentru revenirea la funcționalitatea normală a GIMP-ului este nevoie de dezactivarea acestui mod (prin apăsarea tastelor *Shift+Q*).

Următorul parametru reglabil al unei selecții dreptunghiulare este curbura colțurilor. Il putem activa prin bifarea căsuței „*Rounded corners*” (Figura 8) și alegerea unei valori pentru raza de curbură. În final, alegerea unui model predefinit pentru forma selecției se face prin bifarea căsuței „*Fixed*” și optarea pentru una din opțiunile prezentate:

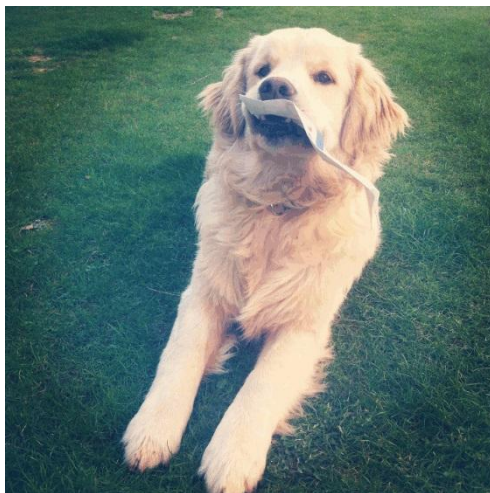
- *Aspect ratio*: regiunea selectată va avea un raport lungime/înălțime fix, definit de utilizator prin căsuța text de dedesubt
- *Width*: selecția va avea o lungime fixă, definită de utilizator
- *Height*: selecția va avea o înălțime fixă, definită de utilizator
- *Size*: selecția va avea atât o lungime, cât și o înălțime prestabilite.

Următoarele două unelte de selecție sunt Selecția Eliptică (*Ellipse Select*) și Lasoul (*Free Select*) ce pot selecta o regiune cu un contur eliptic, respectiv o regiune delimitată de un contur desenat cu mâna liberă. Toate regiunile selectate cu o anumită unealtă pot fi combinate cu regiunile selectate cu alte unelte, dacă modul de selecție permite acest lucru.

Următoarele două unelte din Toolbox pot realiza selecția unei regiuni pe baza culorilor. Prima dintre acestea, denumită „*Fuzzy Select*”, selectează o regiune contiguă de pixeli în care culoarea acestora nu diferă cu mai mult de o valoare prestabilită de valoarea culorii pixelului pe care utilizatorul a apăsător. Valoarea abaterii prestabilite este stabilită în căsuța „*Threshold*” (prag).

Dacă de exemplu pixelul pe care utilizatorul a apăsător are codul RGB al culorii (R_0, G_0, B_0), toți pixelii din selecție vor avea un cod al culorii de forma ($R_0 \pm \text{prag}$, $G_0 \pm \text{prag}$, $B_0 \pm \text{prag}$). Cealaltă unealtă, denumită „*Select by Color*”, face același lucru, însă nu selectează doar o regiune contiguă în care se află pixelul pe care utilizatorul a apăsător, ci toți pixelii din imagine (indiferent de locația lor) care respectă condiția de culoare precedentă.

Ultima unealtă de selecție pe care o vom acoperi în acest îndrumar este „*Scissors Select Tool*”, care este indicată în cazul în care regiunea de selecție urmează conturul unui obiect din imagine. Folosind această unealtă, utilizatorul va trebui să aleagă doar câteva puncte de-a lungul conturului obiectului grafic pe care dorește să-l selecteze, iar programul va încerca să determine conturul obiectului printr-un algoritm de detecție inteligentă.



a.



b.

Figura 9 Selectarea unui obiect grafic în GIMP: a. Imaginea originală, b. Imaginea nouă.

După ce ați selectat o regiune din fotografie, majoritatea uneltelor și filtrelor din GIMP vor acționa pe viitor doar asupra selecției dumneavoastră, neglijând regiunile din afara selecției. Pentru a schimba regiunea selectată într-una neselectată și pe cea neselectată într-o selecție, este suficientă comanda „*Select / Invert*”.

Exerciții

- A. Deschideți fișierul „Laborator GAC/loki.jpg” (Figura 9a.) și selectați câinele din imagine. Copiați selecția (Ctrl+C) și lipiți-o într-o imagine nouă având dimensiunile celei dintâi (Figura 9b.). Salvați noua imagine în fișierul „Laborator GAC/loki_prenumele_dvs.jpg”.

Specificarea culorilor

Înainte de a trece la prezentarea uneltelor pentru pictură ale GIMP-ului, trebuie știut că acestea vor folosi fie culoarea prim-planului, fie pe cea a fundalului pentru a da culoare regiunii selectate. Cele două culori folosite așadar de instrumentele pe care le vom discuta în continuare se pot fixa prin apăsarea mouse-ului pe fiecare din ele, în partea de jos a Toolbox-ului (Figura 1). Fereastra de dialog care se deschide este cea din Figura 10. Se poate observa că reglarea culorii poate fi făcută fie prin folosirea formalismului HSV (Hue – Saturation- Value), fie prin folosirea formalismului RGB (Red – Green – Blue). Codul RGB hexazecimal al culorii selectate în prezent (afișată separat în căsuța „*Current*”) este scris în căsuța text „*HTML notation*”. Acesta este codul care poate fi notat în dreptul proprietăților de culoare ale tagurile HTML dedicate, pentru ca o un element web să împrumute culoarea respectivă.

Poate cea mai utilă metodă de a fixa o culoare este prin folosirea unelei de extragere a culorii, marcată separat în Figura 10 dar prezentă și în Toolbox (la dreapta unelei Traseu din Figura 1). Pentru a o folosi se apasă scurt deasupra iconiței ei din dialogul de schimbare a culorii. După eliberarea mouse-ului, se apasă din nou pe acesta, de data aceasta continuu. În timpul mișcării mouse-ului (ținând butonul apăsat) vom putea observa în căsuța culorii curente („*Current*”) culoarea pe care o are pixel-ul deasupra căruia ne aflăm în momentul respectiv. Putem să ne deplasăm cu mouse-ul pentru a citi culori care se află chiar și în afara aplicației GIMP. Când ne aflăm deasupra culorii dorite, putem elibera mouse-ul, iar culoarea curentă va rămâne fixată cu ultima valoare a ei.

Exerciții

- B.** Folosind unealta de extragere a culorii, realizați o imagine de dimensiunea 1024x900 pixeli, care să aibă culorile unei pagini Facebook de log-in. Imaginea finală trebuie să conțină, aidoma aspectului de fundal al unei pagini facebook, de sus în jos: o regiune albastru-închis de maxim 100 pixeli înălțime, un gradient de la un albastru ultra-deschis la un albastru deschis ce ocupă aproape toată imaginea și (în partea de jos) o regiune albă de maxim 50 pixeli înălțime.

Unelte pentru pictură și desen

Unelte de selecție prezentate mai sus pot fi folosite și ca unelte pentru desen. Ele sunt utile în special în cazul în care vrem să desenăm forme regulate, precum dreptunghiuri, pătrate, elipse, cercuri. Pentru aceasta, este suficient să apăsăm comanda „Edit | Stroke Selection...” imediat după ce terminăm de făcut o selecție având forma dorită. În dialogul care apare se va alege stilul liniei ce va contura forma selectată, precum în Figura 11. Linia poate fi trasă fie direct (opțiunea „*Stroke line*”), printr-o umplere uniformă a conturului folosind culoarea de prim-plan (*Foreground color*), fie folosind una dintre unelte dedicate de pictură și desen (opțiunea „*Stroke with a paint tool*”), pe care le vom explica mai jos. Dar mai înainte, trebuie să remarcăm că pe lângă desenarea conturului unei forme rezultate dintr-o selecție, putem face și pictarea interiorului său prin una din comenzile „*Edit | Fill with BG Color*”, „*Edit | Fill with FG Color*”

sau „*Edit / Fill with Pattern*”, care realizează umplerea formei selectate fie folosind culoarea fundalului, fie pe cea a prim-planului, fie cu textura activă.

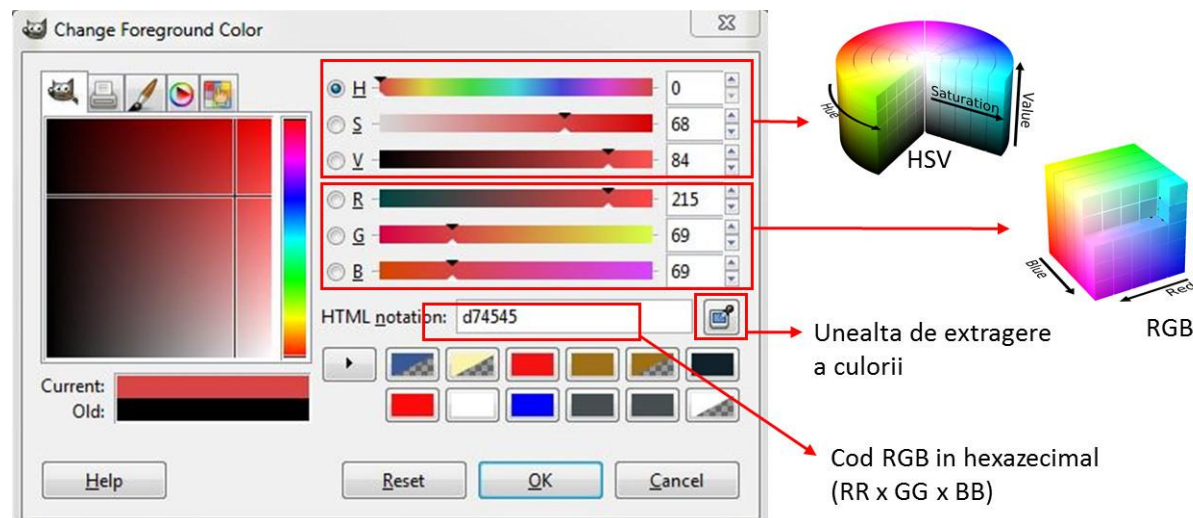


Figura 10 Schimbarea culorii prim-planului

Problema pe care o întâlnim atunci când încercăm să desenăm folosind exclusiv uneltele de selecție constă în imposibilitatea de a desena forme deschise. Uneltele de selecție delimitează doar forme închise. Există însă o unealtă mult mai puternică decât cea de selecție care ne permite să desenăm și forme deschise: Unealta Traseu (*Paths Tool*). Această unealtă, fiind utilă și în generarea de forme și imagini vectoriale, va fi discutată separat, în secțiunea 1.5.

Această secțiune va fi dedicată în continuare trecerii în revistă a principalelor unelte GIMP folosite pentru pictură și desen, și care pot fi folosite atât independent cât și pentru conturarea unui traseu sau unei selecții (prin activarea opțiunii „*Stroke with a paint tool*” din Figura 11).

1. Bucket Fill Tool (unealta de umplere): Această unealtă este folosită pentru umplerea uniformă a suprafeței selectate sau a fotografiei folosind culoarea fundalului, a prim-planului sau folosind textura activă. Alegerea între cele trei tipuri de umpleri se face în dialogul ancorabil de opțiuni ale uneltei („*Tool options*”), proprietatea „*Fill Type*”. Următorul parametru din acest dialog, „*Affected Area*”, fixează dacă umplerea cu noua culoare se va face pe toată suprafața selecționată, sau doar pe o regiune contiguă din preajma pixelului apăsat. În cazul în care se alege această a doua variantă („*Fill similar colors*”) doar pixelii învecinați având culori apropiate de culoarea pixelului apăsat vor fi colorați. Pragul de la care culoarea unui pixel va fi considerată prea depărtată este setat de parametrul „*Threshold*”, într-un mod analog cu ce am discutat mai sus pentru unealta de selecție „*Fuzzy Select*”. Prima opțiune din dialogul „*Tool options*”, opțiunea „*Mode*”, fiind comună tuturor uneltelor de desen, precum și straturilor, va fi tratată separat în altă secțiune a acestei lucrări de laborator.

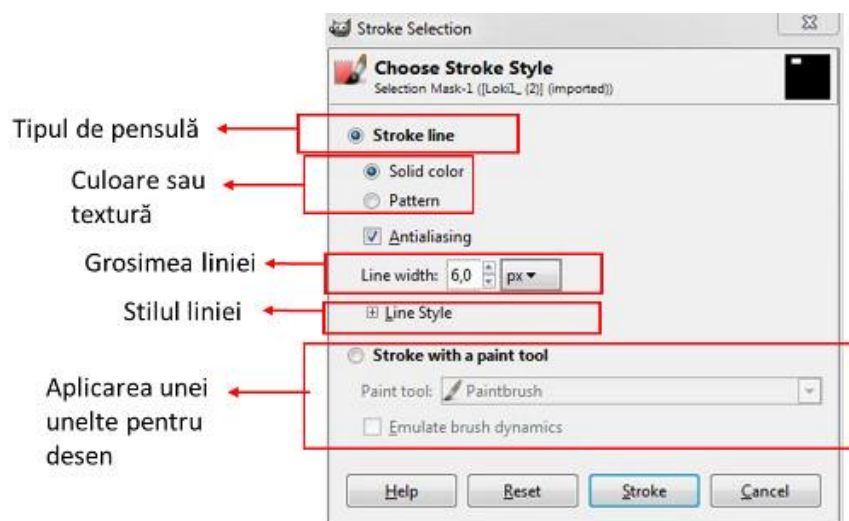


Figura 11 Opțiunile comenzii „Edit | Stroke Selection...”

2. Blend Tool (unealta gradient): Am folosit această unealtă în laboratorul trecut pentru a realiza un gradient de culoare. Opțiunea „Gradient” fixează tipul gradientului. Acesta poate fi de la FG (culoarea prim-planului) la BG (culoarea fundalului) sau invers. Aceeași opțiune fixează dacă trecerea de la o culoare la alta va fi făcută în linie dreaptă pe cubul RGB (Figura 10) sau în spirală pe cilindrul HSV. Există posibilitatea de a alege și alte tipuri de gradient, care nu au legătură cu ce culori sunt fixate pentru prim-plan și fundal. Următorul parametru reglabil este forma gradientului („Shape”). Cele 11 forme ale gradientului sunt exemplificate în meniul derulant al opțiunii. Următoarea opțiune („Repeat”) reglează modul de repetare al gradientului, iar „Offset” setează distanța de pe dreapta de gradient până la care culoarea să rămână constantă (și egală cu prima culoare a gradientului). După setarea tuturor opțiunilor, mai rămâne doar să trasăm cu mouse-ul o linie (printr-un *click and drag*) ce reprezintă dreapta de gradient. Înainte de punctul inițial al dreptei, spațiul selectat va fi umplut cu prima culoare, iar după punctul final al dreptei de gradient, spațiul selectat va fi umplut cu ultima culoare. În caz că opțiunea „Repeat” este selectată (precum în Figura 12), spațiul rămas după dreapta de gradient va fi umplut cu motivul repetitiv al gradientului.

3. Unealta creion („Pencil Tool”): După cum îi spune numele, această unealtă imită urma lăsată de un creion pe o foaie de hârtie. Pentru a desena o linie cu această unealtă (și cu oricare din uneltele prezentate mai departe) este suficient să apăsăm cu mouse-ul în locul de unde vrem ca linia să înceapă. Ținând tasta „Shift” apăsată, facem un al doilea click cu mouse-ul în punctul de sfârșit al liniei. În afară de opțiunea „Mode”, pe care o vom trata separat în această lucrare, unealta creion are atât un reglaj pentru opacitate („Opacity”), precum și un altul pentru tipul peniței („Brush”) și dimensiunea ei („Size”). Două alte proprietăți uzuale ale acestor unelte sunt unghiul („angle”) și raportul de formă („aspect ratio”). Acești ultimi doi parametri sunt explicați în Figura 13. Prima urmă din această figură este trasată cu un creion cu peniță (mină) circulară, raport de formă „0”, sub un unghi de 0°.

Folosim pentru a doua urmă aceeași peniță, însă cu un raport de formă pozitiv, ceea ce înseamnă că dimensiunea orizontală este mai mare decât dimensiunea verticală a peniței. Dacă am fi folosit un raport de formă negativ, raportul ar fi fost inversat. Al doilea parametru schimbat în Figura 13, atunci când trecem de la prima la a doua urmă, este unghiul peniței. Nimic nu ne împiedica să trasăm și a doua linie după direcția primei linii (verticală), însă pentru a distinge mai bine forma peniței, desenul a fost făcut urmând axa de simetrie a peniței.

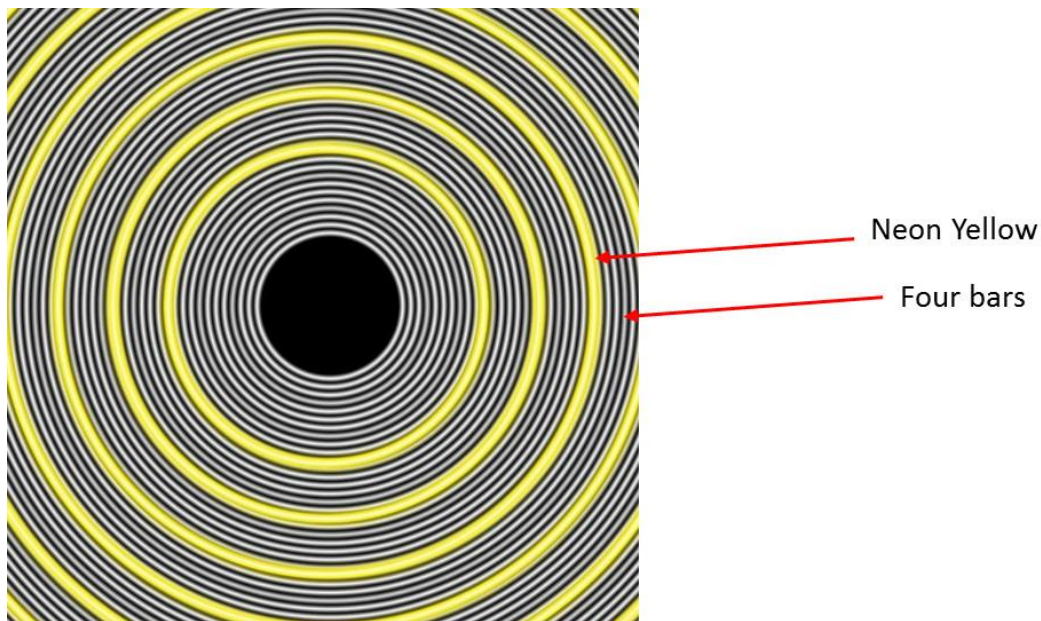


Figura 12 Aplicarea uneltei Blend Tool folosind două tipuri de gradient („Neon yellow” și „Four bars”)

4. Unealta pensulă („*Paintbrush Tool*”): Această unealtă are aceleași opțiuni și aceeași funcționalitate ca și unealta creion. Diferența este în urma lăsată pe foaie, mai lină, semănând mai mult cu o pensulă de pictat decât cu o mină de creion (a treia urmă din Figura 13).

5. Unealta spray („*Airbrush Tool*”): A patra urmă din Figura 13 a fost lăsată cu unealta spray. Este cea mai fină și oferă cea mai difuză urmă, comparativ cu oricare altă unealtă de desen. Are aceleași opțiuni de bază ca și cele două unelte de desen prezentate anterior, oferindu-ne posibilitatea de a-i regla în plus debitul de vopsea (prin opțiunile „*Rate*” și „*Flow*”).

6. Unealta stilou („*Ink Tool*”): Ultima linie trasată în Figura 13 este realizată cu unealta stilou, ale cărei funcții de bază sunt identice cu cele ale uneltelor discutate anterior.

7. Unealta gumă („*Eraser Tool*”): Cea din urmă unealtă pentru desen pe care o enumerăm în acest îndrumar este unealta de șters, care, în ciuda faptului că are aceleași opțiuni în comun cu unelte discutate anterior, realizează efectul contrar: ștergerea suprafeței desenate și

inlocuirea ei cu o regiune transparentă. Prin apăsarea tastei *Alt* în timpul ștergerii, putem restabili opacitatea unei regiuni anterior ștersă fără să vrem.

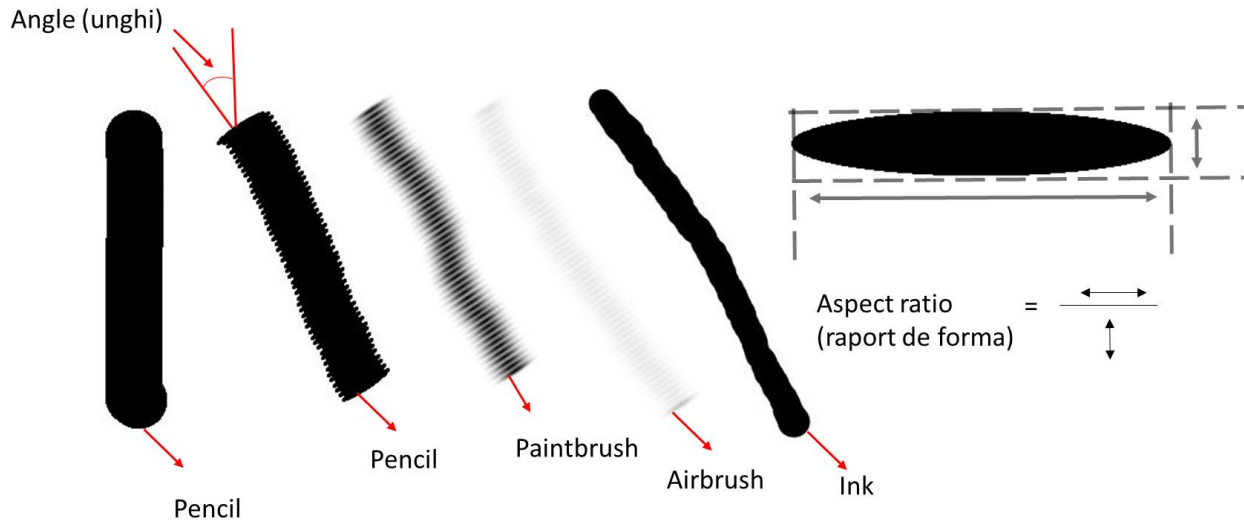


Figura 13 Unelele de pictură și opțiunile lor

Exerciții

- C. Folosindu-vă doar de unealta-gradient (Blend Tool) realizați o imagine cât mai asemănătoare cu cea din Figura 12.
- D. Folosindu-vă de unelele de desen prezentate în această secțiune, realizați o planșă asemănătoare cu cea din Figura 13, cu o singură deosebire: toate cele 5 dăre trasate trebuie să aibă direcția orizontală (rapotul de formă și unghiul peniței trebuie păstrat cât mai aproape de valorile din Figura 13).

Unealta Text

După cum îi arată numele, unealta text este folosită pentru adăugarea de text pe o imagine. Se accesează din toolbox (Figura 1) iar opțiunile sale sunt prezentate în Figura 14. Poziția textului pe imagine se alege prin apăsarea mouse-ului în locația dorită. Înainte însă de a selecta poziția și a introduce textul, utilizatorul trebuie să fixeze în dialogul ancorabil de opțiuni („*Tool options*”) parametrii uneltei (Figura 14):

- Tipul fontului („*Font*”): acest parametru fixează atât familia setului de caractere utilizat, cât și dacă simbolurile vor fi înclinate (*italic*) sau îngroșate (*bold*).

- Mărimea fontului („*Size*”): fixează dimensiunea (înălțimea) caracterelor utilizate. Dimensiunea poate fi scrisă în pixeli, milimetri, inches, etc.
- Culoarea („*Colour*”): culoarea fontului este inițial prestabilită la culoarea de prim-plan, însă poate fi schimbată cu o apăsare de mouse pe căsuța acesteia.
- Alinierea: analog parametrului din Word, stabilește dacă textul să fie aliniat la dreapta, la stânga, centrat, respectiv uniform repartizat între cele două margini laterale ale spațiului de scris.
- Spațiul de început: permite utilizatorului să aleagă distanța dintre primul caracter scris și extremitatea stângă a căsuței alocate scrisului.
- Spațiul dintre linii și spațiul între litere: permit utilizatorului să aleagă distanța dintre liniile, respectiv literele consecutive.

Unealta Traseu

Unealta traseu („*Paths tool*”) este cel mai puternic instrument de desen pe care GIMP-ul îl deține. La crearea sa au fost avute în vedere specificațiile matematice ale curbelor Bezier cubice. Practic fiecare traseu este o grupare de astfel de segmente curbate de tip Bezier. Dacă gruparea care formează un traseu este închisă, atunci traseul poate fi transformat într-o selecție prin comanda „Select | From Path”. Spre deosebire de selecții însă, traseele pot rămâne deschise (capetele traseului nu se unesc între ele). Fiecare segment din traseu este format din două ancure care definesc începutul și sfârșitul segmentului. Aceste ancure se fixează prin apăsarea butonului stânga al mouse-ului în pozițiile dorite. În dreptul fiecărei ancure, segmentul Bezier are un mâner. Depărtarea dintre mâner și ancoră determină curbura segmentului, iar direcția mânerului față de ancoră definește panta segmentului în dreptul capătului său. O linie dreaptă, având o curbura nulă, va avea așadar mânerule suprapuse peste ancure. Pentru a le face vizibile, este nevoie în acest caz de a crea prin altă metodă curbura dorită, prin tragerea cu mouse-ul de centrul segmentului. O ancoră care aparține la două segmente consecutive va avea două mânere (Figura 15). Pentru ca să nu existe nicio discontinuitate de direcție în dreptul unei ancure, trebuie ca unghiul dintre dreptele care unesc cele două mânere cu ancora să fie de 180°. O ancoră poate fi selectată și editată printr-un simplu click al mouse-ului. Atunci când una dintre ancure este selectată și apăsăm cu mouse-ul într-o poziție în care nu există nicio ancoră, vom crea o ancoră nouă precum și un nou segment de traseu care leagă noua ancoră de cea precedent selectată. Atunci când vrem să creem o buclă în traseu între două ancure, vom alege în mod normal una dintre ancurele de închidere, iar pe cea de a doua o vom selecta apăsând simultan tasta *Ctrl*.

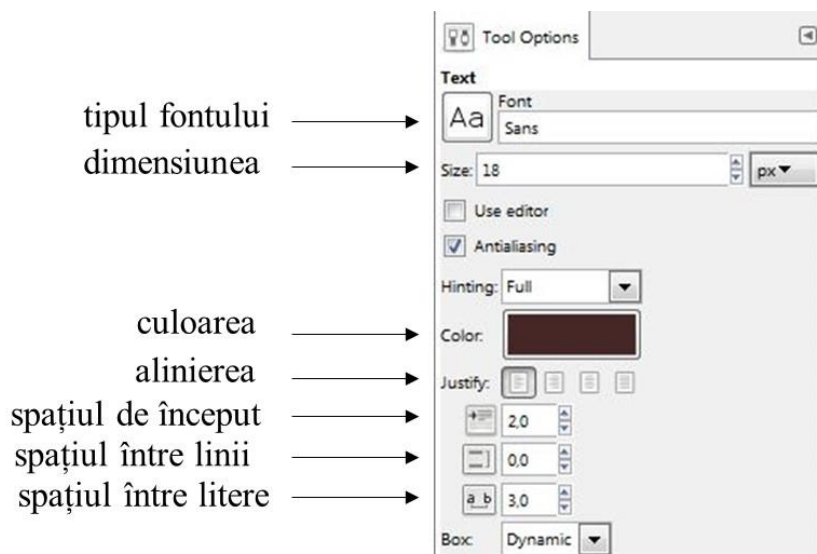


Figura 14 Opțiunile unelei Text

După ce creem un traseu și îl deselectionăm, acesta nu va mai fi vizibil în planșa de lucru, însă rămâne în continuare prezent în proiectul nostru. Pentru a vizualiza toate traseele create, este suficient să activăm dialogul ancorabil „*Paths*” și să marcăm traseele pe care le dorim reprezentate printr-un click pe prima coloană a liniilor acestora. Putem desena conturul fiecărui traseu într-un mod analog felului în care am procedat pentru selecții, folosind comanda „*Edit / Stroke Path*”. În felul acesta, chiar și când deselectionăm un traseu, conturul său va apărea întotdeauna în planșa de lucru.

Nu doar traseele sunt cele care se pot transforma în selecții, ci este posibil și procesul invers. Pentru aceasta este suficient ca după ce formăm selecția dorită, să apăsăm comanda din meniu „*Select / To Path*”.

Chiar și un obiect text poate fi transformat într-un obiect traseu. Pentru aceasta trebuie ca după scrierea testului utilizatorul să apese comanda „*Layer / Text to Path*”.

Exerciții

- E. Folosindu-vă de transformarea selecțiilor și textelor în trasee, precum și de unealta Traseu propriu-zisă, formați o schiță personalizată a deseneului din Figura 9b. După ce obțineți o schiță în genul celei din Figura 16, salvați proiectul GIMP în format .xcf pentru utilizarea lui ulterioară.

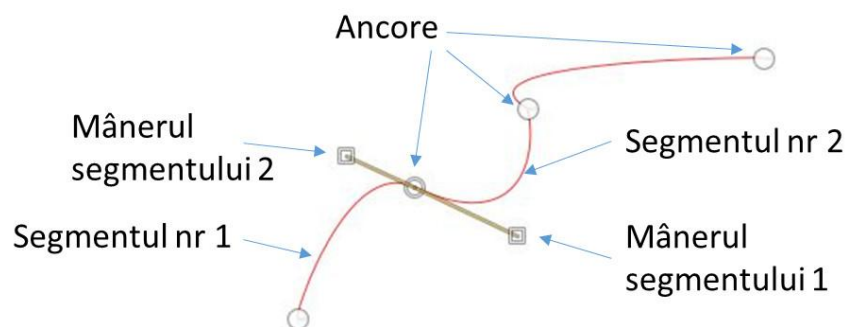


Figura 15 Construirea unui traseu



Figura 16 Schițarea imaginii din Figura 3.b folosind trasee

Imagini vectoriale

Traseele sunt strâns legate de conceptul de imagini vectoriale. Ele sunt singurele obiecte grafice ale GIMP-ului care nu sunt memorate și manipulate sub formă de matrici de pixeli. Reprezentarea lor este făcută sub o formă matematică: fiecare segment de traseu este reprezentat prin coordonatele ancorelor sale și prin vectorii formați de mânerul segmentului față de ancore. De fiecare dată când schimbăm scara la care privim imaginea, GIMP-ul (sau orice alt program capabil să afișeze grafică vectorială) va regenera traseul astfel încât să respecte formula sa matematică. Aceasta înseamnă că o imagine vectorială nu este afectată negativ de schimbarea scării, precum se întâmplă atunci când mărim o imagine de tip bitmap. Pentru a exporta traseele

create sub forma unei imagini vectoriale, utilizatorul trebuie să acceseze dialogul ancorabil „*Paths*”. Acolo poate selecta oricare dintre traseele din imagine (fie el vizibil sau nu), să apese butonul din dreapta al mouse-ului și să selecteze opțiunea „*Export path*”. În fereastra de dialog care se deschide trebuie introdus numele noului fișier (urmat de extensia .svg) și aleasă una din cele două opțiuni:

- „*Export the active path*”: exportă doar traseul curent selectat, cel pe care utilizatorul a făcut click dreapta pentru accesarea comenzii „*Export path*”.
- „*Export all paths from this image*”: exportă toate traseele din imagine (vizibile sau nu) în noul fișier de tip .svg.

Când exportăm traseele, imaginea vectorială astfel obținută nu va conține informații despre culorile, grosimea sau tipul de linie pe care le-am utilizat pentru conturarea lor. Toate imaginile vectoriale exportate din GIMP vor conține așadar doar contururile obiectelor obținute din trasee. Dacă vom folosi unelte de desen pentru a da culoare conturilor sau pentru a umple suprafețele delimitate de trasee, modificările vor apărea doar dacă exportăm imaginea într-unul din formatele bitmap (prezentate în laboratorul trecut). Există programe (de exemplu Inkscape sau Sodipodi) dedicate exclusiv graficii vectoriale 2D, care pot realiza mai multe manipulări grafice asupra obiectelor vectoriale. Felul în care este GIMP-ul construit permite să operăm în domeniul vectorial doar prin intermediul traseelor. Orice altă entitate de pe suprafața de lucru va fi reprezentată exclusiv în format matriceal (bitmap).

GIMP-ul este capabil nu doar să exporte imagini vectoriale, ci și să le importe. La deschiderea unei imagini de tip .svg, aveți grijă să bifați căsuța „*Import paths*” pentru a dispune de toate elementele grafice vectoriale ale acelei imagini sub formă de trasee. Dacă nu bifați această căsuță, GIMP-ul va transforma imaginea vectorială în format bitmap fără să importe niciun traseu. Toate modificările pe care le veți opera asupra traseelor și dimensiunilor imaginii pot fi exportate într-un nou fișier de tip .svg. Restul modificărilor pe care doriți să le operați la nivelul culorilor, texturilor sau altor elemente vizuale ale imaginii se vor reflecta doar în imaginea expotată de tip bitmap.

Exerciții

- F.** Deschideți proiectul GIMP salvat anterior la punctul 1.5.1. Salvați traseele din proiect sub forma unei noi imagini vectoriale, denumită „*schita.svg*”. Închideți proiectul și deschideți noua imagine salvată, bifând căsuța „*Import paths*”. Salvați imaginea în format .png. Deschideți ambele fișiere (*schita.svg* și *schita.png*) într-un browser web. Măriți cele două imagini la maxim (*Ctrl+Scroll up*) și comparați-le precum în figura de mai jos.

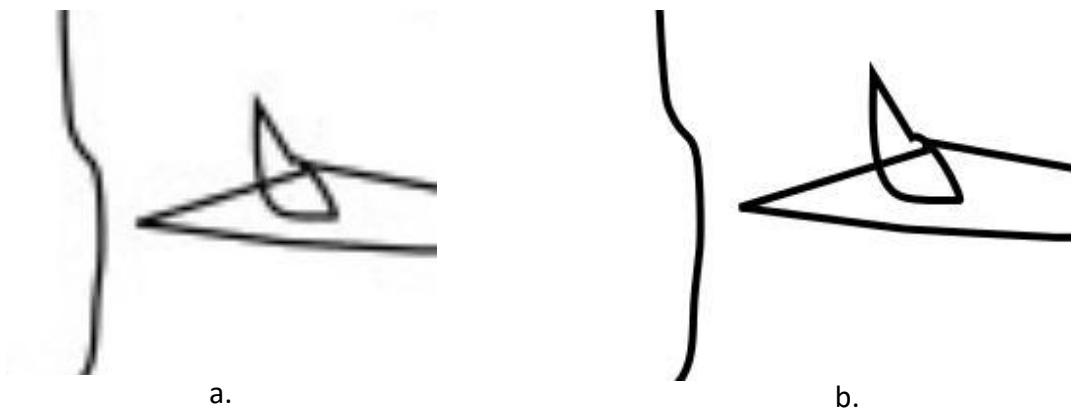


Figura 17 Comparație „la microscop” între o imagine bitmap (a.) și una vectorială (b). Ambele imagini reprezintă o regiune mărită a desenului din Figura 16, după ce am înregistrat-o în format png (a), respectiv în format svg (b).

Straturi și Moduri de superpoziție

O altă caracteristică puternică a GIMP-ului este oferită de straturi („layers”). Ceea ce un utilizator vede drept suprafața unei imagini multi-strat este de fapt superpoziția mai multor imagini diferite, fiecare contribuind cu o pondere și într-un mod fixat de utilizator. Pentru a vizualiza straturile unei imagini trebuie să deschidem dialogul ancorabil „Layers” (Figura 18). În lista de straturi astfel formată putem activa sau dezactiva efectul pe care un strat îl are asupra imaginii prin apăsarea markerului de vizibilitate al stratului, reprezentat de un ochi. În cazul în care markerul de vizibilitate este dezactivat, stratul respectiv va fi complet ignorat la formarea imaginii finale. O altă proprietate pe care o putem regla pentru fiecare strat este opacitatea („opacity”) care poate lua valori între 0 și 100. Observăm din nou prezența parametrului „Mode”, același parametru pe care l-am întâlnit și când am discutat uneltele de desen (Secțiunile 1.3.3 – 1.3.6) și pe care am amânat să-l descriem atunci. Acest parametru fixează felul în care pixelii stratului selectat se combină cu pixelii stratului de sub el (în stiva de straturi). Atunci când folosim o unealtă de desen peste o imagine, GIMP interpretează urma pe care aceasta o lasă peste imagine ca pe un nou strat ai cărui pixeli sunt automat introduși în imaginea de dedesubt conform modului de superpoziție ales. Pentru a descrie matematic modurile de superpoziție, vom nota valoarea unui pixel din stratul de dedesubt cu S de la substrat, iar valoarea intensității unui pixel din stratul de deasupra cu M , de la mască. Intensitatea pixelului rezultat din combinarea celor două straturi va fi notată cu F . Vom considera că atât stratul de dedesubt, cât și stratul de deasupra, au o valoare maximă a opacității. Modurile principale de superpoziție, atât pentru straturi, cât și pentru uneltele de desen, sunt:

- *Normal*: în care $F=M$. Observație: aceasta este doar o notație prescurtată. În realitate, fiecare pixel poate fi descris în spațiul RGB de trei valori. În cazul acesta, ecuația completă este: $F_R=M_R$ (pentru roșu), $F_B=M_B$ (pentru albastru), $F_G=M_G$ (pentru verde). Aceeași notație prescurtată va fi aplicată pentru descrierea tuturor celorlalte moduri.
- *Lighten only*: $F=\max(M, S)$.
- *Darken only*: $F=\min(M, S)$.

- *Addition:* $F = \min(M + S, 255)$.
- *Difference:* $F = |M - S|$.
- *Multiply:* $F = M \cdot S / 255$.
- *Divide:* $F = 255 \cdot S / (M + 1)$.
- *Screen:* $F = 255 - (255 - M)(255 - S) / 255$.
- *Overlay:* $F = S \cdot (S + 2 \cdot M \cdot (255 - S) / 255) / 255$.

După ce selectăm un strat putem, prin comenzile „Layer / Duplicate layer” sau „Layer / New layer” adăuga un strat identic cu el însuși, respectiv un strat transparent de dimensiuni echivalente. În Figura 18, imaginea inițială a fost duplicată în alte două straturi. Stratului al doilea i s-a aplicat un filtru alb-negru (comanda „Colors | Desaturate”) iar stratului de la suprafață un filtru de optimizare a culorilor (comanda „Colors | Auto | Equalize”). Opacitatea primelor două straturi a fost fixată la aproximativ 33%. Dacă dorim, putem schimba ordinea straturilor în stivă prin metoda *drag and drop*.

Exerciții

- G.** Deschideți fișierul „Laborator GAC/loki.jpg” și încercați să repetați pașii exemplificați pentru obținerea imaginii din Figura 18. Adăugați la sfârșit, folosind unealta Text, propria voastră semnătură pe imagine. Observați cum textul adăugat ocupă propriul strat în stiva de straturi. Mutați-i poziția în această stivă, punându-l pe rând în prima, a doua, a treia și a patra poziție. Ce observați?

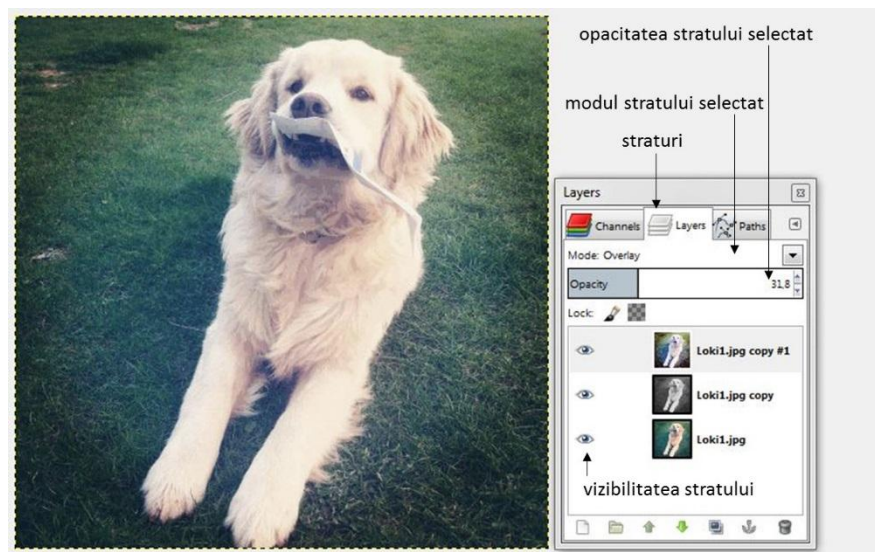


Figura 18 Folosirea straturilor si modurilor pentru schimbarea culorilor

- H.** Deschideți fișierul „Laborator GAC/loki.jpg”. Duplicați imaginea de bază într-un nou strat. Aplicați peste noul strat un filtru alb-negru. Modificați modul de superpoziție dintre cele două straturi, încercând toate variantele discutate în secțiunea 3. Salvați de fiecare

dată imaginea obținută într-un nou fișier cu numele „*loki-xyz.jpg*” (în numele fișierului *xyz* trebuie înlocuit cu numele modului de superpoziție).

Desfășurarea lucrării

1. Familiarizați-vă cu toate uneltele din Toolbox-ul GIMP-ului și rezolvați toate exercițiile din Secțiunea 1 a lucrării: A, B, C, D, E.
2. Ce este aceea o imagine vectorială și cum se diferențiază aceasta de o imagine bitmap? Rezolvați exercițiul F.
3. Familiarizați-vă cu manipularea straturilor unei imagini și rezolvați exercițiile G și H din secțiunea precedentă.
4. Deschideți în GIMP o poză color. Păstrați color obiectul principal al pozei, în timp ce restul fotografiei (fundalul) transformați-l în alb-negru. Indicație: se folosește o unealtă de selecție pentru a selecta obiectul (precum în Figura 9) apoi se folosește comanda „Select | Invert” pentru a selecta fundalul, care ulterior trebuie desaturat (comanda „Colors | Desaturate”).
5. Faceți un colaj fotografic folosindu-va de obiecte găsite în cel puțin două imagini diferite (fotografii proprii sau de pe internet). Colajul trebuie să poarte un mesaj cu impact social.
6. Creați o nouă imagine. Impărțiți foaia de desen în 4 linii și 4 coloane. Desenați folosind un creion negru pe un fundal alb, pe aceeași linie dar pe coloane diferite, un cerc, un pătrat, o elipsă cu raportul semiaxelor egal cu 3, respectiv un dreptunghi cu raportul lungime/lățime egal cu 2. Pe prima linie folosiți unealta creion (*Pencil*) pentru a desena cele patru forme, pe a doua linie folosiți unealta pensulă (*Paintbrush*), pe a treia linie folosiți unealta spray (*Airbrush*), iar pe ultima linie stiloul cu cerneală (*Ink*). Lăsați toate formele neumplute.
7. Realizați imaginea vectorială a unei elipse. Figurați în interiorul fișierului .svg salvat de dumneavoastră semi-axele elipsei folosind săgeți și notați-le cu literele „a”, respectiv „b”.
8. Salvați de pe internet poza de profil a unei persoane publice și realizați-i o caricatură vectorială personalizată, asemănător cu modul de operare de la exercițiul E.

Capitolul 3. Funcții speciale și tehnici de prelucrare automată cu GIMP

Obiectivul lucrării

Lucrarea prezintă constituie un studiu aprofundat al tuturor uneltelor GIMP-ului precum și al unor funcții și filtre speciale. Sfârșitul lucrării prezintă noțiunile de bază necesare studentului pentru realizarea primului său program de prelucrare automată a imaginilor folosind Script-Fu. La sfârșitul acestui laborator, studentul va fi capabil să realizeze o animație de tip *.gif* atât manual precum și automat cu Script-Fu. Studentul va putea de asemenea să realizeze o pagină web simplă folosind maparea linkurilor pe o imagine cu ajutorul GIMP-ului.

Breviar teoretic

Uneltele de culoare

Am văzut în laboratorul precedent cum putem folosi în GIMP uneltele de selecție, uneltele pentru pictură și desen, unealta traseu și unealta text. Un alt set foarte important de unelte este grupul uneltelor de culoare. Spre deosebire de grupurile de unelte pe care le-am folosit în laboratorul precedent, acest grup de unelte grafice nu poate fi accesat direct din Toolbox. Il putem însă accesa prin cele două modalități descrise în Figura 1, fie prin intermediul meniului „*Tools / Color Tools*” (a), fie direct prin intermediul meniului „*Colors*” (b). Această secțiune va fi dedicată în continuare prezentării acestor unelte.

Balanța de culori (Color Balance)

Această unealtă este cel mai simplu accesată din meniul „*Colors / Color Balance*”. Odată deschis dialogul său de opțiuni, putem selecta căror pixeli să le fie modificată culoarea folosindu-ne de opțiunea „*range to adjust*”. Putem în acest fel să modificăm pixelii cei mai închiși („*Shadows*”), pe cei medii („*Midtones*”) sau pe cei mai luminoși („*Highlights*”). Este în general indicat să nu modificăm cei mai deschiși sau cei mai închiși pixeli, pentru a nu schimba aspectul pixelilor albi și negri. În acest fel, deși putem modifica destul de mult pixelii de luminozitate medii, putem păstra aspectul natural al fotografiei. În Figura 8 de exemplu am modificat imaginea originală „*Laborator GAC/loki.jpg*” (a) pentru a schimba culoarea predominantă a imaginii din verde în albastru (b), respectiv roșu (c). Ajustarea culorilor se realizează pe cele 3 axe de culori RGB, fiecare axă având la un capăt una dintre culorile roșu, verde sau albastru, iar la celălalt capăt complementele lor. Bifarea opțiunii „*Preserve luminosity*”

va asigura păstrarea luminozității fiecărui pixel. Prin selectarea și deselectarea opțiunii „Preview” putem alterna între imaginea originală și imaginea modificată.

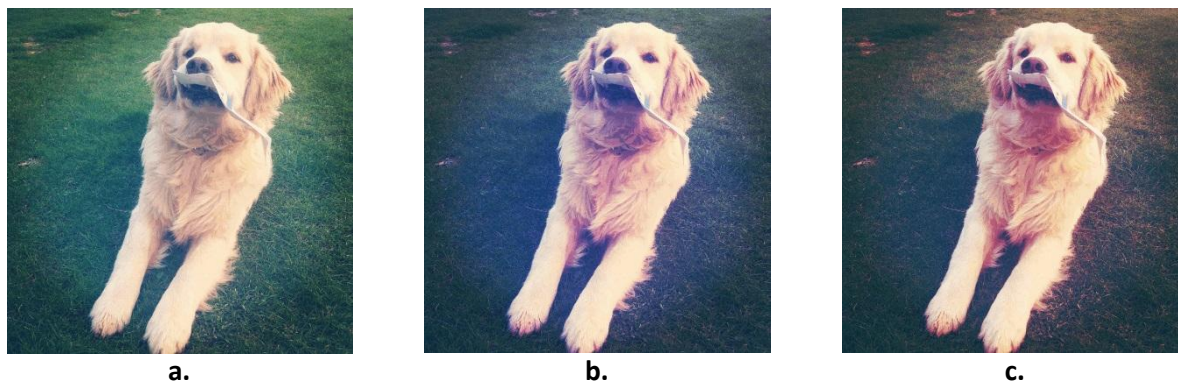


Figura 19 Folosirea unelei “Color Balance” pentru modificarea echilibrului culorilor: a. imaginea originală; b. imaginea deplasată spre albastru; c. imaginea deplasată spre roșu

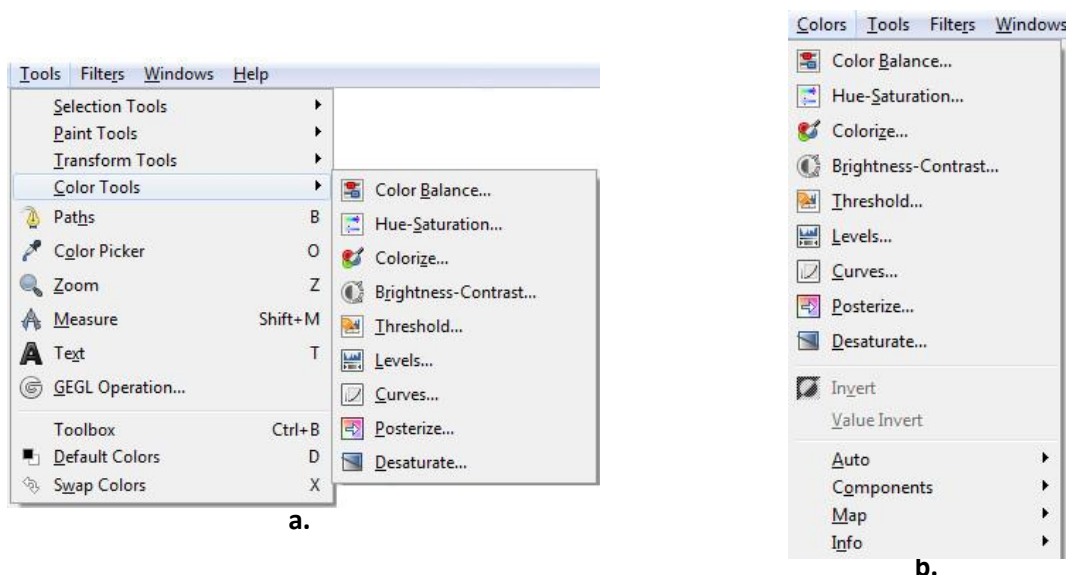


Figura 20 Accesarea uneltelor de culoare GIMP: a. prin intermediul meniului Tools; b. direct prin intermediul meniului Colors

Nuanță și Saturație (Hue-Saturation)

Această unealtă poate fi folosită pentru schimbarea nuanței culorilor. Spre deosebire de funcția „Color Balance” prezentată precedent, selecția pixelilor a căror nuanță va fi schimbată este făcută pe baza culorilor (prin intermediul opțiunii „Primary Color to Adjust”), nu a luminozității. Cercul culorilor pe baza căruia vom face selecția pixelilor este alcătuit din 6 zone, fiecare zonă corespunzând uneia din culorile de bază (R – roșu, G – verde, B – albastru) sau uneia din culorile lor complementare (C – azuriu, M – purpuriu, Y – galben).

După selecția culorii a fi schimbate, vom opera modificările prin deplasarea cursorului „Hue”. Modificarea este exprimată în grade, iar aceasta reprezintă unghiul deplasării culorii selectate pe cercul culorilor. Atunci când unghiul are valoare pozitivă, deplasarea se face în sens invers acelor de ceasornic, iar când unghiul are valoare negativă, în sens direct. Rezultă așadar că atât pentru o rotație maximă de 180° cât și pentru una inversă de -180° , culoarea selectată se va transforma în complementul ei.

Prin deplasarea cursorului „Lightness” putem micșora sau mări luminozitatea pixelilor selectați, iar prin deplasarea cursorului „Saturation” putem regla intensitatea culorii. Pentru o saturație de 100 culoarea selectată va avea o intensitate maximă, iar pentru o saturație de -100, pixelii selectați vor fi reprezentați doar în nuanțe de gri.

Folosind așadar doar opțiunile disponibile în unealta „Hue-Saturation” putem recrea efectul obținut anterior folosind unealta „Color-Balance”. Noul set de imagini este reprezentat în Figura 21.

Niveluri (Levels)

Această unealtă poate fi folosită pentru a închide sau deschide o imagine, pentru schimbarea contrastului sau pentru corecția culorilor. Activarea acestei unelte se face din meniul „Tools / Color Tools” (Figura 1a) sau din meniul „Colors” (Figura 1b). Imediat după activare vom putea observa o fereastră de dialog precum cea din Figura 22.

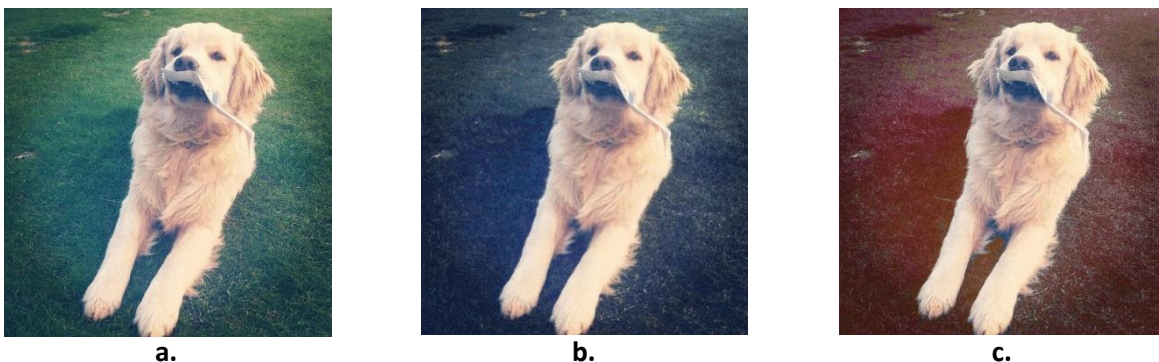


Figura 21 Folosirea uneltei „Hue-Saturation” pentru modificarea echilibrului culorilor: a. imaginea originală; b. imaginea deplasată spre albastru; c. imaginea deplasată spre roșu

Modificările pe care le operăm cu această unealtă se vor face în general separat pentru fiecare canal în parte. Primul lucru pe care îl vom face așadar după activarea uneltei este să selectăm canalul pe care îl dorim modificat, folosind meniul derulant „Channel” din Figura 22. Histograma canalului selectat este reprezentată imediat sub meniul derulant de selecție.

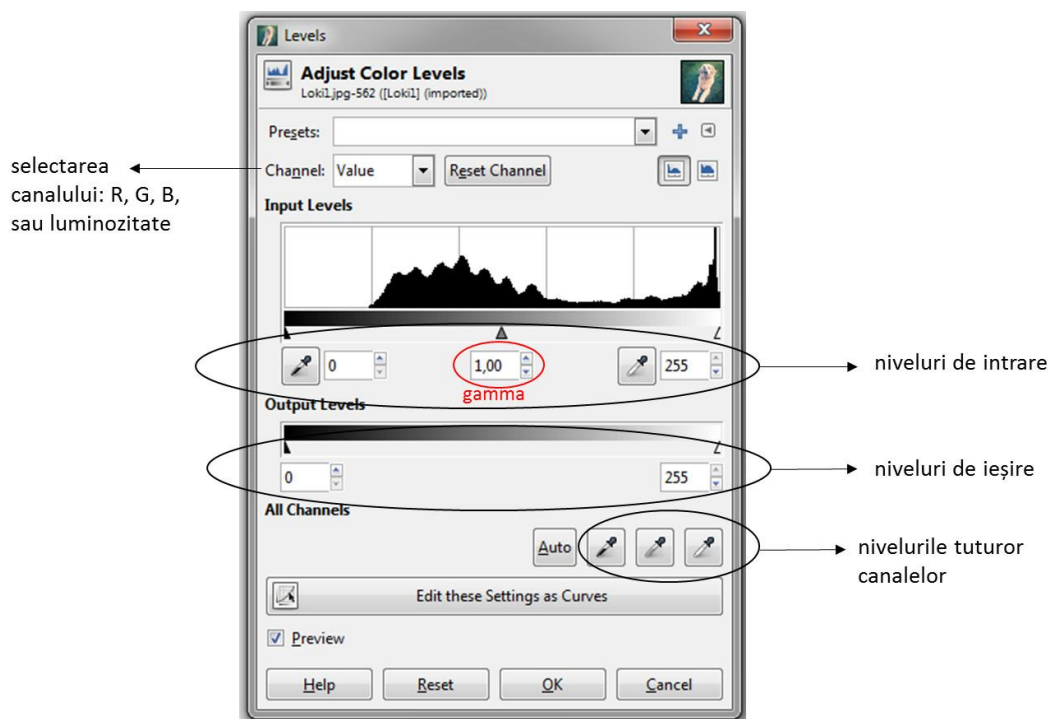


Figura 22 Dialogul folosit pentru reglarea unei „Niveluri”(Levels)

Nivelurile de intrare pot fi reglate în trei moduri: fie prin intermediul triunghiurilor mobile de sub histogramă, fie prin intermediul valorilor numerice aflate în căsuțele de sub histogramă, fie prin intermediul uneltelor de extracție a culorilor. Nivelurile de ieșire pot fi reglate doar în două moduri: folosind triunghiurile mobile și folosind căsuțele numerice din dreptul acestora.

După selecția limitelor nivelurilor de intrare și de ieșire, această unealtă va transforma valoarea fiecărui pixel având un anumit nivel de intrare în nivelul de ieșire proporțional corespunzător. De exemplu, să presupunem că selectăm canalul roșu (red). La intrare alegem limita inferioară la 30 și limita superioară la 200, iar la ieșire fixăm limita inferioară la 0, iar limita superioară la 250. Atunci toți pixelii care aveau inițial valoarea canalului roșu inferioară lui 30 vor avea după transformare o valoare a canalului roșu nulă, iar toți pixelii care erau superiori lui 200 vor avea o valoare a canalului roșu de 250. Toate valorile de roșu intermediare, cuprinse între 30 și 200, se vor transforma, după o lege liniară, în valori de roșu în intervalul 0 – 250. Transformarea valorilor între nivelurile de intrare și cele de ieșire se va face după o lege liniară doar dacă parametrul gamma (parametrul notat cu roșu în Figura 22) corespunzător nivelurilor de intrare este egal cu 1. În cazul că acest parametru este supraunitar, valorile pixelilor pe canalul ales vor fi mai mari decât pentru cazul liniar (imaginea se va lumina), iar în cazul invers aceste valori vor fi inferioare cazului liniar (imaginea se va închide).

Reglaje automate (Auto)

Meniul „Colors” conține, pe lângă uneltele de culoare discutate mai sus, și un submeniu destinat reglajelor automate. Acestea pot îmbunătăți net imaginea în multe situații, dar efectul pe care-l produc variază mult de la o imagine la alta. În unele situații aceste reglaje pot produce o înrăutățire a imaginii procesate. Cele mai folosite reglaje automate sunt:

- *Equalize*: Acționează asupra distribuției intensității fiecărui canal în parte (prin aplicarea uneltei „Curves” asupra fiecărui canal) pentru ca histograma luminozității să fie cât mai plată pe tot intervalul de definire (0-255).
- *White Balance*: Extinde caracteristica fiecărui canal (prin aplicarea uneltei „Levels” pe fiecare canal în parte) până ce acestea ocupă în întregime intervalul lor de definire.
- *Normalize*: Acționează asupra distribuției luminozității imaginii (prin aplicarea automată a uneltei „Levels” pe canalul luminozității) pentru ca cel mai întunecat pixel din imagine să devină negru absolut (#000000) iar cel mai luminos pixel din imagine să devină alb absolut (#FFFFFF). Nu schimbă nuanța culorilor imaginii, deoarece normalizarea se face doar pe canalul luminozității.

Filtre speciale

Filtrele speciale (Figura 16) reprezintă o colecție de funcții matematice complexe ce pot fi aplicate pe imagine sau pe o regiune selectată din imagine. Aceste funcții au aplicații foarte diverse, de la încetoșarea imaginii până la re-pictarea acestuia folosind tehnici specifice unui anume curent artistic. Toate comenzile destinate aplicării acestor filtre speciale pe imagine se găsesc în meniul „Filters”. Cele mai uzuale dintre acestea sunt enumerate în Figura 16.

Reducerea și îmbunătățirea clarității

Reducerea clarității unei imagini se face folosind una din comenzile meniului „Filters / Blur”. Dacă doriți doar o ușoară încetoșare a regiunii selectate, atunci cea mai rapidă cale este prin folosirea comenzii simple „Blur” (Figura 16). Pentru un efect mai puternic este recomandată folosirea comenzii „Gaussian Blur”. Atât prima, cât și a doua comandă, produc o încetoșare omnidirecțională. Dacă se dorește o încetoșare după o anumită direcție trebuie folosită comanda „Motion Blur”.

Procesul invers, de îmbunătățire a clarității unei imagini, este realizat prin una din comenzile meniului „Filters / Enhance” (Figura 16). Comenzile „Sharpen” și „Unsharpen Mask” produc efectul invers filtrului „Blur”. Cea mai eficientă dintre ele este „Unsharpen Mask”. Pentru a preveni schimbările de nuanță ce pot apărea în timpul aplicării acestei comenzi, este indicat ca imaginea să fie descompusă în componentele ei HSV (prin comanda „Colors / Components / Decompose” și selectarea modelului de culoare „HSV”) iar filtrul de îmbunătățire a clarității „Unsharpen Mask” să fie aplicat doar asupra componentei V (Value). Reconstrucția

imaginii se realizează ulterior prin comanda inversă „Colors / Components / Compose”. Dacă nu dorim ca fiecare tranziție lină în imaginea inițială să fie accentuată, atunci putem mări pragul de activare („Threshold”) din dialogul comenzii „Unsharpen Mask”. Puterea filtrului este reglată din parametrul „Amount”.

Filtrul „Deinterlace” (Figura 16) elimină defectele (liniile lipsă) în imaginile capturate după o sursă video cu baleiaj intercalat (*interlaced*). Filtrul „Despeckle” elimină zgomotul granulat care apare atunci când creștem foarte mult claritatea unei imagini sau când realizăm o fotografie digitală folosind un ISO mare. Filtrul „Destripe” elimină dungile verticale parazite.

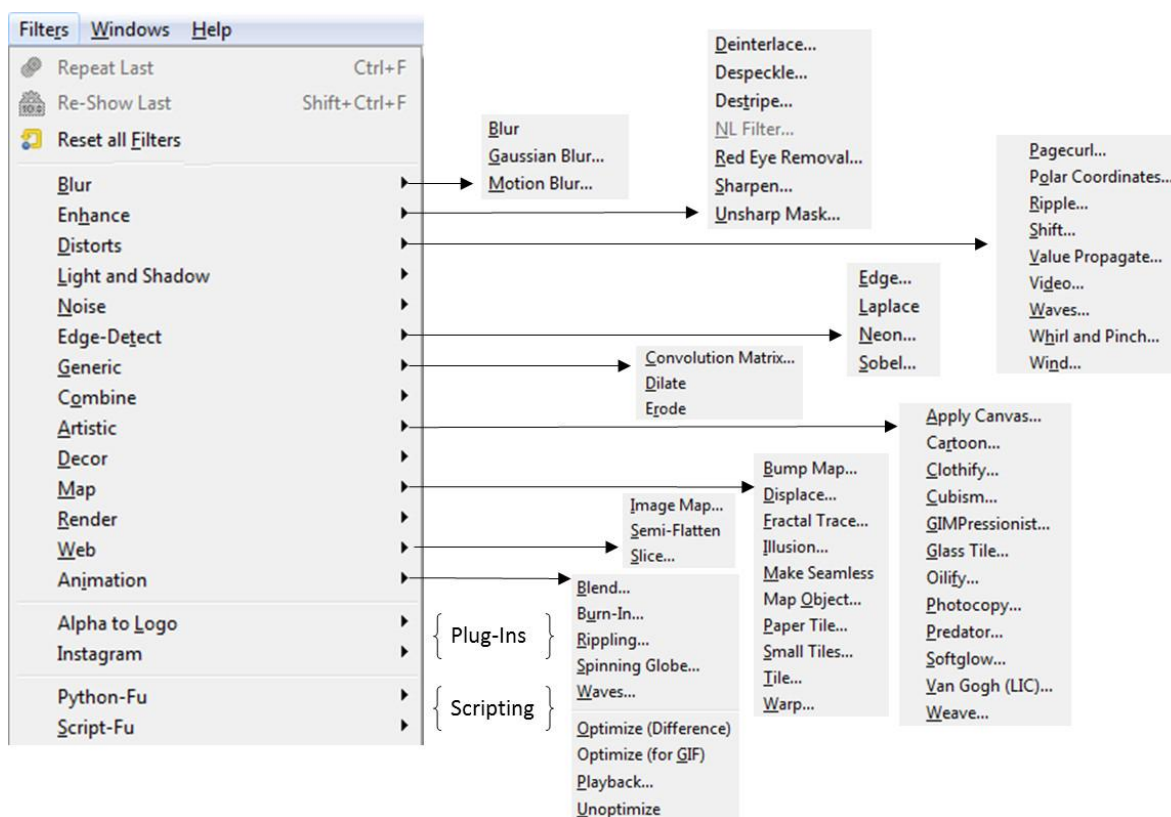


Figura 23 Meniul filtrelor speciale GIMP

Detecția marginilor și convoluția cu matrici

Pentru conturarea obiectelor dintr-o imagine putem folosi oricare dintre algoritmi de detecție a conturilor, prezenți în meniul „Filters | Edge-Detect”. Cei mai importanți dintre aceștia sunt enumerați în Figura 16: *Edge*, *Laplace*, *Neon*, *Sobel*. Toate aceste filtre funcționează în general prin calcularea derivatei intensității pixelilor din imagine. Filtrul *Laplace* se bazează pe calcularea derivatei de ordinul 2 și nu este în general indicat pentru detecția conturului, dar este eficient pentru mărirea contrastului.

Putem așadar să construim conturul unei imagini mult mai simplu decât am făcut-o în laboratorul trecut prin intermediul uneltelor de selecție. Să luăm exemplul din Figura 24a, unde s-

a folosit filtrul „Edge” pentru conturarea subiectului din Figura 8a. Filtrul „Edge” a fost aplicat folosind parametrii standard, după conversia imaginii în nuanțe de gri („Colors | Desaturate”). Dacă micșorăm parametrul „Amount” al filtrului „Edge” vom putea detecta doar schimbările abrupte de intensitate, iar conturul va fi desenat folosind linii fine. Invers, prin creșterea acestui parametru vom putea detecta toate schimbările de intensitate, iar conturile obiectelor vor fi îngroșate.

Atât funcțiile de desenare a conturilor, precum și cele de micșorare sau mărire a clarității prezentate în această secțiune, rezultă din înmulțirea matricii-imagine cu o matrice mobilă (kernel) de ponderi care se plimbă pe toată suprafața imaginii. Această operație se numește convoluție și poate fi comandată de utilizator folosind filtrul „Filters | Generic | Convolution Matrix”. După cum se poate vedea în Figura 25, dimensiunea maximă a matricii pe care o putem folosi drept kernel este de 5x5, dar este la latitudinea utilizatorului să aleagă care elemente ale acestei matrici să fie folosite și care nu.



Figura 24 Desenarea conturului unui obiect cu GIMP: a. folosind filtrul Edge; b. după aplicarea operatorului de dilatație „Filters | Generic | Dilate”.

Luminozitatea fiecărui pixel în noua imagine ($B'_{j,i}$) va fi egală cu suma luminozităților pixelilor vecini din vechea imagine $B_{j+y,i+x}$ ponderate cu elementele matricii de convoluție ($K_{y,x}$), conform formulei:

$$B'_{j,i} = \frac{\sum_{y=-2}^2 \sum_{x=-2}^2 K_{y,x} \cdot B_{j+y,i+x}}{Divizor} + Offset \quad (1)$$

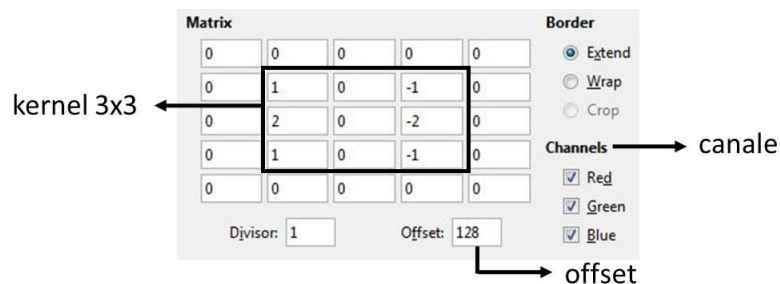


Figura 25 Folosirea matricii de convoluție pentru calcularea gradientului pe direcție orizontală.

Dacă fixăm elementele matricii de convoluție precum în Figura 25, atunci vom realiza o derivare a imaginii pe direcție orizontală. Putem repeta operația înlocuind matricea kernel din Figura 25 cu transpusa ei, pentru a realiza derivarea pe direcție verticală. Pentru a accentua efectul derivării, putem înlocui divizorul 1 în fereastra de dialog din Figura 25 cu un divizor subunitar.

Efecte artistice și speciale

În meniul GIMP rezervat filtrelor este dedicat un număr foarte mare de comenzi efectelor artistice (Figura 16) și efectelor speciale. Comenzile submeniului „Filters | Distorts” pot genera o gamă largă de distorsiuni geometrice asupra imaginii. Cele din submeniul „Filters | Light and Shadow” sunt destinate adăugării de surse de lumină sau de umbre pe imagine.

Câteva din cele mai importante filtre ale submeniului „Filters | Artistic”, care este destinat realizării de efecte artistice pe imagine, sunt următoarele:

- „Apply canvas”: modifică textura imaginii în așa fel încât să creeze impresia că are ca suport o pânză de pictură.
- „Cartoon”: modifică imaginea pentru a crea impresia că face parte dintr-un desen animat.
- „Clothify”: modifică textura imaginii, creând impresia de țesătură
- „Cubism”: modifică imaginea ținând cont de tehnicile curentului artistic cubist
- „GIMPpressionist”: este cel mai complet efect artistic. În fereastra de opțiuni a acestui efect utilizatorul poate regla textura suprafeței de pictură (tab-ul „Paper”), tipul de pensulă folosit (tab-ul „Brush”), variația unghiului de orientare al pensulei (tab-ul „Orientation”), mărimile pensulelor folosite (tab-ul „Size”), etc.

Submeniul „Filters | Decor” conține o colecție de scripturi (*Script-Fu*) destinate adăugării de margini și a altor câteva artefacte interesante pe imagine. Comenzile submeniului „Filters | Map” pot realiza o suprapunere a imaginii curente peste o formă predefinită sau chiar în mai multe poziții ale aceleiași imagini. Putem suprapune imaginea curentă chiar și peste o sferă, folosind comanda „Filters | Map | Map Object” și selectând „Sphere” în dreptul opțiunii „Map to”.

Filtrele din submeniul „Filters | Render” se diferențiază de celelalte filtre prin faptul că nu procesează conținutul aflat deja în stratul (layer-ul) selectat, ci creează noi forme din nimic, de cele mai multe ori ștergând complet conținut precedent al stratului. Este de aceea indicat ca aceste filtre (care ar trebui numite mai degrabă unelte) să fie aplicate deasupra unui strat nou.

Aplicații web

Harta unei imagini

Cele trei filtre ale submeniului „Filters | Web” sunt destinate imaginilor folosite în aplicațiile web. Primul dintre acestea, „Image Map” este utilizat la realizarea hărții unei imagini. Această hartă (descrisă în cod html) delimitează regiunile „calde” dintr-o imagine web. Regiunile calde sunt acele regiuni ce pot răspunde în mod independent de restul imaginii unui eveniment al mouse-ului. Un click așadar pe zona corespunzătoare unei astfel de regiuni poate deschide o nouă adresă web sau poate declanșa o funcție javascript specifică. O serie de alte schimbări pot însoți de și alte evenimente ale mouse-ului, cum ar fi modificarea cursorului mouse-ului la trecerea sa peste regiunea delimitată.



Figura 26 Crearea unei hărți de imagine pentru o pagină web folosind filtrul Filters | Web | Image Map.

Operarea cu acest filtru se realizează într-o planșă nouă de lucru. În Figura 26 putem vedea rezultatul aplicării filtrului pe imaginea din Figura 24a. În partea stângă a acestei figuri apar uneltele folosite pentru demarcarea regiunilor calde din imagine. Cea mai precisă demarcare poate fi realizată cu unealta poligon (ultima unealtă încercuită în Figura 26). Imediat după demarcarea unei regiuni, pe ecran se va deschide o nouă fereastră responsabilă cu fixarea proprietăților noii regiuni demarcate. În dreptul opțiunii „URL to activate...” trebuie trecută adresa către care browserul utilizatorului va fi trimis ca urmare a unui click pe regiune. În dreptul tab-ului „JavaScript” putem fixa acțiunile browserului în cazul altor evenimente ale mouse-ului.

După demarcarea tuturor regiunilor de interes rezultatul va fi salvat într-un fișier text folosind iconița dedicată din Figura 26. Pentru a testa rezultatul se poate crea o pagină .html care să conțină doar imaginea și care să includă conținutul fișierului hartă.

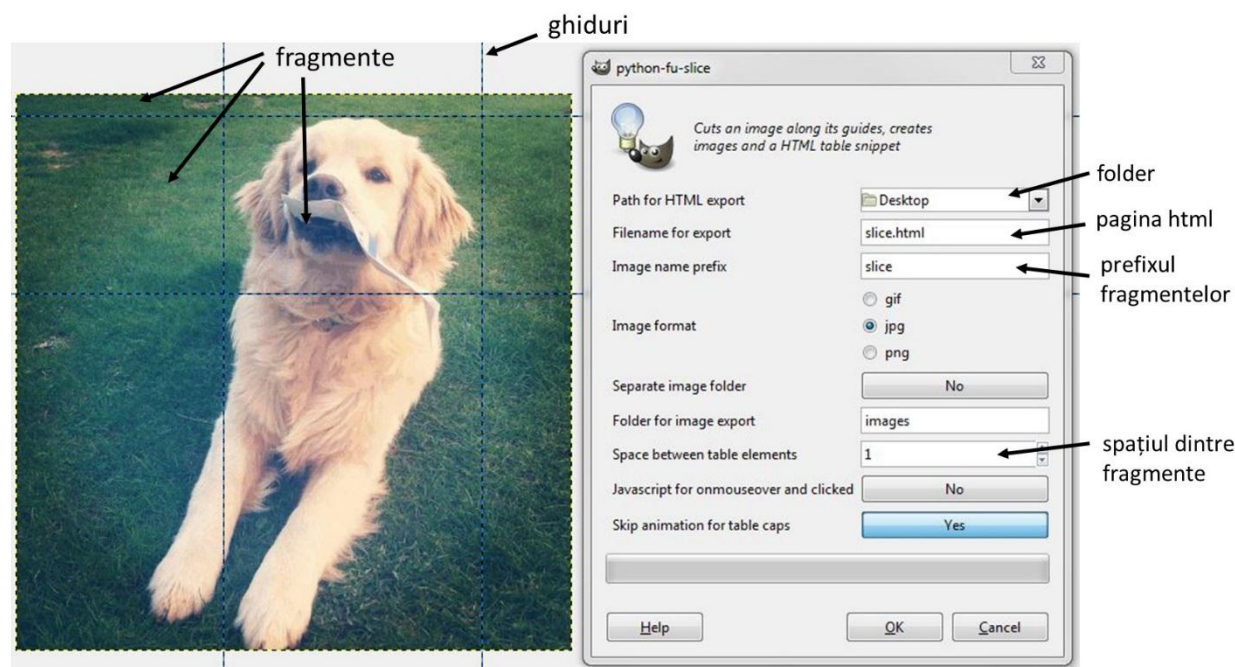


Figura 27 Tăierea unei imagini în fragmente pentru web folosind filtrul „Filters | Web | Slice”.

Tăierea unei imagini

O imagine poate fi tăiată în fragmente folosind comanda „Filters | Web | Slice”. Fragmentele vor fi automat aranjate într-un tabel html pentru ca imaginea să poată fi reconstituită în integralitate într-o pagină web. Fiecare fragment-imagine astfel creat acționează total independent față de celelalte fragmente și poate avea proprietăți html total diferite de ale celorlalte porțiuni din imagine. De exemplu un singur fragment dintr-o imagine gif poate fi animat, sau un singur fragment poate avea transparență redusă, sau putem încercui cu un cadru fragmentul peste care are loc trecerea mouse-ului, etc. Înainte însă de a rula filtrul „Slice” este nevoie să divizăm imaginea folosindu-ne de comenzile disponibile în submeniul „Image / Guides”. Să luăm exemplul prezentat în Figura 27. Întâi s-a realizat selectarea capului câinelui folosind unealta de selecție rectangulară. Ulterior s-au trasat ghidurile folosind comanda „Image / Guides / New Guides from Selection”. Ghidurile realizează demarcarea fragmentelor.

Pentru a face fragmentarea propriu-zisă a imaginii, putem porni acum filtrul „Image / Guides | Slice”. În fereastra de dialog care se va deschide trebuie introdus numele folderului unde întreaga structură a noii pagini web va fi copiată, urmat de numele paginii web. Pentru cele 9 fragmente ale imaginii din Figura 27 trebuie apoi introdus un identificator text comun, care se va constitui în prefixul numelui tuturor celor 9 imagini independente create. Se mai poate fixa și

spațiul dintre celulele tabelului ce vor conține fragmentele din imagine. Dacă este selectată opțiunea „*Skip animation for table caps*” atunci fragmentele situate la frontiera imaginii nu vor răspunde la evenimentele mouse-ului asupra lor.

Animații gif

GIMP ne oferă posibilitatea de a realiza animații .gif prin intermediul submeniului său „*Filters / Animation*”. Să presupunem că imaginea noastră inițială are un singur strat. Putem să folosim una din comenzile „*Rippling*”, „*Spinning Globe*” sau „*Waves*” din meniul „*Filters / Animation*” pentru generarea mai multor straturi care ulterior vor constitui cadrele animației noastre. În fereastra de dialog care se va deschide vom fixa numărul cadrelor pe care dorim ca GIMP-ul să le genereze. Imaginea pe care GIMP-ul o va deschide într-o nouă fereastră după apăsarea butonului „*Ok*” va avea numărul cadrelor dorit și va putea fi imediat salvată în format .gif folosindu-ne de comanda „*File | Export*”. Atenție! Nu uitați să denumiți noul fișier folosindu-vă de extensia „.gif”. În următorul dialog destinat exportului în format „gif” vor trebui selectate opțiunile „*As animation*” și „*Loop forever*”. Puteți de asemenea testa animația înainte de a o exporta folosindu-vă de comanda „*Filters / Animation | Playback*”.

Script-Fu

Script-Fu este un limbaj de scripting derivat din limbajul Scheme care poate fi folosit pentru generarea automată a unui număr mare de operații grafice. Este în special util când se dorește repetarea identică a unei suite de operații pe mai multe imagini. Prin folosirea unui script în locul repetării manuale a operațiilor putem câștiga timp și putem de asemenea împărtăși foarte rapid tehnică grafică altor utilizatori. Se pot de asemenea downloada de pe internet filtre grafice create de alți utilizatori GIMP, fie sub formă de Script-Fu, fie sub formă de Plug-Ins (scrise în general într-un limbaj de programare precum C, folosind librării specifice). În cazul în care ați downloadat un script, acesta trebuie copiat în folderul dedicat, care poate fi găsit folosind comanda „*Edit | Preferences*”, urmată de selecția opțiunii „*Folders | Scripts*”. În cazul unui Plugin, acesta trebuie copiat în folderul selectat de opțiunea „*Folders | Plug-Ins*”. Pentru actualizarea listei de scripturi este nevoie de rularea comenzii „*Filters | Script-Fu | Refresh Scripts*”.

Există două categorii de scripturi: cele care creează o nouă imagine (pe care în general le putem găsi în submeniul „*File | Create*”) și cele care operează pe imaginea deja încărcată (pe care le putem găsi în interiorul submeniului „*Filters*”). Ambele categorii de scripturi pot fi găsite pe disc la aceeași locație (fixată de opțiunea „*Folders | Scripts*” discutată în paragraful precedent). Dacă deschidem locația respectivă de pe disc vom găsi lista tuturor scripturilor instalate. Avem posibilitatea să vedem codul oricăruia dintre ele folosind un editor de text

(*Notepad++*). Putem așadar copia codul oricărui script deja existent și crea propriul nostru script plecând de la acesta.

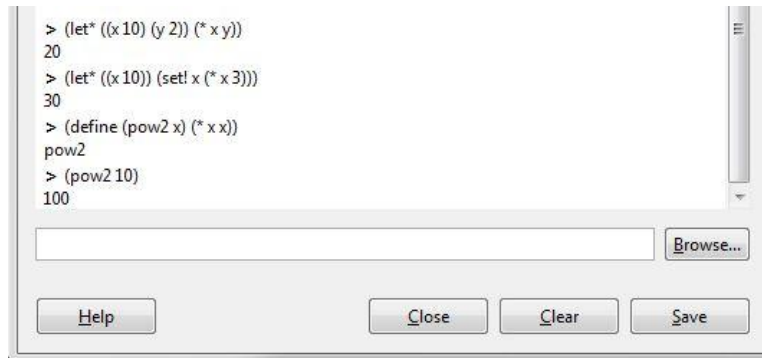
Scrierea unui script

Pentru a testa comenzile Script-Fu primul pas pe care trebuie să-l facem este să pornim consola limbajului Scheme. Aceasta se realizează folosind comanda „*Filters / Script-Fu / Console*”. Fiecare comandă sau operație matematică va fi scrisă între paranteze rotunde (). Primul termen după deschiderea parantezei va fi numele comenzii sau operatorul matematic folosit, urmat de un spațiu alb (obligatoriu). Imediat urmează seria operanzilor, separați de spații albe. Pentru scrierea operațiilor matematice de exemplu, comenzile vor arăta precum în Figura 28. Se poate observa că fiecare operator matematic poate accepta mai mult de 2 operanzi și de asemenea că fiecare operand poate la rândul lui să fie rezultat în urma unei operații matematice separate (scrisă între două paranteze interioare). Rezultatul fiecărei operații apare scris pe linia imediat următoare, după apăsarea tastei „*Enter*” (Figura 28).

Pentru definirea unei variabile se folosește comanda *let** a cărei utilizare este exemplificată în Figura 29, folosind consola. Doar atunci când folosim consola este nevoie să scriem toată paranteza pe un rând. Când vom realiza un script într-un fișier Script-Fu având terminația *.scm*, atunci putem desfășura conținutul unei paranteze pe mai multe rânduri. Pe prima linie din Figura 29 am inițializat așadar variabila *x* la valoarea 10 și variabila *y* la valoarea 2, folosind comanda *let**. În aceeași paranteză care cuprinde comanda *let**, am scris și operația de înmulțire (** x y*). Aceasta pentru că variabilele *x* și *y* nu au sens în afara parantezei care conține comanda *let**. Pentru a schimba valoarea unei variabile după ce aceasta a fost definită, putem folosi comanda *set!*, precum în a doua linie de comandă din Figura 29.



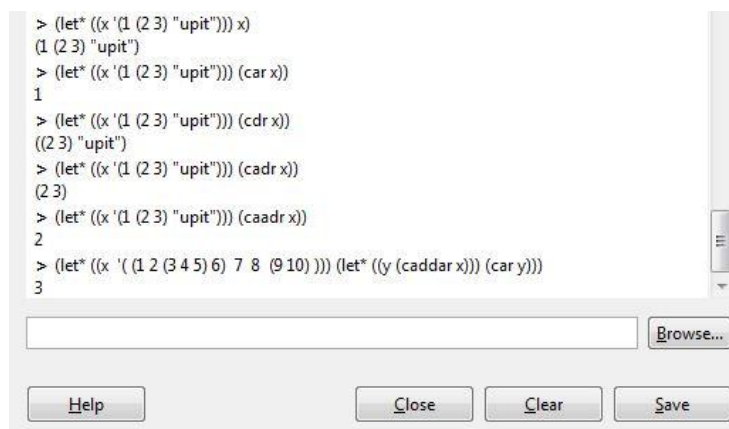
Figura 28 Folosirea operatorilor matematici în consola Script-Fu



```
> (let* ((x 10) (y 2)) (* x y))
20
> (let* ((x 10)) (set! x (* x 3)))
30
> (define (pow2 x) (* x x))
pow2
> (pow2 10)
100
```

The console window includes a text input field, a 'Browse...' button, and a row of buttons: 'Help', 'Close', 'Clear', and 'Save'.

Figura 29 Definirea variabilelor și a funcțiilor în consola Script-Fu



```
> (let* ((x '(1 (2 3) "upit")))) x)
(1 (2 3) "upit")
> (let* ((x '(1 (2 3) "upit"))) (car x))
1
> (let* ((x '(1 (2 3) "upit"))) (cdr x))
((2 3) "upit")
> (let* ((x '(1 (2 3) "upit"))) (cadr x))
(2 3)
> (let* ((x '(1 (2 3) "upit"))) (caadr x))
2
> (let* ((x '(1 2 (3 4 5) 6) 7 8 (9 10))) (let* ((y (caddr x))) (car y)))
3
```

The console window includes a text input field, a 'Browse...' button, and a row of buttons: 'Help', 'Close', 'Clear', and 'Save'.

Figura 30 Folosirea listelor în Script-Fu

Pentru definirea unei funcții se folosește comanda „*define*”, urmată de o paranteză care va conține numele funcției și numele tuturor parametrilor funcției. Uitându-ne la a treia linie din Figura 29, noua funcție poartă numele de „*pow2*” și primește un singur parametru „*x*”. Ultima operație înainte de închiderea parantezei funcției fixează valoarea returnată. Pentru cazul funcției *pow2*, aceasta va returna valoarea $(* x x)$, adică x^2 . Funcția poate fi testată apoi într-o comandă separată, precum în ultima linie de comandă din Figura 29.

Putem defini o variabilă (de exemplu *x*) nu doar sub forma unei valori unitare, ci și ca un vector (listă). Lista este o structură de date tipică limbajului Scheme și poate fi creată precum în prima linie din Figura 30. Se poate observa cum variabilei *x* i se alocă o listă definită ca o paranteză precedată de un apostrof. Observăm că al doilea element al listei este o altă listă. De data aceasta, fiind vorba de o listă interioară, nu este necesară folosirea apostrofului. Mai putem observa că listele pot conține variabile de tipuri diferite. Intr-adevăr, spre deosebire de limbaje precum C sau Java, în Scheme nu este nevoie să fim preocupați de tipurile de date. Pentru a accede la primul element al listei, s-a folosit în a doua linie comanda „*car*”. Pentru a accesa

restul listei se folosește comanda „*cdr*” (a treia linie din Figura 30). Există comenzi care înlănțuiesc cele două instrucțiuni, cum este exemplificat în ultimele linii de comandă din Figura 30. De exemplu, o comandă „*car*” aplicată valorii returnate de o comandă „*cdr*” se scrie prescurtat drept „*cadr*” precum în a patra linie de comandă din Figura 30.

Un script de animație

Putem genera o animație gif folosind un filtru creat de noi, asemănător celei pe care am realizat-o folosind filtrul nativ „*Filters / Animation / Waves*” (exercițiul 2.6.7). Vom face aceasta analizând codul filtrului nativ, schimbându-l acolo unde este necesar. Primul pas este așadar copierea filtrului „*waves-anim.scm*” într-un fișier „*whirl-anim.scm*”. Dacă studiem codul filtrului inițial putem observa că la un moment dat în cod se face apel la un plug-in numit „*plug-in-waves*”. Acest plug-in poate fi apelat și direct de utilizator folosind comanda „*Filter / Distorts | Waves*”. Filtrul de animație „*waves-anim.scm*” instanțiază așadar imaginea într-o serie de cadre identice, iar apoi va aplica fiecărui cadru în parte plugin-ul „*Waves*” folosind de fiecare dată alți parametri. Atunci când salvăm noua serie de cadre în format gif, schimbările produse fiecărui cadru se vor traduce în iluzia propagării unor valuri pe suprafața apei.

Noul filtru („*whirl-anim.scm*”) va face apel la o altă comandă, anume „*Filter | Distorts | Whirl and Pinch*”, pentru a genera efectul de animație. Este bine ca înainte de a începe să programăm noul filtru de animație să ne familiarizăm cu parametrii pe care-i poate primi această comandă („*Whirl and Pinch*”) și să-i comparăm cu parametrii pe care-i primește plugin-ul „*Waves*”. Observăm așadar că plugin-ul „*Filter | Distorts | Whirl and Pinch*” poate primi doar trei parametri: *Whirl angle*, *Pinch amount* și *Radius*. Filtrul de animație pe care-l vom scrie în această secțiune va varia doar primul dintre cei trei parametri (*Whirl angle*). Ceilalți doi parametri se vor păstra constanți în toate cadrele generate de filtrul de animație.

Primele modificări pe care trebuie să le aducem filtrului „*waves-anim*” pentru a-l adapta scopului nostru sunt prezentate în Figura 31. Ultima funcție (*script-fu-menu-register*) fixează plasamentul în meniu unde vom putea găsi filtrul nou creat. Funcția *script-fu-register* fixează ce parametri vor putea fi reglați în dialogul filtrului (fereastra care se deschide imediat după selectarea filtrului și care precede execuția sa).

Parametrii proprii noului filtru (*Whirl*, *Pinch* și *Radius*) sunt toți trei reglați folosind un control de tipul „*SF-ADJUSTMENT*” (Figura 31b). Acest tip de control primește întotdeauna ca prim parametru șirul de caractere care va fi afișat în fereastra de dialog în dreptul său, urmat de o listă ce va conține valoarea inițială a parametrului, valoarea minimă, valoarea maximă, pasul, pasul mărit (prin apăsarea butoanelor *pageUp/pageDown*), numărul de zecimale după virgulă și modul de lucru al controlului (0 sau 1). Numărul de cadre al animației noastre este reglat folosind tot un control de tipul „*SF-ADJUSTMENT*”, lucrând în modul de lucru restrâns (ultimul parametru este 1, în loc de 0).

Pinch” este „*whirl-pinch.exe*”. Primii trei parametri ai plugin-urilor sunt identici: modul de lucru, imaginea curentă și cadrul activ. Următorii parametri sunt specifici fiecărui filtru în parte.

Codul explicat al noului filtru de animație „*Whirl*” este afișat în întregime în anexa acestei lucrări de laborator. Dacă faceți o comparație între acesta și codul filtrului nativ „*Waves*” veți putea observa că noul filtru va instanția un număr dublu de cadre decât o face filtrul nativ. Numărul suplimentar de cadre este necesar pentru a asigura o tranziție fină între cadrul în care parametrul de răsucire (*whirl*) este maxim și cadrul inițial (în care răsucirea este nulă). Acest număr suplimentar de cadre nu era necesar pentru cazul filtrului „*Waves*”, al cărui parametru variabil era faza (parametru periodic).

Desfășurarea lucrării

1. Incercați să repetați cele două imagini obținute în Figura 8_{b,c}, folosindu-vă atât de unealta „Color Balance” cât și de unealta „Hue-Saturation”. Inregistrați rezultatele obținute în patru fișiere .jpg distincte.

2. Transformați în alb-negru imaginea din Figura 8a folosind comanda „Colors | Desaturate” (Figura 34a). Folosiți apoi unealta „Levels” pentru a transforma liniar spațiul luminozităților de intrare din intervalul 28-238 în intervalul de ieșire 0-255 (Figura 34b). Ce modificări apar pe histograme după această transformare? Pentru a accesa histogramele culorilor și intensității luminoase, este suficientă folosirea comenzii din meniu „Colors | Info | Histogram”, utilizată și în prima lucrare de laborator. Repetați din nou aceeași transformare însă folosind de data aceasta pentru parametrul „gamma” valoarea de 1,5 (Figura 34c). Studiați din nou histograma fiecărui canal în parte și observați diferențele. Reveniți la imaginea din Figura 34a. Aplicați din nou transformarea de la punctul precedent ((Figura 34c), însă selectând în locul canalului luminozităților (*Value*), canalul nivelurilor de roșu (*Red*). Imaginea care se va obține este cea din Figura 34d. Studiați din nou histogramele tuturor canalelor noii imagini.

3. Faceți o comparație între histogramele culorilor și intensității luminoase înainte și după aplicarea fiecărui reglaj automat discutat în Secțiunea 1.5 pe imaginea din Figura 8a.

4. Folosiți-vă de uneltele de culoare *Levels* („Colors | Levels”), *Desaturate* („Colors | Desaturate”) și *Invert* („Colors | Invert”) precum și de filtrele *Blur* („Filters | Blur | Gaussian Blur”) și *Edge* („Filters | Edge-Detect | Edge”) pentru a transforma imaginea din Figura 8a în imaginea din Figura 24a.

5. Folosiți-vă de filtrul de dilatație („Filters | Generic | Dilate”) și de uneltele de selecție pentru a transforma imaginea obținută la exercițiul precedent (Figura 24a) în imaginea din Figura 24b.

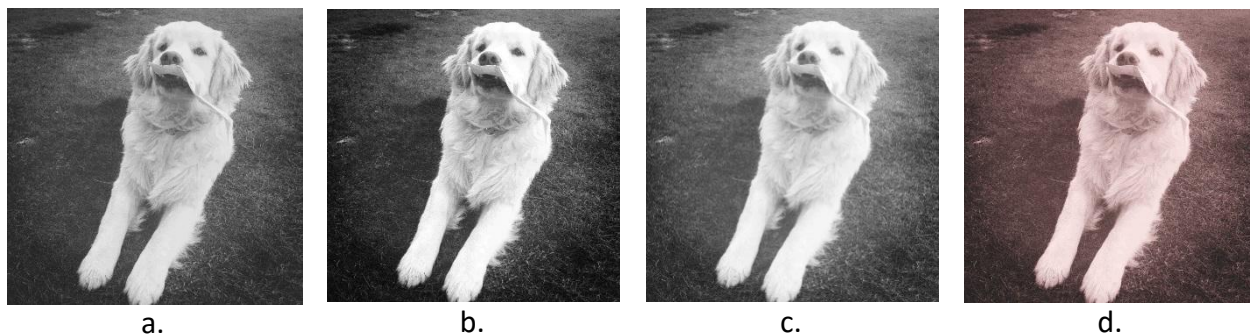


Figura 34 Folosirea unelei „Levels” pentru editarea imaginii din Figura 8a: a. imaginea originală desaturată; b. după aplicarea unelei Levels pe imaginea de la punctul „a” folosind o transformare liniară a intervalului 28-238 în intervalul 0-255; c. după aplicarea aceleiași transformări de la punctul „b”, însă folosind un gamma de 1,5; d. după aplicarea aceleiași transformări de la punctul „c”, însă pe canalul roșu

6. Folosiți matricea de convoluție din Figura 25 pentru a obține harta gradientului (atât pe orizontală cât și pe verticală) a imaginii din Figura 8a. Ce se întâmplă dacă înjumătățim divizorul în fereastra de dialog din Figura 25?

7. Creați un puzzle din 49 de bucăți pe baza imaginii din Figura 8a folosind comanda „Filters | Render | Pattern | Jigsaw”.

8. Realizați o hartă de imagine pentru poza din Figura 24a. Selectați 4-5 regiuni de interes din image și instruiți browserul ca în urma unui click pe una dintre aceste regiuni să răspundă cu o fereastră de dialog (*javascript: alert('text')*) în care să fie afișată denumirea regiunii respective. Rezultatul va fi integrat într-o pagină web și testat în browser.

9. Realizați meniul unei pagini web personalizate sub forma unei imagini originale pe care să o fragmentați în mai multe zone independente corespunzătoare etichetelor: *Home*, *History*, *Contact*, etc., care să-și schimbe transparența la trecerea mouse-ului peste ele.

10. Realizați o animație .gif a unei poze alese de dumneavoastră folosindu-vă de filtrul „Filters | Animation | Waves”.

11. Folosind instrucțiuni de tip *car*, *cdr* și înlănțuirile lor, accesați elementul „6” din ultima listă inițializată în Figura 30: ((1 2 (3 4 5) 6) 7 8 (9 10)).

12. Implementați scriptul de animație „*whirl-anim.scm*” explicat în această secțiune și aplicați-l pe o imagine aleasă de dumneavoastră pentru generarea unui fișier *gif* de cel puțin 20 de cadre. Animația trebuie să pornească de la un unghi minim de 0°, să ajungă la un unghi de răsucire de 600° și să se întoarcă treptat la unghiul inițial de 0° înainte de a se repeta.

13. Notați dimensiunea fișierului *gif* obținut la exercițiul precedent. Aplicați acum pe imaginea *gif* obținută anterior filtrul „Filters | Animation | Optimize (for GIF)” și salvați din nou

imaginea într-un nou fișier *gif*. Cum s-a modificat dimensiunea noii imagini? Cum vă explicați modificarea?

14. Realizați un filtru de animație bazat în continuare pe plugin-ul „*Whirl and Pinch*”, de data aceasta însă păstrând răsucirea constantă și variind doar strangularea (parametrul *pinch*). Aplicați filtrul astfel implementat pe o imagine aleasă de dumneavoastră. Animația trebuie să pornească de la o strangulare nulă și să ajungă la o strangulare maximă de 0,5 (*pinch=0.5*). Revenirea la cadrul inițial trebuie să se facă treptat. Animația astfel creată se va repeta în buclă.

Exerciții recapitulative

1. Deschideți în GIMP imaginea unui curcubeu.
 - a. Inversați în interiorul curcubeului poziția primei culori (roșu închis) cu poziția celei de a cincea culori (albastru deschis). Noul curcubeu va avea așadar prima culoare albastru deschis, iar a cincea roșu, restul culorilor rămânând pe poziții neschimbate.
 - b. Modificați reprezentarea ultimelor trei etaje inferioare ale curcubeului din culori în nuanțe de gri, păstrând cele 4 etaje superioare în culori.
 - c. Aflați câte nuanțe de culori are inițial imaginea. Indexați imaginea pentru a rămâne cu numai 4 culori și salvați-o într-un fișier *.png*, pe desktop.
2. Deschideți în GIMP o imagine cu portretul fotografic al unei persoane.
 - a. Inchideți culoarea ochilor personajului (de exemplu de la albastru spre negru).
 - b. Realizați conturul ochilor folosind trasee precum în figura de mai jos și salvați conturul în format vectorial (*.svg*) pe Desktop.
 - c. Salvați imaginea într-un fișier *.jpg* pe Desktop cu un factor de calitate de 90. Modificând apoi doar dimensiunea imaginii (numărul de pixeli) reduceți spațiul ocupat de imagine pe disc de aproximativ 16 ori. Factorul de calitate al fișierului *.jpg* trebuie să rămână neschimbat la 90.

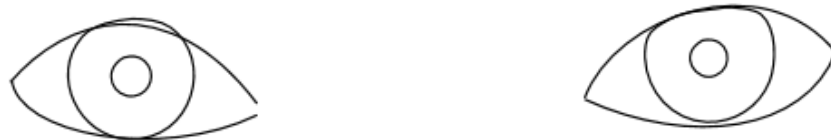


Figura 17. Realizarea conturului ochilor folosind trasee

3. Deschideți în GIMP o imagine oarecare.
 - a. Reprezentați imaginea pe 2 straturi: pe primul dintre acestea (cu transparența de 50%) imaginea trebuie să fie reprezentată doar în niveluri de roșu (canalul roșu) iar pe stratul de jos (cu opacitatea de 100%) imaginea trebuie să fie reprezentată doar în niveluri de verde (canalul verde).
 - b. Realizați o animație .gif de 6 cadre în care stratul de sus (canalul roșu) trebuie să se deplaseze cu viteză constantă spre stânga față de stratul de jos (canalul verde) până la o deplasare maximă de 30 pixeli.
 - c. Reveniți la imaginea inițială. Reduceți numărul de pixeli prezenți pe imagine de 4 ori fără să tăiați din imagine și salvați rezultatul într-un fișier .jpg pe Desktop.

4. Realizați în GIMP o imagine-colaj care să conțină un șir de obiecte (cel puțin 6) nesuprapuse, luate din diferite imagini de pe web, toate așezate pe un fundal alb. Pentru a ușura decuparea lor, recomand ca și imaginile inițiale să conțină obiectele de decupat tot pe un fundal alb. O căutare pe google a obiectului dorit urmat de sintagma „on white background” va rezolva ușor problema.
 - a. Schimbați poziția primului obiect (de la stânga la dreapta) cu cel de al treilea.
 - b. Reprezentați al cincilea obiect doar în nuanțe de gri (restul obiectelor păstrând reprezentarea în culori).
 - c. Realizați maparea imaginii (harta web a imaginii) în așa fel încât fiecare obiect să trimită către o pagină web distinctă (de exemplu primul obiect către “1.html”, al patrulea către “4.html”, etc.). Incercați ca hyper-linkurile să urmeze cât mai fidel conturul obiectelor din imagine. Testați codul html generat de GIMP într-un browser.

5. Deschideți în GIMP o imagine oarecare.
 - a. Copiați imaginea în două straturi inițial identice. Stratul de jos, cu opacitate de 100%, va rămâne neschimbat. Modificați transparența stratului de sus la 50% și înjumătățiți valoarea de albastru a fiecărui pixel din acest strat (de exemplu dacă un pixel are codul RGB 0x121488, atunci valoarea de albastru a acestui pixel este de 88, iar după transformare va ajunge la 44).
 - b. Realizați o animație .gif de cel puțin 6 cadre în care trecerea de la imaginea inițială la imaginea obținută la punctul precedent (a) să fie făcută treptat.
 - c. Câte culori are inițial imaginea? Indexați imaginea pentru a rămâne cu numai 10 culori și salvați-o într-un fișier .png, pe desktop.

6. Deschideți în GIMP fotografia unei flori.

- a. Selectați floarea. Stergeți fundalul și înlocuiți-l cu un fond alb. Reprezentați floarea în nuanțe de gri, precum în Figura 3 (a, b).
- b. Alegeți 3 petale cat mai distantate și scrieți trei cuvinte diferite, de-a lungul fiecărei petale (precum în figura de mai jos).
- c. Realizați maparea imaginii (harta web a imaginii) în așa fel încât fiecare din cele trei petale marcate să trimită către o pagină web distinctă (de exemplu prima petală către “fraternitate.html”, a doua către “libertate.html”, etc.). Incercați ca hyper-linkurile să urmeze cât mai fidel forma celor trei petale. Testați codul html generat de GIMP într-un browser.
- d. Realizați o mini-animație .gif de 6 cadre în care floarea să facă o rotație completă.

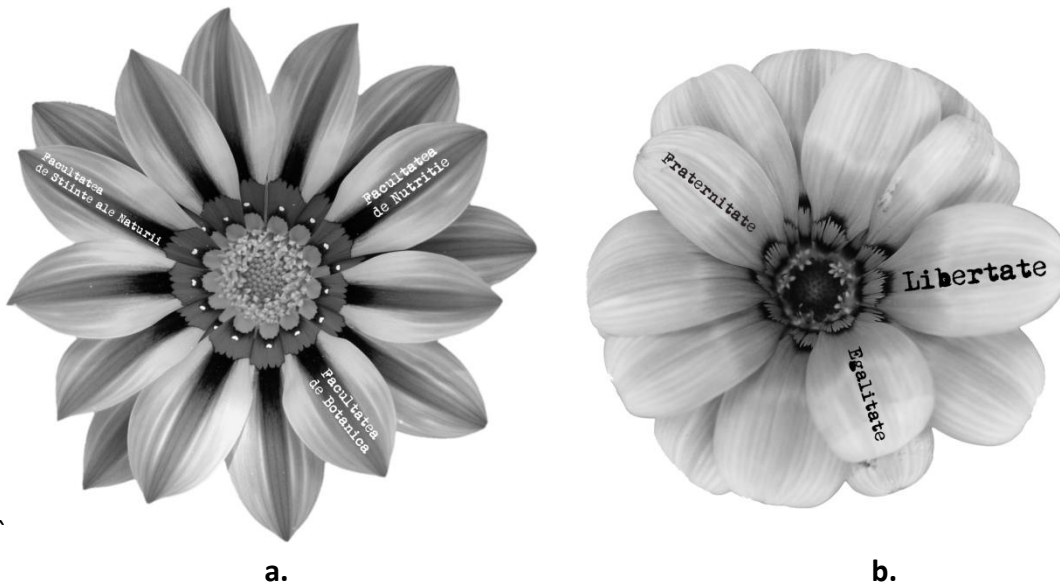


Figura 18. Exemplu de fotografii de flori, la care fundalul original a fost înlocuit cu un fond alb, iar trei petale au fost folosite ca suport pentru obiecte text.

7. Căutați pe web imaginea unor baloane colorate situate pe un fond uniform (precum cerul) și deschideți imaginea în GIMP.
 - a. Câte culori are inițial imaginea? Indexați imaginea pentru a rămâne cu numai 10 culori și salvați-o într-un fișier .png, pe Desktop.
 - b. Detectați conturul baloanelor din imagine și salvați imaginea într-un fișier .jpg, pe desktop. Precum în imaginea de mai jos, conturul baloanelor trebuie marcat cu negru pe un fond alb.
 - c. Realizați o animație .gif de minim 6 cadre în care baloanele să se ridice până ies în întregime din ecran, folosindu-vă de fișierul .jpg creat anterior.

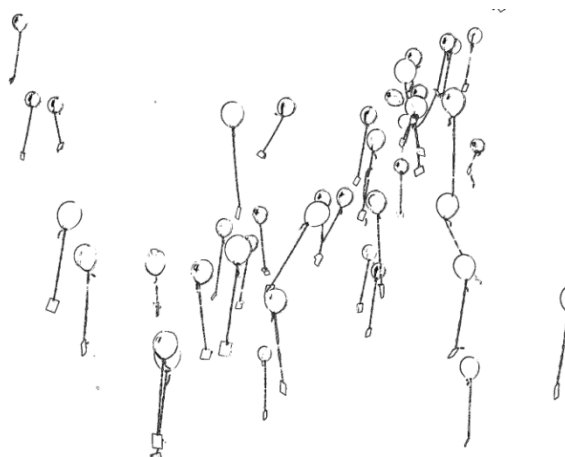


Figura 19. Exemplul unei fotografii la care obiectele din prim plan (baloanele) au fost detasate de fundal.

Capitolul 4. Securizarea serverelor web. Certificate self-signed. Virtual Private Network (VPN).

Obiectivul lucrării

Lucrarea își propune studierea modului în care se realizează accesarea conținutului multimedia și a câtorva metode pentru conectarea sigură la aceste resurse. Aplicațiile investighează instalarea și configurarea unui server web folosind mecanisme de securizare a traficului clienților, a unui server web folosind mecanisme de securizare bazate pe certificate auto-semnate (self-signed) cât și conectarea securizată la resursele unei rețele prin intermediul VPN.

Breviar teoretic

Servere web

HTTP- Hypertext Transfer Protocol este un protocol de comunicație de nivel aplicație (cel mai apropiat de utilizator) prin care sunt accesate informațiile hypertext din World Wide Web (de obicei numit www sau web). Documentele hypertext permit organizarea ușoară a conținutului oferit utilizatorilor. Informația www este transferată către utilizatori după modelul server – client. În acest model:

-clienții sunt reprezentați de aplicațiile pe care un utilizator le deschide (de exemplu un browser). Clientul are nevoie de informațiile de pe server și în acest sens trimite cereri către acesta. Exemple de metode pe care un client le poate trimite către un server HTTP:

- GET: este cea mai folosită metodă, pentru a cere serverului o resursă.
- PUT: metoda este folosită pentru a pune documente pe server, atunci când se știe exact locația resursei care se dorește a fi actualizată. La fiecare apelare funcția are același efect, adică este idempotentă;
- POST: metoda este folosită pentru a trimite date către server, atunci când de serverul este lăsat să decidă unde va plasa datele respective. PUT și POST implementează funcționalitatea similară, însă POST nu este idempotentă;
- HEAD: este similară metodei GET, dar serverul returnează doar antetul resursei, ceea ce permite clientului să inspecteze antetul resursei, fără a fi nevoit să obțină și corpul resursei.
- DELETE: șterge resurse de pe un server.

- TRACE: este o metodă folosită de obicei pentru diagnosticare, putând da mai multe informații despre traseul urmat de legătura HTTP, fiecare server proxy adăugându-și semnătura în antetul Via.
- OPTIONS: este folosită pentru identificarea capacităților serverului Web, înainte de a face o cerere.

-serverul este o aplicație care rulează pe un calculator și poate răspunde cererilor clienților. Datele solicitate de clienți sunt pe server sau acestea le poate obține pentru clienți în urma primirii unei cereri de la aceștia. Serverele HTTP ascultă în mod implicit cererile clienților pe portul 80.

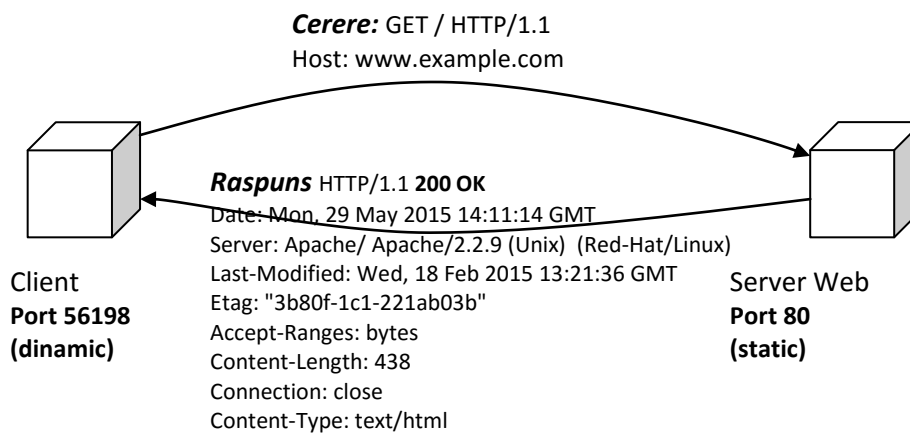


Figura 1 Un exemplu de cerere și de raspuns HTTP. Se observa portul static (80) al serverului web și cel dinamic ales de client pentru acea conexiune cât și codul de raspuns al serverului trimis catre client .

Dacă se solicită o resursă, serverul poate returna următoarele coduri:

- 1xx - erori informaționale: această clasă de status indică un răspuns provizoriu al serverului care conține numai linia de status și alte antete opționale. Nu sunt antete obligatorii pentru această clasă de răspuns/status. Aceste status-uri pot fi ignorate.
- 2xx - răspuns corect: aceasta clasă de răspuns/status indică utilizatorului că cererea a fost primită, înțeleasă și acceptată. Cel mai cunoscut cod este 200 – ok care indică faptul că cererea clientului a fost executată cu succes.
- 3xx - redirectări: această clasă de răspuns/status indică faptul că acțiuni suplimentare vor trebui luate de browserul clientului pentru a putea fi îndeplinită cererea. Redirectarea ar putea fi făcută de browser automat (fără a interacționa cu utilizatorul) dacă metoda folosită în cea de a doua cerere este GET sau HEAD, altfel se cere intervenția utilizatorului. Cel mai cunoscut cod este 301 - mutat permanent: resursa solicitată are un nou URI permanent iar cererile următoare ar trebui să folosească una din sursele returnate în câmpul Location al răspunsului.

- 4xx - erori ale clienților: această clasă de mesaje/statusuri este folosită în cazurile în care clientul a greșit formularea cererii. De obicei serverul ar trebui să returneze o explicație a erorii și dacă e o eroare temporară sau permanentă. Cele mai cunoscute coduri sunt:
 - 400 - cerere greșită: Cererea nu a putut fi înțeleasă de către server din cauza unei sintaxe greșite/incomplete.
 - 401 – neautorizat. Cererea are nevoie de un câmp (Authenticate) prin care să poată identifica clientul înainte de a îi transmite date. Transmiterea de nume și parola prin folosirea acestui câmp face nesigură autentificarea prin protocolul HTTP (fără criptare) deoarece orice interceptare a traficului permite vizualizarea acestor informații.
 - 403 - interzis: Serverul a înțeles cererea, dar refuză să o îndeplinească.
 - 404 - negăsit: Serverul nu a găsit resursa care să corespundă cererii URI. Nu se dau indicații dacă condiția temporară sau permanentă.
- 5xx - erori de server: aceste coduri de răspuns indică faptul că serverul recunoaște că a greșit sau este incapabil să execute cererea. Răspunsul ar trebui să includă o explicație a situației de eroare, fie că e temporară sau permanentă. În funcție de setările de securitate ale serverului este posibil ca mesajul de eroare să nu fie trimis clientului, în acest caz afișând un mesaj că nu poate afișa mesajul de eroare. Cele mai cunoscute coduri sunt:
 - 500 - eroare internă de server: Server-ul a întâmpinat o condiție neașteptată și nu va putea îndeplini cererea.
 - 503 Service Unavailable – indică faptul că serverul nu poate procesa cererea din cauza unei supraincercări sau a unor lucrări de mentenanță efectuate pe acesta.
 - 505 - versiunea HTTP nu e suportată/susținută: Serverul nu suportă versiunea de protocol a HTTP ce a fost folosită în formularea cererii.

Deoarece HTTPS nu oferă mecanisme pentru securizarea accesului clienților, s-a propus înlocuirea acestuia cu HTTPS.

HTTP Secure (HTTPS) – este un protocol de comunicare securizat de nivel aplicație prin care sunt accesate informațiile hypertext din www. Securizarea se face prin Transport Layer Security sau Secure Sockets Layer, protocoale de nivel aplicație bazate pe TCP. Toate cererile HTTP sunt suportate, singura diferență fiind că ele circulă prin canalul securizat creat de TLS/SSL.

HTTPS ascultă cererile clienților pe portul 443 și accesarea în browser se face cu https:// spre deosebire de HTTP care preceda numele site-ului cu http://.

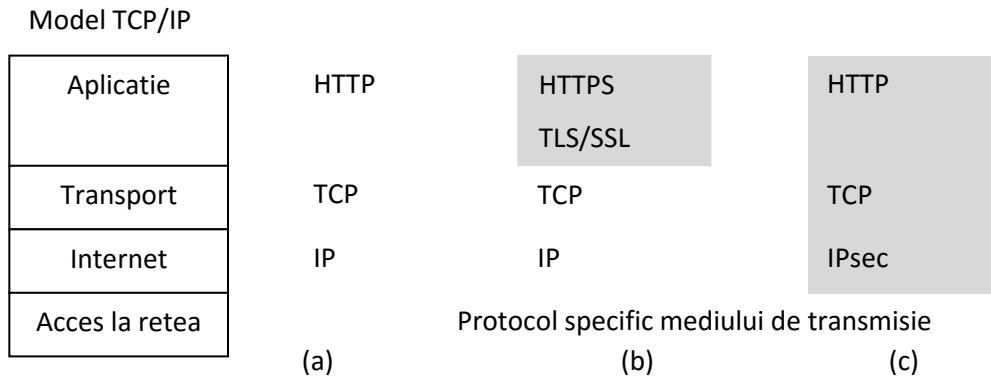


Figura 2 Compararea metodelor pentru accesarea nesecurizata a continului unui site (a), respectiv securizata folosind TLS/SSL (b) și VPN (c)

Spre deosebire de HTTP, HTTPS solicita aplicatiei browser sa foloseasca un nivel de criptare pentru a proteja schimbul de informatii intre client și server. Se creaza in acest mod un canal de comunicatie securizat intre client și server chiar dacă rețeaua prin care sunt schimbate datele nu ofera acest lucru. Suplimentar serverul va oferi clientului un certificat (pe care acesta il cunoaste deja sau il poate verifica cu o terta parte de incredere) in scopul autentificarii identitatii sale pentru a preveni atacurile in retea.

Singurele informatii expuse de HTTPS in mod necriptat sunt cele legate de protocoalele inferioare:

- adresa IP, necesara protocolului IP pentru a identifica destinatia traficului la nivelul internetului;
- numarul de port necesar protocoalelor TCP sau UDP pentru a sti carei aplicatii de la adresa IP anterioara ii este destinat traficul HTTPS.

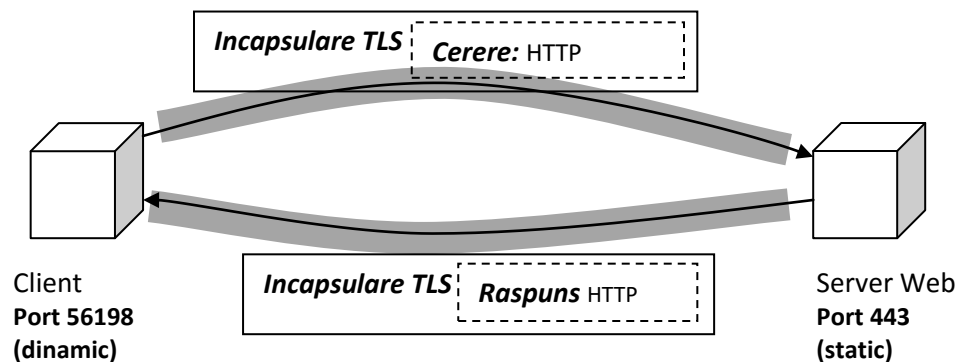


Figura 3 Un exemplu de cerere și de raspuns HTTPS. Se observa portul static (443) al serverului web și incapsularea cererilor HTTP de catre protocolul TLS pentru a folosi canalul securizat deschis de acesta .

Rolul certificatelor și al certificatelor auto-semnate

Pentru a putea să ne conectăm la o rețea privată folosind conexiuni realizate printr-un mediu de transmisie nesigur, este necesar să un sistem capabil să securizeze comunicația și să ne ofere mecanismele pentru verificarea identitatii participanților la un schimb de mesaje. Acest suport este oferit de către Public Key Infrastructure (PKI) și are următoarele componente:

- o autoritate de certificare (CA) care eliberează și verifică certificatele digitale;
- autoritate de înregistrare (RA) care verifică identitatea utilizatorilor care solicită informații de la autoritatea de certificare
- o locație sigură de stocare care se păstrează cheile private și certificatele
- o baza de date care primește cererile pentru certificare și returnează certificatele.

O autoritate de certificare (Certificate Authority - CA) este o entitate care emite certificate digitale. Un certificat digital dovedește dreptul de proprietate asupra unei chei publice al celui pe numele căruia a fost eliberat certificatul.

Există numeroase autorități care eliberează certificate (Comodo, GoDaddy, Symantec) pentru a își verifica identitatea. Certificatele sunt semnate folosind cheia privată. Cheia publică este pusă la dispoziția celor interesați drept un certificat auto-semnat deoarece nu există o autoritate centrală care la rândul ei să autentifice aceste autorități.

Browseerle sunt setate sa genereze mesaje de avertizare în cazul în care întâlnesc certificate auto-semnate, în afara certificatelor care sunt deja incorporate în browser. Aceste certificate auto-semnate eliberate de CA nu generează erori și sunt considerate root certificate. Toate certificatele trebuie sa fie autorizate de o autoritate aparținând unui root certificate, altfel vor genera o eroare.

Având la baza certificatul se pot crea semnături care să certifice autenticitatea documentelor. Autoritatea de certificare este în acest caz componentă care autentifica faptul că documentul citit de o terța parte are o semnătura corectă și previne atacurile de tipul *man in the middle* în care un atacator se interpune între sursă și destinație în încercarea de a falsifica traficul de date.

Un certificat digital conține o cheie publică și identitatea proprietarului. Cheia privată nu este disponibilă public și este folosită de către proprietarul certificatului pentru a semna documentele. Utilizatorii pot verifica folosind cheia publică identitatea documentelor semnate cu cheia privată.

Un certificat auto-semnat este un certificat de identitate care este semnat de către aceeași entitate a cărei identitate o certifică. Acest termen nu are nimic de-a face cu identitatea persoanei sau organizației care a efectuat de fapt procedura de semnare. Din punct de vedere tehnic un certificat auto-semnat este unul semnat cu propria cheie privată .

VPN (Virtual Private Network)

Uneori este necesara conectarea la resursele unei rețele private dintr-o locație care nu este sigura (de exemplu o conexiune publica WiFi). Deoarece nu dorim sa expunem acest trafic privat în mod nesecurizat prin rețeaua publica, se poate folosi tehnologia VPN care permite crearea unui canal de comunicație securizat între doua dispozitive folosind o rețea nesigura pentru

transmiterea pachetelor de date. Acesta creează un adevărat tunel de la locația publică de la care s-a realizat conexiunea către resursele securizate ale rețelei locale. Accesarea resurselor private de la distanță dar în mod securizat duce la reducerea costurilor.

VPN asigură confidențialitatea datelor prin criptarea acestora; autentificarea sursei mesajelor pentru a preveni primirea de mesaje de la surse necunoscute și verificarea integrității mesajelor prin calcularea unui hash (cum ar fi hmac-md5 sau hmac-sha1) pentru a preveni modificarea rău intenționată a acestora.

Serverul care permite accesul la rețeaua VPN trebuie să fie pornit și conectat atât la internet cât și la rețeaua privată ale cărei resurse dorim să le accesăm. Modul în care accesăm rețeaua privată poate să nu funcționeze corect dacă accesarea se face prin VPN deoarece aceasta este o tehnologie point to point care în mod normal nu permite extinderea domeniului de broadcast. Pentru a depăși acest dezavantaj, se vor folosi tehnologii de nivelul 2 care permit tunelarea și a acestui tip de mesaje.

Protocoale folosite de VPN sunt:

PPTP (Point to Point tunneling protocol)

Acesta este protocol de suport pentru tehnologia VPN. Utilizatorii se pot conecta de la distanță prin acest protocol pentru a accesa rețeaua privată. Unii specialiști consideră că acest protocol este depășit din punct de vedere al securității și recomandă folosirea IPsec.

L2TP/IPsec (Layer 2 Tunneling Protocol)

Acest protocol este folosit împreună cu IPsec pentru a asigura criptarea, autentificarea și verificarea integrității mesajelor. Mesajele sunt trimise nemodificate din rețeaua sursă către rețeaua destinație.

IPSec (IP Security)

Este o platformă operând la nivelul Internet care are ca scop securizarea datelor care traversează nu numai nesigur, cum este internetul. IPsec asigură confidențialitatea, integritatea, autentificarea și protecția împotriva atacurilor prin repetarea pachetelor.

SSL VPN (Secure Socket Layer VPN)

Aceasta este un protocol folosit în special pentru stabilirea unei conexiuni criptate între serverul web și browser. Comparat cu IPsec, SSL este orientat pentru conectarea utilizatorilor la aplicații sau servicii în vreme ce IPsec este focalizat pe conectarea între două rețele diferite. Dezavantajul în utilizarea IPsec este că necesită instalarea în prealabil a unui software suplimentar, lucru care se poate dovedi dificil pentru unii clienți. Ca un avantaj însă, IPsec poate suporta toate aplicațiile bazate pe IP. Comparativ, SSL oferă implementări ale interfeței cu utilizatorul care diferă de la caz la caz însă utilizatorii pot accesa resursele fără a avea nevoie de un proces complicat de instalare și configurare a clientului de acasă.

SSL este recomandat pentru permiterea accesului la resursele rețelei în mod punctual (de exemplu doar serverul web), de la un client la altul, în vreme de IPsec permite accesarea tuturor resurselor unei rețele în grup și se potrivește pentru cazurile în care nivelul de încredere în rețeaua de la celalalt capăt al comunicației este mai mare.

Aplicații

Setarea unei conexiuni VPN

Setarea unei conexiuni VPN are doua componente: utilizatorul plasat într-o rețea nesecurizata care apelează un server din rețeaua locală pentru a deschide un tunel către rețeaua interna; serverul din rețeaua privata care primește cererile pentru stabilirea conexiunilor securizate VPN.

Sistemul de operare Windows are suport pentru stabilirea de conexiuni VPN începând cu sistemul de operare Windows XP. Exemplele sunt prezentate pentru Windows 10 însa cu modificări minimale pot fi adaptate pentru versiuni mai noi sau mai vechi ale acestui sistem de operare.

Setare server VPN

Se deschide din *Control panel* (Win+X și selectează Control Panel) meniul de *Network and Sharing*, și se deschide setarea *Change Adapter Settings*.

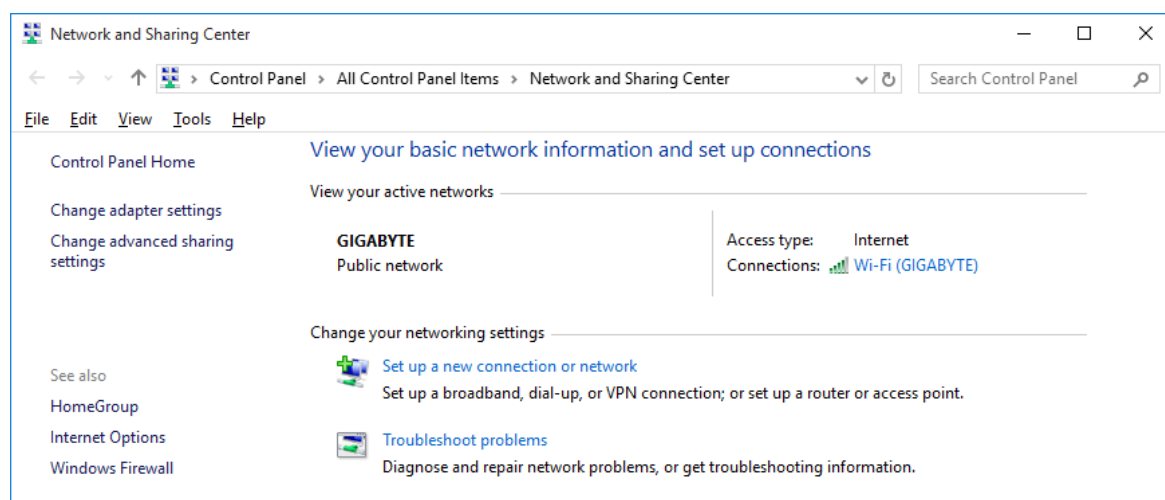


Figura 4 Network and Sharing Center – Windows 10

În fereastra nou apăruta, se selectează din meniul *File* (daca acest meniu nu este vizibil se selectează *Meniu bar* din meniul *Organize*) opțiunea *New Incoming connection*. În fereastra nou apăruta se selectează utilizatorii care vor beneficia de aceasta facilitare. Atenție aveți nevoie de un nume de utilizator și de o parola pentru conectare.

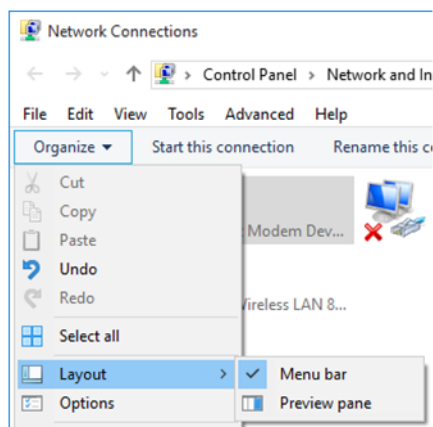


Figura 35 Pentru ca meniul "File" sa devina vizibil este necesara activarea opțiunii "Menu bar" din Layout.

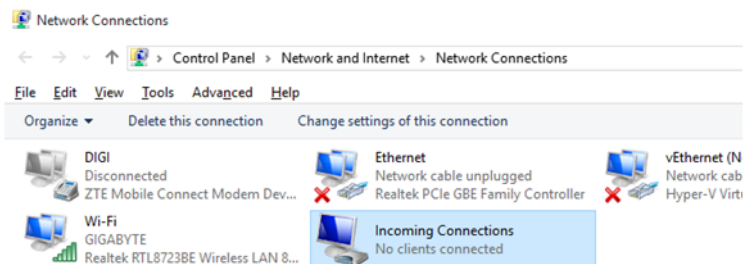


Figura 6 In „Network Connections” este selectata conexiunea VPN

În continuare se va selecta *Through Internet* pentru a indica faptul ca utilizatorii se vor conecta prin Internet și versiunea de protocol utilizata. În momentul de fata Internet Protocol Version 4 este încă cea mai utilizata conexiune și va fi selectată (pe viitor este posibila conectarea prin IP versiunea 6). În final se va selecta *Allow access* pentru a finaliza dialogul de setare a conexiunii VPN. Vă va fi afișată în acest moment informația necesară pentru a accesa calculatorul la distanță.

Serverul VPN este în acest moment configurat și în secțiunea *Network connections* va deveni vizibila o noua conexiune care arata dacă sunt sau nu clienți conectați prin VPN.

Setare client VPN

În secțiunea de Network și Internet din *Control Panel* se selectează opțiunea VPN (poate fi căutată din meniul de start direct) și se adaugă o noua conexiune VPN.

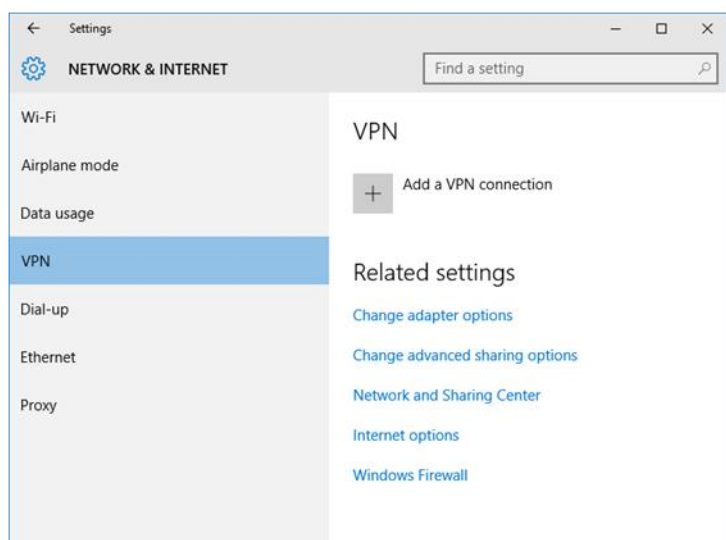


Figura 7 Fereastra „Network and Internet” din sistemul de operare Windows 10 care permite crearea unei noi conexiuni VPN

În fereastra nou apăruta se completează datele necesare pentru stabilirea conexiunii (IP, nume conexiune, etc.). Numele de utilizator și parola sunt opționale în acest pas deoarece dacă nu sunt introduse acum, vor trebui introduse la momentul stabilirii conexiunii. Mai multe tipuri de conexiune VPN sunt suportate:

- Point to Point Tunneling Protocol (PPTP);

- Layer 2 Tunneling Protocol with IPsec (L2TP/IPsec);

- Secure Socket Tunneling Protocol (SSTP) – este o forma de VPN care permite transmiterea de trafic printr-un canal SSL 3.0. SSL asigura negocierea cheilor, criptarea și verificarea integrității traficului. Deoarece este folosit portul 443 (utilizat și pentru site-urile web HTTPS) face ca acest VPN sa poată trece prin majoritatea sistemelor firewall.

- Internet Key Exchange v2 (IKEv2) – este o componenta a IPsec folosita pentru realizarea autentificării mutuale intre două părți.

Dacă nu se cunosc detalii despre modul în care a fost setat serverul se selectează opțiunea Automat care ar trebui să aleagă variată corectă pentru conexiunea realizată. În final după ce setările au fost realizate, se selectează conexiunea configurată și se apăsă *Connect*. Conexiunea VPN este stabilită în acest moment și aveți acces la serverele din rețeaua privată/locală. Puteți testa conectivitatea rapid prin ping.

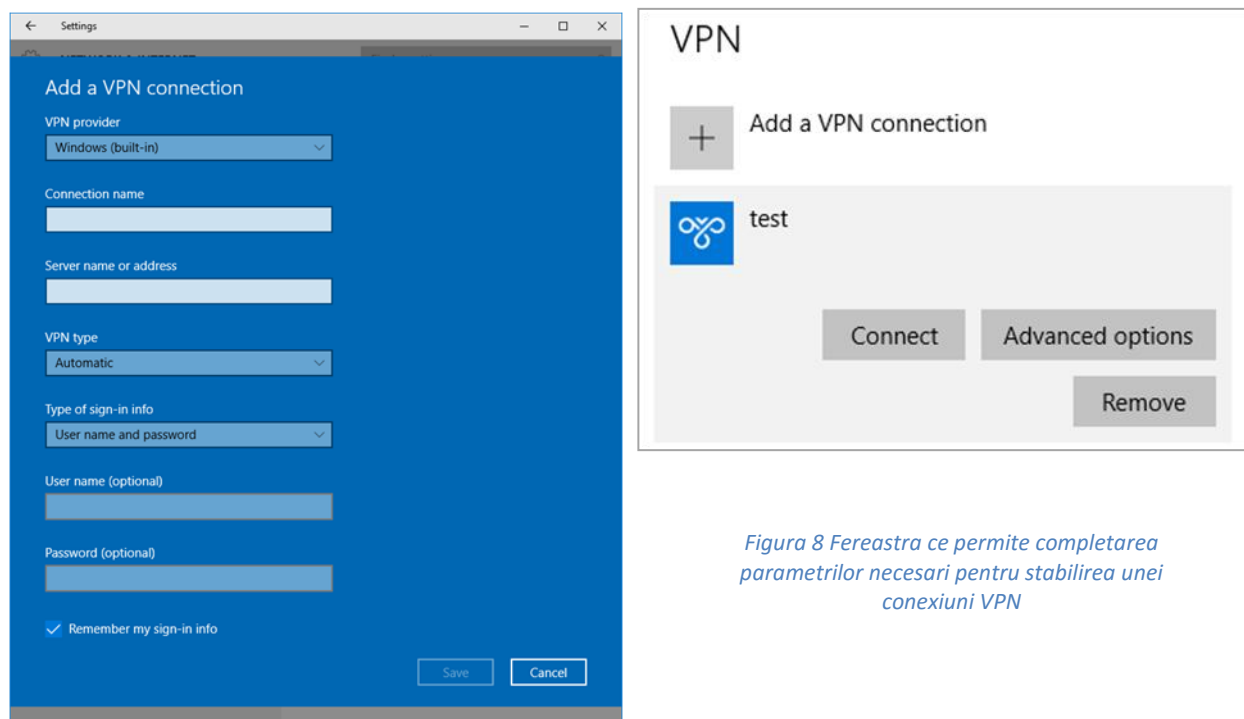


Figura 8 Fereastra ce permite completarea parametrilor necesari pentru stabilirea unei conexiuni VPN

În cazul în care conexiunea nu reușește:

- verificati dacă numele de utilizator și parola sunt setate și sunt corecte;

-verificati dacă nu este setat un Firewall pe mașina destinație care nu permite aceste conexiuni.

Instalarea unui server web Apache

XAMPP este un pachet complet care conține atât un server web (Apache) cât și un server de fișiere, SQL, mail, etc.

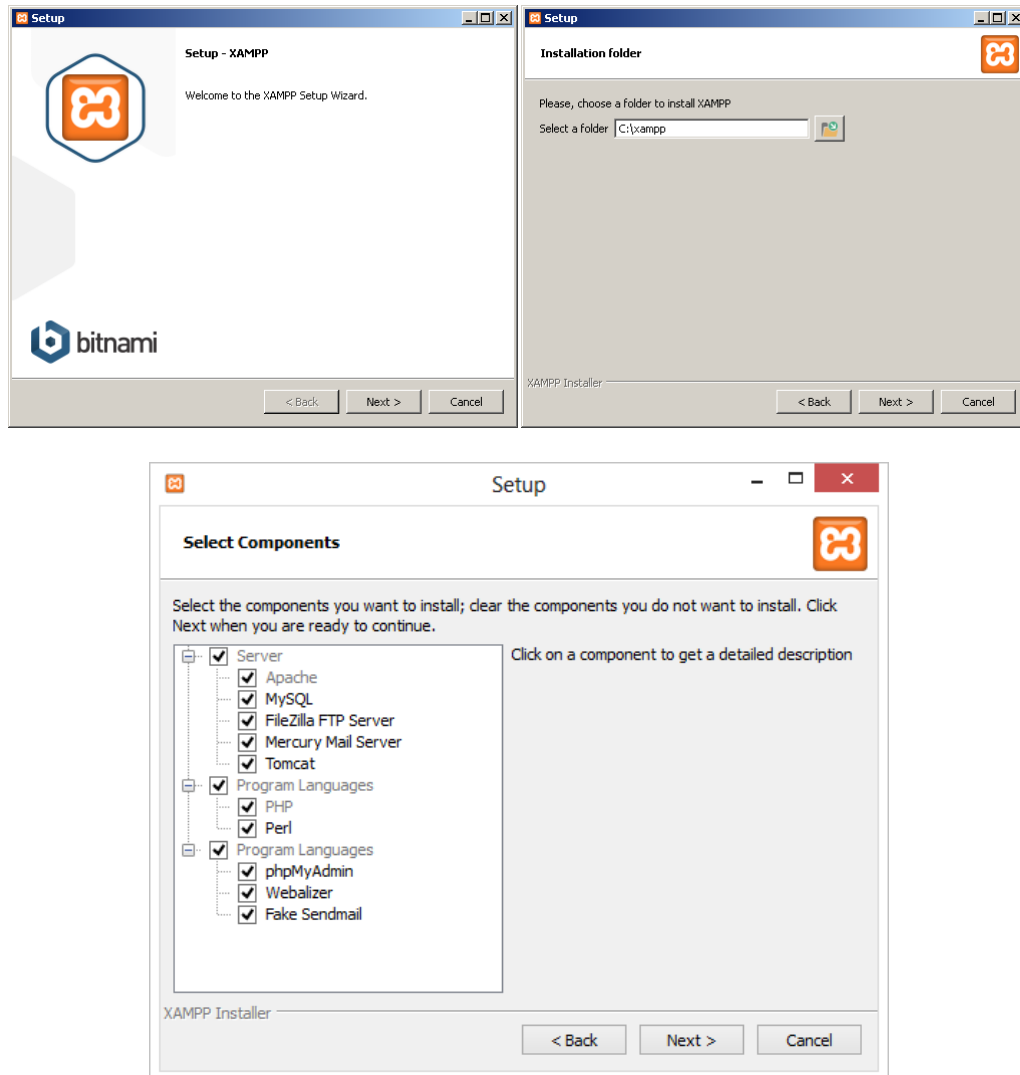


Figura 9 Interfața de instalare a aplicației XAMPP

Observație: Pentru Windows XP se recomandă folosirea serverului web din pachetul XAMPP 1.8 și mașina virtuală VirtualBox 4.3.12 din motive de compatibilitate. Pentru versiunile mai de Windows se recomandă folosirea ultimei versiuni a serverului.

Pentru o simplă instalare a unui server web (necurizat, HTTP), este suficientă instalarea XAMPP și pornirea serviciului web (Apache) din panoul de control XAMPP prin apăsarea butonului de Start.

Documentele care pot fi accesate de pe internet in mod implicit (de exemplu index.html) sunt cele plasate in calea:

`c:\xampp\htdocs`

Generarea și utilizarea unui certificat cu semnătură proprie în Apache

Observații:

- Deși este necesară instalarea serverului web Apache Nu este necesară pornirea serverului XAMPP înainte de configurarea cheii self-signed.
- *Comenzile din acest ghid presupun ca s-a instalat instalarea XAMPP s-a realizat în folderul c:\xampp. Comenzile se vor executa în consola Windows (Win+R și se tastează cmd după care se apasă Enter).*

Rolul folosirii unui certificat self-signed

Folosirea unui certificat este utilă pentru criptarea traficului web între sursa și destinație. La navigarea pe o pagină care folosește un certificat auto semnat, browserele vor afișa o atenționare privind încrederea în autoritatea care a eliberat acest certificat deoarece el este detectat drept un root certificate însă browserul nu îl are instalat în lista de certificate:

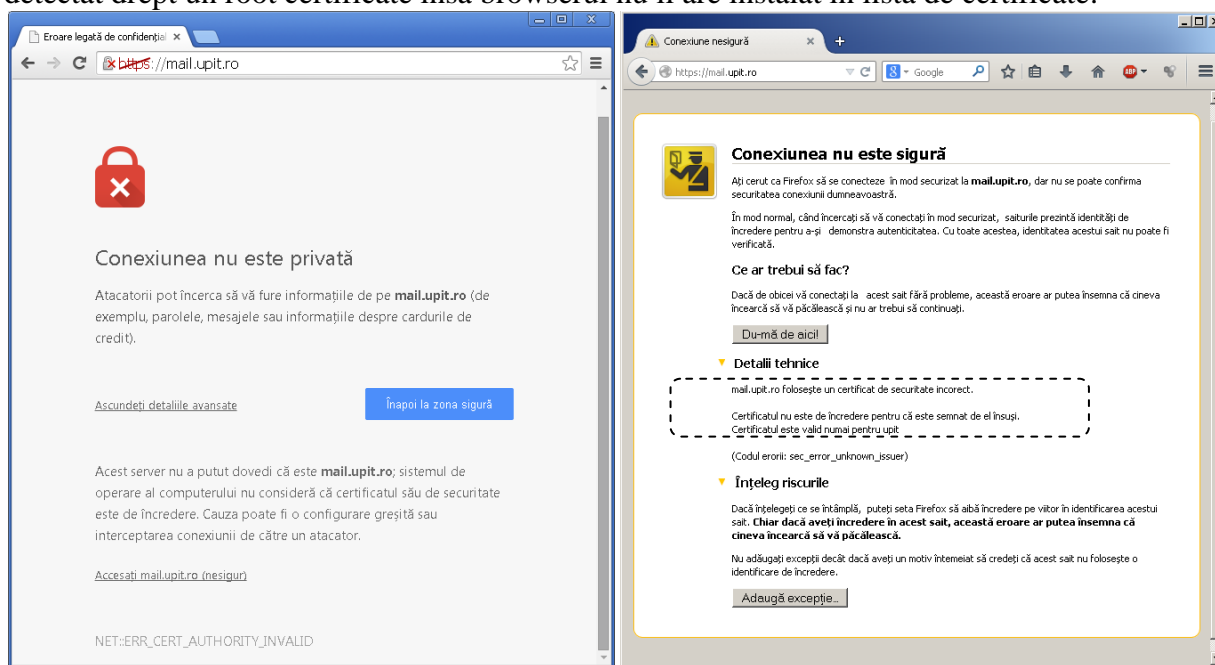


Figura 10 Atenționarea browserelor la detectarea unui certificat auto semnat

Configurarea serverului Apache în Windows pentru a folosi un certificat self-signed.

1. Generarea unei chei private

Se navighează în consola (Win+R și se tastează cmd după care se apasă Enter) în directorul în care s-a instalat serverul XAMPP:

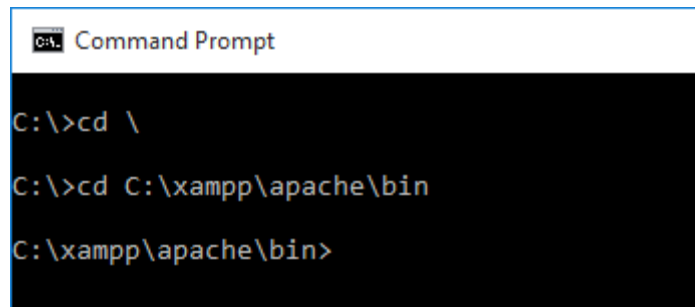


Figura 11 Navigarea în consola către directorul XAMPP

Se introduce în continuare:

```
C:\xampp\apache\bin>openssl genrsa -des3 -out server.key 1024
```

Loading 'screen' into random state – done

Generating RSA private key, 1024 bit long modulus

.....++++++

.....++++++

e is 65537 (0x10001)

Enter pass phrase for server.key: xxxxxxxx

Verifying – Enter pass phrase for server.key: xxxxxxxx

2. Generarea unui CSR (Certificate Signing Request)

```
C:\xampp\apache\bin>openssl req -new -key server.key -config
```

```
"C:\xampp\php\extras\openssl\openssl.cnf" -out server.csr
```

WARNING: can't open config file: c:/openssl-1.0.1i-win32/ssl/openssl.cnf

Enter pass phrase for server.key:

Loading 'screen' into random state - done

You are about to be asked to enter information that will be incorporated into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

Country Name (2 letter code) [AU]:RO

State or Province Name (full name) [Some-State]:Arges

Locality Name (eg, city) []:Pitesti

Organization Name (eg, company) [Internet Widgits Pty Ltd]:UPIT

Organizational Unit Name (eg, section) []:FECC

Common Name (eg, YOUR name) []:nomane

Email Address []:me@m.com

Please enter the following 'extra' attributes

to be sent with your certificate request

A challenge password []:test

An optional company name []:Nume Optional

Se va reține parola utilizată în acest pas deoarece este necesară în pasul următor.

3. Ștergere parola din cheie

```
C:\xampp\apache\bin>copy server.key server.key.org
```

1 file(s) copied.

```
C:\xampp\apache\bin>openssl rsa -in server.key.org -out server.key
```

Enter pass phrase for server.key.org:

writing RSA key

4. Generare Self-Signed Certificate

```
C:\xampp\apache\bin>openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt
```

Loading 'screen' into random state – done

Signature ok

subject=/C=RO/ST=Arges/L=Pitesti/O=UPIT/OU=FECC

Technology/CN=nomane/emailAddress=me@m.com

Getting Private key

5. Instalare cheie privata și certificat

```
C:\xampp\apache\bin>copy server.crt c:\xampp\apache\conf\ssl.crt
```

```
C:\xampp\apache\bin>copy server.key c:\xampp\apache\conf\ssl.key
```

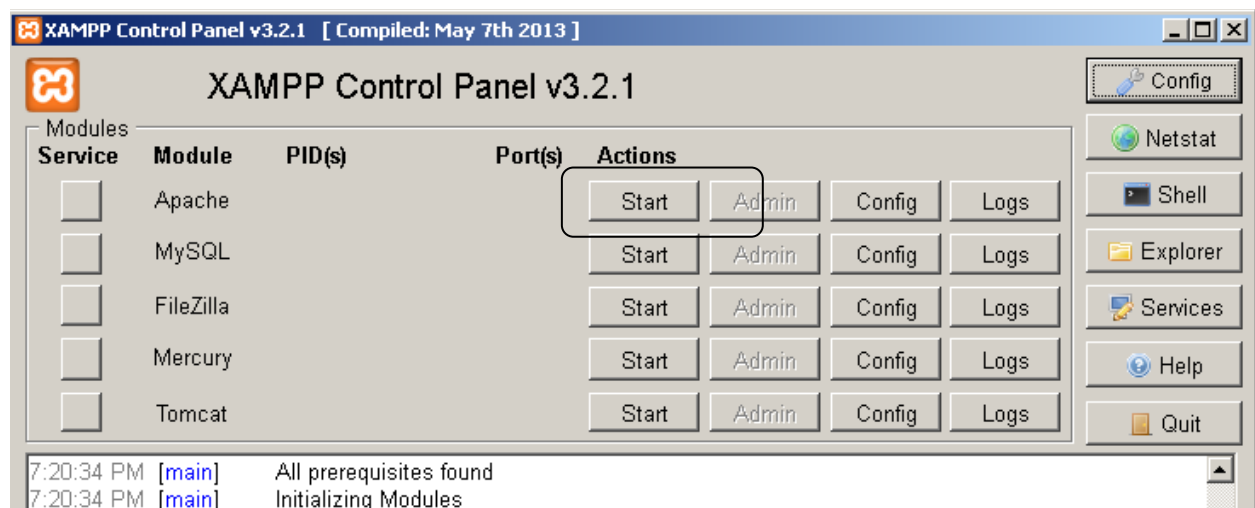


Figura 12 Panoul de control XAMPP

În acest moment serverul este configurat. Serverul Apache va necesita repornirea dacă este deja pornit. Aceasta acțiune se realizează din panoul de control prin apăsarea butonului de Start. După apăsarea acestui buton de vor afișa în panoul de control porturile pe care poate fi accesat serverul (443 indica protocolul HTTPS).

Desfășurarea lucrării

1. Se va studia breviarul teoretic.
2. Se va deschide pagina <https://www.google.ro>. Se va da click pe lacătul care prezintă elementele de identificare a certificatului folosit de această pagină.
3. Se va deschide pagina <https://mail.upit.ro>. Pagina folosește certificat self-signed. Se va confirma excepția de securitate pentru a naviga la aceasta pagina. Se va da click pe lacătul care prezintă elementele de identificare a certificatului folosit de această pagină.

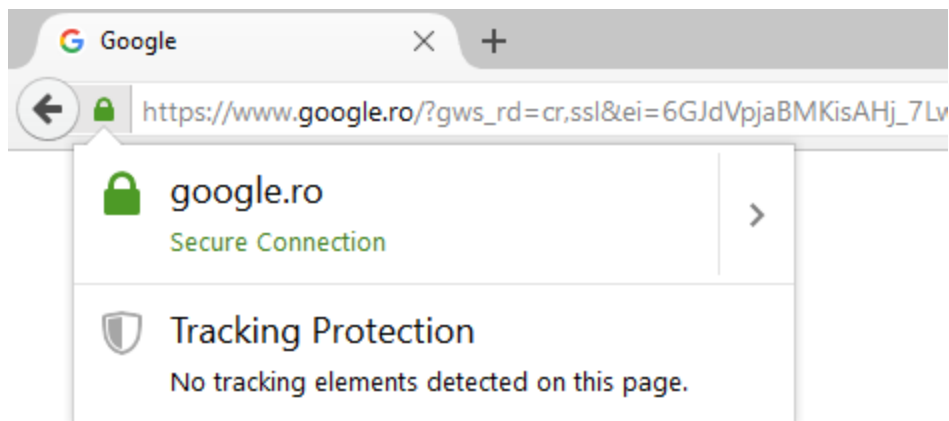


Figura 13 Pagina HTTPS care are certificat de încredere pentru browser

Dacă accesăm deseori un site care are cheie auto-semnată (self-signed), ea se poate adăuga permanent la lista de certificate recunoscute de un sistem de calcul.

Pentru Google Chrome acest lucru se realizează prin salvarea certificatului local pe calculator, apoi în Settings -> Preferences -> Under the Hood -> Manage Certificates se importa certificatul salvat anterior.

4. Se va configura un server HTTP cu cheie auto-semnată și se va observa comportarea browserului la deschiderea acestui site.
5. Se va monitoriza comunicația din rețea cu acest site folosind aplicația Wireshark.

Capitolul 5. Modelarea traficului în Linux

Obiectivul lucrării

Lucrarea arată modul în care putem controla traficul care trece printr-un echipament de rețea în scopul controlului calității serviciului sau pentru testarea aplicațiilor în condiții diferite de trafic.

Kerneul Linux permite funcții avansate de control și modelare a traficului prin folosirea comenzii tc (Traffic Control). Modelarea traficului se poate realiza doar pentru protocoalele care au suport în acest sens. De exemplu protocolul TCP implementează un mecanism prin care crește fluxul de date după fiecare transmisie reușită până când ocupa toată lățimea de bandă (sau este limitat de sursă). În cazul în care unele pachete sunt pierdute, fluxul de date este redus. Prin urmare simplă eliminare a pachetelor la intrarea în ruter permite controlul traficului pentru acest protocol. Alte protocoale însă, precum UDP, nu sunt sensibile la pierderile de pachete și acest mecanism nu poate fi folosit.

Breviar teoretic

Latenta în rețelele de calculatoare este o componentă importantă a calității serviciului deoarece exprimă timpul necesar pachetelor pentru a ajunge de la sursă la destinație. Cu cât latența este mai mare cu atât calitatea serviciului scade. Anumite servicii de rețea sunt mai sensibile la creșterea latenței sau la variația latenței decât altele.

Componentele latenței în rețele de calculatoare

Considerăm P ca fiind lungimea unui pachet exprimată în biți, L lungimea traseului de date exprimată în metri și R rata de transfer exprimată în biți pe secunda. Atunci putem defini:

- **întârzierea de propagare** ca fiind timpul necesar unui bit pentru a parcurge lungimea traseului (L) de comunicație, circulând cu viteza c (viteza de propagare prin mediul de transmisie):

$$\text{PROP} = \frac{L}{c}$$

De exemplu întârzierea de propagare prin liniile de cupru (cablu torsadat sau cablu coaxial) este de 4.76 microsecunde pe kilometru. Deoarece liniile de comunicație sunt foarte lungi, valoarea aceste întârzieri este de ordinul milisecundelor.

- **întârzierea de transmisie** este timpul necesar pentru a transmite un pachet de lungime P. Acest timp apare deoarece transmisiile în rețele de calculatoare sunt seriale și este necesar un anumit timp pentru a transfera datele din bufferul de transmisie în linia de comunicație (respectiv preluarea datelor din linie în bufferul de recepție). De exemplu pentru o linie de comunicație Ethernet de 1Gbps durează 10 microsecunde pentru a recepționa 10000 de biți.

$$\text{TRANSM} = \frac{P}{R}, \text{ unde } R \text{ este rata de transmisie (b/s)}$$

Într-o rețea ideală latența totală este prin urmare suma celor doi timpi:

$$\text{Latența} = \text{PROP} + \text{TRANSM};$$

Reprezentând grafic aceste valori se obține graficul următor:

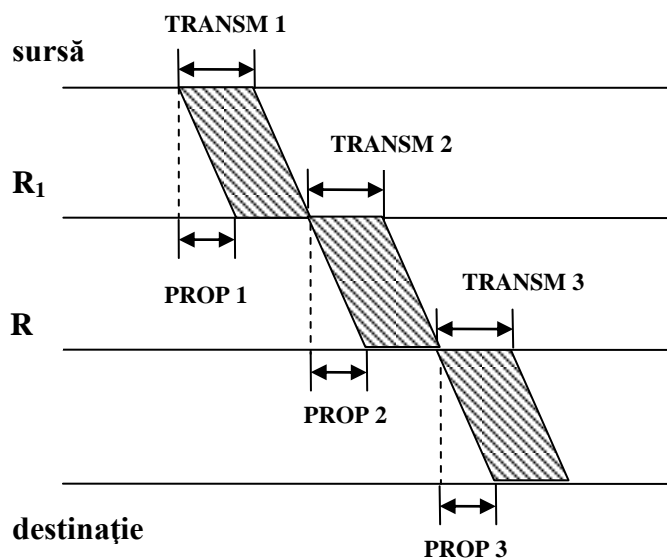


Figura 1 Componentele latenței datorate întârzierii de propagare și transmisie

Prin urmare valoarea minimă a latenței pentru transmiterea tuturor pachetelor este:

$$\text{Latența} = \sum_{i=0}^n (\text{TRANSM}_i + \text{PROP}_i)$$

Din păcate în rețelele reale echipamentele intermediare (rutere, switch-uri) nu procesează datele instantaneu, fiecare având o coadă de așteptare variabilă care este procesată secvențial. Suplimentar, orice echipament are o întârziere implicită, datorată traversării datelor de la intrarea în echipament până la ieșirea din echipament, plus timpul de căutare a în tabela de adrese.

Întârzierea depinde la switch-uri de arhitectura acestuia (cut-through sau store-and-forward) și este mult mai mică decât la rutere.

La rutere într-un interval de timp scurt sau chiar în același timp la un echipament pot sosi mai multe pachete simultan, fiecare având destinații diferite. Echipamentele care direcționează

traficul trebuie sa calculeze pentru fiecare pachet destinația însă, deoarece calculul nu se poate efectua simultan, unele vor fi procesate mai rapid și altele cu o anumită întârziere. Datele care urmează a fi procesate vor fi stocate temporar într-o structura de tip coada. Întârzierea datorată cozii de așteptare Q este singura variabilă în rețeaua de calculatoare.

Această componentă face ca latența să nu fie compusă doar din componente de valoare constantă. Prin urmare, notând cu Q_i întârzierea datorată cozii de așteptare formula de calcul a latentei se modifică în modul următor:

$$\text{Latenta} = \sum_{i=0}^n (\text{TRANSP}_i + \text{PROP}_i + Q_i)$$

Etapale parcurse de un pachet în traversarea unui echipament de rețea cu funcții de calitate a serviciului

În rețelele de calculatoare traficul care traversează un ruter ce oferă funcția de calitate a serviciului parcurge etapele de CQS (Classification Queing Scheduling):

- clasificare: pachetele recepționate vor fi direcționate către una din clasele disponibile în funcție de caracteristicile sale;
- plasarea în coada de așteptare: pachetele vor rămâne în coada de așteptare până când vor fi scoase de un planificator.
- planificarea pentru ieșirea în rețea: planificatorul determină care pachet va fi extras din coada (sau cozile) de așteptare pentru a fi trimis în rețea. Trebuie să fie cât mai simplă, dar totuși să îndeplinească condițiile solicitate: latența, variația latenței, etc.

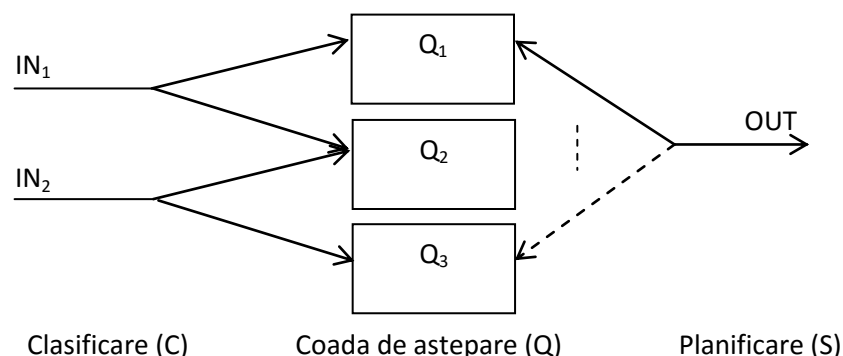


Figura 2 Etapele pe care le parcurge un pachet la traversarea unui echipament de rețea cu funcții de calitate a serviciului

Modelarea traficului (traffic shaping) este un alt mecanism pe care aceste echipamente de rețea îl pot implementa, prin care pachetele sunt întârziate pentru a îndeplini cerințe legate de rata de ieșire. Deoarece acest mecanism limitează traficul, el are ca efect secundar reducerea vârfurilor de trafic care nu sunt de dorit într-o rețea deoarece pot încălca dincolo de posibilitățile de procesare echipamentele de rețea (și pot genera creșterea excesivă a latentei sau pierderi de pachete).

Traffic Control (tc)

Utilizatorul Traffic Control permite: modelarea, planificarea ieșirilor, politica aplicată intrărilor și eliminarea traficului.

qdisc este termenul folosit pentru disciplina de punere în coada de așteptare („queuing discipline”) și este o componentă fundamentală a controlului de trafic. Când kernelul sistemului de operare dorește să trimită un pachet, el este plasat în *qdisc*-ul asociat acelei interfețe. În continuare kernelul încearcă să scoată cât mai multe pachete din *qdisc*, pentru a fi trimise prin adaptorul de rețea.

În mod implicit este setată politica *fifo_fast*.

Există două tipuri de discipline de punere în coada de așteptare:

- cu clase: Aceste discipline permit crearea claselor de filtrare a traficului. Fiecare clasă are propriul set de reguli pentru filtrarea traficului. În plus, fiecare clasă poate avea asignată alte discipline de plasare în coada de pachete, cu sau fără clasă.

- fără clase: Nu permit adăugarea de mai multe discipline de plasare în coada de pachete.

Comanda care șterge orice disciplină de plasare în coada de pachete setată pentru interfața *eth0* este următoarea:

```
tc qdisc del root dev eth0
```

Pentru a adăuga o disciplină de plasare în coada pentru *eth0* se folosește:

```
tc qdisc add dev eth0 root QDISC ParametriQDISC
```

În continuare vor fi prezentate cele mai folosite tehnici pentru disciplina de punere în coada de așteptare.

Discipline de plasare în coada de pachete fără clase

Aceste cozi permit managementul simplu al traficului prin reordonare, încetinind sau eliminând pachete. Ele nu permit crearea de clase de trafic.

pfifo

Este cel mai simplu qdisc, care implementează mecanismul FIFO. Permite limitarea traficului prin numărul de pachete sau numărul de octeți.

pfifo_fast

Este politica utilizată în mod implicit pe root pentru dispozitivele care nu au fost configurate. Algoritmul folosit este First In First Out, pachetele de date fiind procesate în ordinea sosirii. Deși nu se efectuează o procesare, coada permite păstrarea temporară a pachetelor dacă placa de rețea nu le poate prelua momentan.

Random Early Detection (red)

Random Early Detection simulează congestia în rețea prin eliminarea de pachete în mod aleatoriu atunci când traficul se pregătește să depășească lățimea de banda maximă alocată. Fata de cozile simple care doar elimina pachetele în cazul în care coada de așteptare este plină, RED realizează eliminarea pachetelor cu mult înainte de a se atinge lățimea de banda maximă. În acest mod retransmișile datorate pachetelor picate nu se vor realiza toate în același timp ci în mod treptat, încărcând treptat, în timp, rețeaua. Cu cât numărul de pachete în coada de așteptare este mai mare, cu atât probabilitatea de eliminare a unui pachet este mai mare.

Câteva din filtrele ce pot fi utilizate: min – dimensiunea medie a cozii pentru care marcarea pachetelor pentru eliminare devine posibilă; max – dimensiunea medie a cozii pentru care probabilitatea de marcarea este maximă; probability – probabilitatea maximă de marcarea exprimată de un număr între 0.0 și 1.0; burst – folosită în determinarea vitezei cu care calculul a valorii medii a cozii, ia în calcul valoarea reală a cozii de pachete (o valoare mare face ca algoritmul RED să constate modificarea valorii medii mai lent și să înceapă marcarea pachetelor mai târziu); avpkt – valoare specificată în octeți, folosită împreună cu burst pentru a determina constanta de timp pentru calculul dimensiunii medii a cozii;

```
tc qdisc add dev eth0 root red limit 4000 min 300 max 900 probability 0.2 avpkt 10 burst
```

Token Bucket Filter (tbf)

Aceasta politica permite trecerea pachetelor dacă nu a fost depășit un nivel limitat stabilit.

Algoritmul Token Bucket are rolul doar de a modela traficul și nu realizează o planificare a acestuia. În acest algoritm se folosesc jetoane pentru a controla fluxul de pachete astfel:

- pentru fiecare jeton generat, un pachet pleacă pe o interfață;
- jetoanele se generează în mod constant în funcție de regulile configurare. Numărul maxim de jetoane care pot fi generate dacă nu sunt pachete disponibile este egal cu traficul maxim care este ieși într-o singura transmisie;
- dacă nu sunt jetoane disponibile, pachetele sunt adăugate într-o coada de așteptare, pana la limita maxima configurata. Dacă în acest caz sosesc jetoane, pachetele vor ieși imediat producând pentru perioade scurte de timp trafic de nivel foarte mare. Această a doua situație nu este de dorit. Dacă jetoanele nu sosesc suficient de rapid, pachetele care sunt în așteptarea unui jeton, vor fi eliminate;

Parametri folosiți:

- limit este numărul de octeți care pot fi plasați în coada în așteptarea jetoanelor. Este folosita complementar comenzii latency.
- latency specifica timpul maxim pe care un pachet poate sa îl petreacă în coda de așteptare. Este folosita complementar comenzii limit.
- burst este numărul de octeți maxim care pot ieși într-o singura transmisie. Dimensiunea pentru burst trebuie corelata cu lățimea de banda maximă. De exemplu pentru 10Mbps trebuie alocati 10kB pentru buffer. Dacă bufferul este prea mic, viteza maxima de transfer nu va putea fi atinsa.

Următoarea comanda realizează pentru dispozitivul eth0 o limitare a vitezei de 250kbps, pachetele pot sta în coada de așteptare 100ms înainte de a fi eliminate și un maxim de trafic de 1,5kbps.

```
tc qdisc add dev eth0 root tbf rate 250kbit latency 100ms burst 1500
```

Stochastic Fairness Queueing (SFQ)

Această disciplină de plasare în coada de pachete grupează conversațiile în fluxuri („hash bucket”) și fiecare flux are, pe rând, posibilitatea transmiterii date conform unui algoritm Round-Robin. În acest mod fiecare flux are posibilitatea de a transmite date și previne situația în care un flux ar transmite semnificativ mai multe date decât celelalte fluxuri.

Clasificarea în fluxuri se face în urma calculării unui hash pe baza adresei sursă, a destinației și a portului sursă. Deoarece uneori fluxuri multiple pot genera același hash și în acest mod vor fi grupate incorect, este necesara stabilirea unui interval la care se vor recalcula (numit perturb) hash-urile. În acest mod gruparea incorectă durează foarte puțin timp.

În cazul în care numărul de pachete depășește lungimea maximă a cozii de așteptare, SFQ elimină pachete din fluxul care are cele mai multe pachete în așteptare. SFQ devine util doar în cazul în care interfața de transmisie este complet ocupată altfel nu există cozi de așteptare și algoritmul nu intră în acțiune.

În următorul exemplu se setează pentru interfața eth0 disciplina SFQ și se stabilește un interval de 10 secunde la care să se recalculeze hash-ul pentru a evita grupare incorectă:

```
tc qdisc add dev eth0 root sfq perturb 10
```

Discipline de plasare în coada de pachete cu clase

Disciplinele de plasare în coada de pachete cu clase sunt necesare dacă există mai multe tipuri de trafic pe care dorim să le tratăm diferit. Fiecare clasă de trafic creată are nevoie de un nume dar și de un identificator pentru părintele clasei. Părintele poate fi clasa rădăcină (root) a unei interfețe sau o altă clasă.

Numele vor fi setate cu *nume_radacina:nume_clasa* în care *nume_radacina* este de obicei 1 iar numele clasei este cel dorit.

Hierarchical Token Bucket (HTB)

Această disciplină de plasare în coada de pachete este potrivită pentru situațiile în care lățimea de bandă trebuie împărțită între mai multe tipuri de trafic, fiecare având specificată lățimea de bandă garantată. HTB modelează traficul folosind algoritmul Token Bucket.

Într-o instanță HTB pot exista mai multe clase, fiecare dintre ele putând fi împărțită în mai multe sub-clase. În mod implicit se folosește pfifo.

Când plasează un pachet în coada de așteptare, HTB începe de la root și determină în continuare de la nod la nod care clasa ar trebui să recepționeze datele. Dacă clasa nu are alte sub-clase, pachetul este adăugat la acea coadă de așteptare. În cazul în care sunt sub-clase, se continuă procesul de căutare.

Configurarea HTB pentru interfața eth0 urmează următorii pași:

1. se șterge metoda PFIFO_FAST dacă acesta este configurat:

```
tc qdisc del dev eth0 root
```

2. se adaugă disciplina HTB la nivelul root (1):

```
tc qdisc add dev eth0 root handle 1: htb
```

Suplimentar poate fi precizata o clasa care va fi folosita în mod implicit pentru pachete cu ajutorul comenzii:

```
tc qdisc add dev eth0 root handle 1: htb default 20
```

3. Se creează clasele 10 și 20 în care vor fi plasate pachetele și se specifica pentru fiecare clasa rata dorita de ieșire:

```
tc class add dev eth0 parent 1: classid 1:10 htb 200kbit
```

```
tc class add dev eth0 parent 1: classid 1:20 htb 1mbit
```

4. Se adaugă filtrele care vor selecta care pachete vor ajunge în clasa 10 și care pachete vor ajunge în clasa 20. Formatul general este:

```
tc filter add dev eth0 protocol ip parent 1: prio 1 u32 match ip dst [Adresa IP]/[Masca] flowid 1:[Id flux]
```

În cazul următor este folosita adresa sursă a pachetului (src) pentru a determina unde va fi plasat pachetul:

```
tc filter add dev eth0 protocol ip parent 1: prio 1 u32 match ip src 10.0.0.1/255.0.0.0  
match ip dport 80 flowid 1:10
```

```
tc filter add dev eth0 protocol ip parent 1: prio 2 u32 match ip dst 10.0.0.1/255.0.0.0  
match ip dport 22 flowid 1:20
```

Vizualizarea disciplinei setate se face cu:

```
tc filter show dev eth0
```

Aplicații

Network Emulation (netem)

Utilitarul netem asigură funcționalitatea necesară pentru a testa protocoalele de rețea prin emularea proprietăților unei rețele. Se asigură suportul pentru a emula:

- introducerea latentei în traficul de rețea;
- variația întârzierii (jitter) în mod variabil sau constant;
- pierderea de pachete;
- duplicarea pachetelor;
- reordonarea pachetelor.

Netem este configurat cu ajutorul utilitarul în linie de comandă tc.

Introducerea latentei în traficul de rețea

Latenta are următoarele componente:

- întârzierea de propagare;
- întârzierea de transmitere;
- întârzierea datorată cozii de procesare a echipamentelor de rețea (în special ruterele).

Pentru a adăuga o întârziere se folosește comanda:

```
tc qdisc add dev eth0 root netem delay 200ms
```

Variația întârzierii (jitter)

Variația întârzierii este un factor important de influență a calității comunicațiilor de voce. Dacă se dorește ca întârzierea introdusă să se apropie de cea a unei rețele reale, se poate folosi comanda:

```
tc qdisc change dev eth0 root netem delay 100ms 10ms
```

Această comandă introduce o întârziere de 100ms cu o variație de +/- 10ms.

Uneori variațiile latentei dintr-o rețea nu sunt complet aleatorii. Când ruterele devin încărcate datorită numărului mare de conexiuni sau a traficului susținut în rețea, ele introduc perioade mai lungi în care variațiile latentei cresc sau scad puțin în jurul unor valori. Pentru a emula acest comportament se poate adăuga pentru fiecare pachet o întârziere a cărei variație să difere de valoarea anterioară doar într-un anumit procent. Această variație poate fi simulată cu ajutorul comenzii:

```
tc qdisc change dev eth0 root netem delay 100ms 10ms 25%
```

Pierderea de pachete

Comunicațiile prin rețele de calculatoare au uneori pierderi de pachete, datorate echipamentelor defecte, coliziunilor cu alte pachete, mediului de transmisie care introduce zgomot în semnalul transmis cât și din alte cauze. Este important să analizăm ce se întâmplă în aceste situații cu aplicațiile pe care le folosim.

Pentru a simula pierderea de pachete se folosește comanda:

```
tc qdisc change dev eth0 root netem loss 0.2%
```

Aici 2 din 1000 pachete este eliminat în mod aleatoriu. Similar cu situația anterioară, se poate preciza suplimentar o marja de variație a pierderii de pachete, în acest mod se pot simula pierderile grupate de pachete.

```
tc qdisc change dev eth0 root netem loss 0.3% 25%
```

Duplicarea pachetelor

Dacă un protocol de rețea decide ca un pachet este pierdut și trebuie retransmis, dar acesta este în realitate doar întârziat în mod excesiv, este posibil ca pachete duplicate să ajungă la destinație. Deoarece aceasta situație care nu este de dorit poate să apară destul de rar, pentru a o simula cu netem se introduce:

```
tc qdisc change dev eth0 root netem duplicate 1%
```

Reordonarea pachetelor

Atunci când circula pe internet, pachetele sunt direcționate (rutate) individual în funcție de condițiile din rețea. Este prin urmare posibil ca pachetele să nu ajungă la destinație urmând același traseu deci pot să ajungă în ordine diferită de cea în care au fost trimise. Acest comportament poate fi simulat cu netem prin comanda:

```
tc qdisc change dev eth0 root netem gap 5 delay 10ms
```

Această comandă permite trecerea unui singur pachet din 5 fără întârziere, celelalte 4 având o întârziere de 10ms. În acest mod ordinea în care pachetele ajung la destinație va fi modificată.

Desfășurarea lucrării

1. Se va studia breviarul teoretic.
2. Se va construi o rețea de forma:

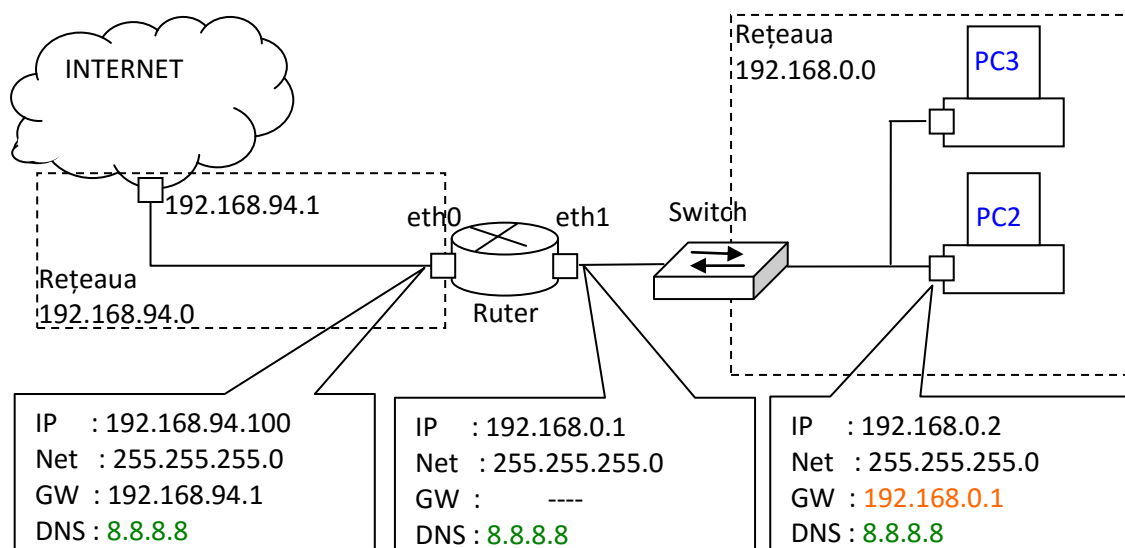


Figura 3 Structura rețelei de test pentru modelarea traficului

Ruterul din aceasta rețea va fi implementat în Linux și se vor testa pe acesta diferiți algoritmi de planificare a mesajelor. Configurația prezentată permite selecția traficului și în funcție de adresa IP.

3. Se va folosi aplicația ping pentru a testa efectele setărilor efectuate pe ruter care vizează traficul pe calculatoarele PC1 și PC2. Traficul va fi monitorizat în Windows cu *TaskManager* → *Performance* → *Open Resource Monitor* pentru tabul Network, iar în Linux cu utilitarul de consola *iptraf*.

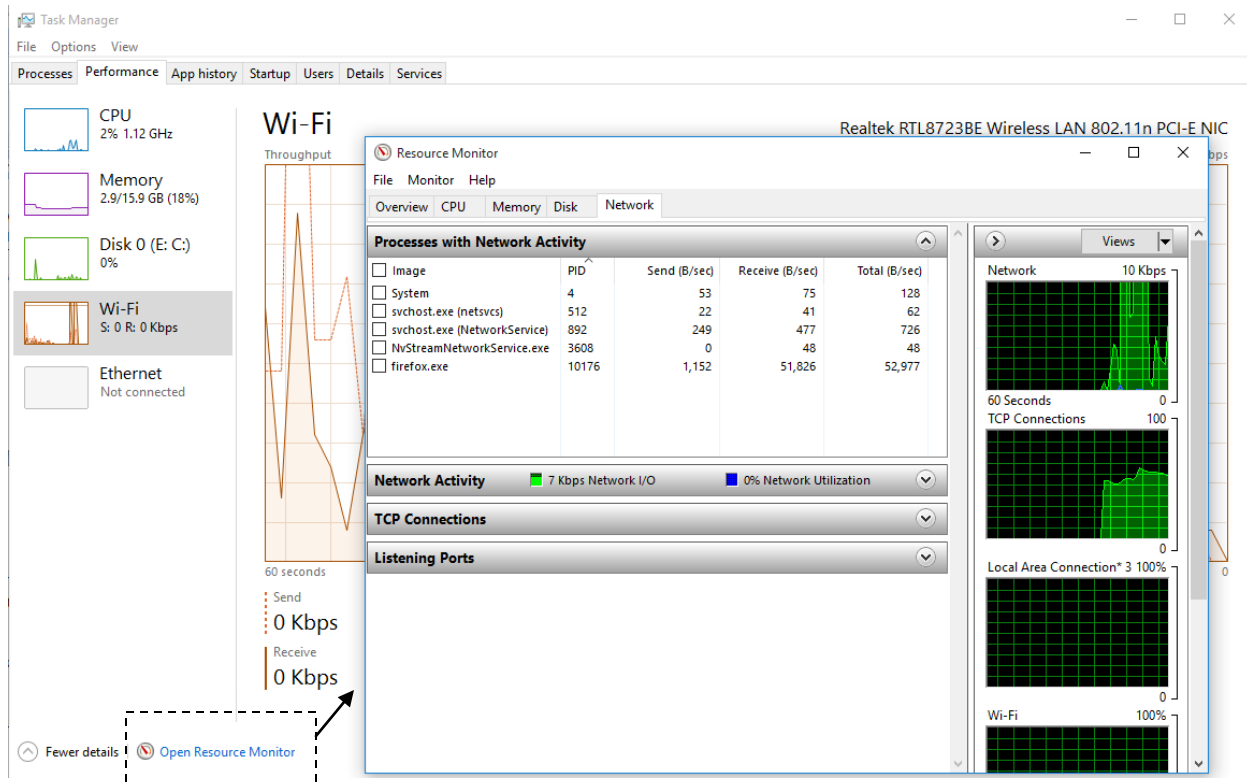


Figura 4 Monitorizarea traficului în Windows din Resource Monitor

4. Se va folosi aplicația Wireshark pentru a monitoriza schimbul de pachete din rețea.

Capitolul 6. Transmiterea fluxurilor multimedia. Port forwarding

Obiectivul lucrării

Lucrarea își propune prezentarea noțiunilor teoretice și configurarea unui sistem pentru transmiterea unui flux video în timp real.

Nivelul transport asigură transmiterea informației de la sursă la destinație furnizând mijloacele pentru depistarea apariției erorilor și reasamblarea ordonată a informației la destinație. Cele mai cunoscute protocoale sunt: Transmission Control Protocol (TCP), User Datagram Protocol (UDP). Majoritatea comunicațiilor multimedia în rețele locale de calculatoare se bazează pe aceste două protocoale. Unitatea de protocol la nivelul transport este segmentul pentru protocolul TCP și datagrama pentru protocolul UDP.

Aplicațiile din acest laborator vor viza implementarea unui sistem de transmisii de tip flux multimedia în rețeaua locală cât și în rețelele învecinate prin folosirea funcției de *port forwarding*.

Breviar teoretic

Transmission Control Protocol

TCP (Transmission Control Protocol) este un protocol orientat pe conexiune deoarece înainte de a transmite datele utilizatorului sursă realizează un schimb de date cu destinația pentru a determina dacă aceasta este prezentă și dacă este disponibilă pentru a transfera datele utilizatorului. Dacă acest schimb de date reușește se va stabili o conexiune prin care pot circula date în mod bidirecțional. Închiderea conexiunii se face tot cu un schimb dedicat de date. În cazul în care una dintre părți nu reușește închiderea conexiunii în mod normal, există implementat un mecanism de expirare a conexiunii în timp (time-out).

Algoritmul TCP este similar desfășurării unei convorbiri telefonice între două persoane deoarece informația care nu este înțeleasă este retrimisă imediat.

TCP – ul este orientat pe conexiune și folosește algoritmul în 3 pași de stabilire a conexiunii. Algoritmul de închidere a conexiunii are 4 pași. TCP transmite octeții în grupuri de octeți numite segmente.

TCP este fiabil:

- recepția de date sunt confirmată cu un ACK;
- se folosesc sume de control pentru depistarea datelor eronate;
- se folosesc numere de secvență pentru reasamblarea în ordine a datelor primite;
- datele eronate se retransmit după o pauză;
- TCP realizează controlul traficului de date pentru a împiedica umplerea bufferului

de recepție.

User Datagram Protocol

UDP (User Datagram Protocol)– nu are o secvență de inițializare, prin urmare pachetele pot urma oricând, fără a fi anunțate în prealabil. Are o structură mai simplă, deci pachetul de date va fi mai mic.

Pentru serviciile care nu sunt critice (DNS) se recomandă utilizarea UDP, ca protocol de suport, deoarece încarcă mai puțin rețeaua.

Datele transmise de UDP sunt însoțite de o sumă de verificare care permite împreună cu indicatorul de lungime să se ia decizia dacă datele au ajuns corect la destinație.

Nu conține mecanisme pentru detectarea pachetelor lipsă sau celor care nu sunt în ordine. Nu există mecanism pentru controlul traficului de date.

Printre cele mai frecvente aplicații UDP sunt: DNS, DHCP, aplicații audio – video. Alte aplicații care nu au nevoie de transmisii sigure sunt aplicațiile de monitorizare a rețelei.

Tipuri de fluxuri

Aplicația VLC folosită în teste permite crearea mai multor tipuri de fluxuri video:

1. UDP / RTP - Acesta este un flux unicast, ceea ce înseamnă că nu există un alt calculator trimite un flux direct la computer. Trebuie să setați portul UDP corect (acest lucru este stabilit de oricine inițiază trimiterea fluxului către calculatorul client; implicit este 1234 pentru fluxurile trimise prin VLC)

2. UDP / RTP Multicast - Aceasta este versiunea multicast a precedente, ceea ce înseamnă că serverul trimite fluxuri de mai multe calculatoare în același timp. Trebuie să setați adresa corectă și portul UDP (încă o dată, stabilit de server, deci de către cel care înființarea fluxul)

3. HTTP / FTP / MMS - Aceasta folosește protocolul HTTP (Shoutcast, etc.), FTP sau flux Microsoft Media Server - în general tipul de fluxuri care se regăsesc pe site-urile web. Pur și simplu completează URL-ul complet în câmpul URL pentru accesarea acestui flux.

4. RTSP - Real Time Streaming Protocol, acesta este un alt mod de a trimite un flux multimedia. Doar se completează URL-ul în caseta de text pentru a funcționa.

Port forwarding

Port forwarding (sau Port Address Translation) se referă la procesul realizat de un router prin care redirecționează cererile primite către un IP și un număr de port către un alt IP și port. După cum se observă port forwarding este însoțită de traducerea adreselor de rețea (NAT).

În acest proces numărul portului poate fi păstrat sau schimbat.

Dacă se alege să se păstreze numărul portului, acest lucru se face pentru a putea implementa servicii foarte cunoscute pe diferite calculatoarele din rețeaua locală însă accesarea acestora să se facă dintr-un punct comun. Servicii uzuale sunt HTTP (portul 80), HTTPS (portul 443), FTP (porturile 20 și 21), etc.

Dacă se alege modificarea portului, aceasta se efectuează din mai multe cauze dintre care enumerăm:

- motive de securitate în cazul în care se alege ca un port mai puțin cunoscut să fie expus internetului (de exemplu portul 9000), urmând ca traducerea să direcționeze traficul trimis pe acest port către o aplicație bine cunoscută (de

exemplu HTTP, portul 80). Aceasta translată a porturilor se face pentru a preveni atacurile automate către un port bine cunoscut.

- pentru evitarea conflictelor. Dacă de exemplu se dorește testarea mai multor servere web fără ca serviciul să nu fie accesat din eroare de către un utilizator, se poate expune portul 80 pentru un server web și portul 8080 pentru alt server.
- motive de compatibilitate cu o configurație de rețea existentă. Dacă setările unui firewall din rețea previn conectarea directă a calculatoarelor din rețeaua locală la un anumit IP și port, este posibil să deschidem un tunel SSH către un terț calculator, care va realiza pentru noi port forwarding între portul local și portul de la IP-ul restricționat.

În Linux, port forwarding se poate realiza prin reguli de filtrare introduse în iptables sau netfilter.

În Unix/Linux unde un număr de port sub 1024 poate fi creat numai de către software-ul care rulează ca administrator (root), port forwarding este folosit pentru a redirecționa traficul de intrare de la un port cu număr sub 1024 către un port cu număr mai mare. Acest software poate rula prin urmare ca un utilizator normal, evitând riscurile de securitate cauzate de rularea ca un administrator.

iptables

Iptables filtrează pachete în funcție de tipul pachetului, clasificate conform unor tabele predefinite. În total sunt trei tabele:

- mangle;
- filter;
- nat.

Pentru a inspecta aceste tabele vom folosi comenzile în modul privilegiat:

```
iptables -t nat -L
```

```
iptables -L
```

Tabelul mangle este responsabil pentru modificarea biților de serviciu din antetul protocolului TCP. Tabelul filter este responsabil cu filtrarea pachetelor. Tabelul nat efectuează translatărea adreselor de rețea (Network Address Translation-NAT). Fiecare tabel poate avea câteva lanțuri implementate în care pot fi plasate regulile de politica firewall.

Tabelul de filtrare are trei lanțuri implementate:

- Lanțul FORWARD: filtrează pachete destinate rețelelor protejate de către firewall
- Lanțul INPUT: filtrează pachete destinate pentru firewall
- Lanțul OUTPUT: filtrează pachete care provin din firewall

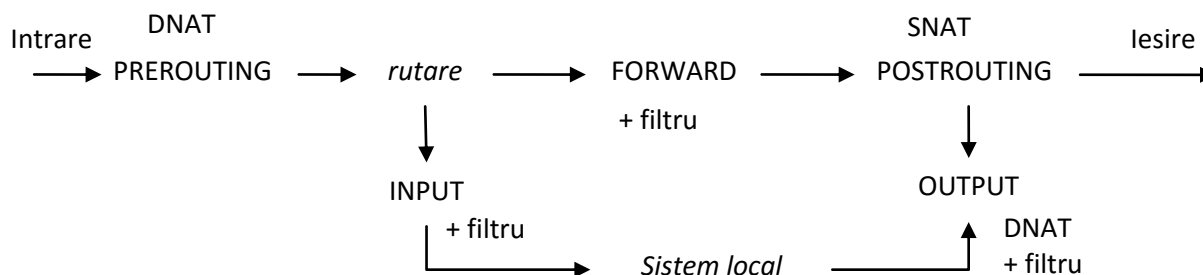


Figura 1 Lanțul de procesare a pachetelor într-un ruter Linux

Tabelul nat are implementate următoarele lanțuri:

- Lanțul pre-routing: translatează adrese de rețea pentru pachete atunci când adresa destinație a pachetului trebuie să fie schimbată
- Lanțul post-routing : translatează adrese de rețea pentru pachete atunci când adresa sursă a pachetului trebuie să fie schimbată
- Lanțul output : translatează adrese de rețea pentru pachete care provin din firewall.

Pachetul care intra în firewall este inspectat de regulile lanțului PREROUTING din tabelul NAT pentru a vedea dacă este necesară modificarea destinației (DNAT). Dacă pachetul este destinat sistemului local nu se face nici o modificare la adresa destinație, altfel se face înlocuirea corespunzătoare tabelului de traducere a adreselor.

Pachetul este apoi trimis către secțiunea de rutare în care se caută adresa destinație. Pachetul care este destinat pentru o rețea locală este filtrat de regulile din lanțul FORWARD ale tabelului de filtrare iar cele destinate sistemului local, către lanțul INPUT.

În cele din urmă, se va realiza modificarea adresei sursă în SNAT din lanțul POSTROUTING înainte de a ieși din ruter. Acest lucru este necesar pentru că în pachetele trimise să pară că sunt trimise de la un anumit IP.

De exemplu atunci când mai multe calculatoare din rețeaua locală trimit pachete pe internet, ruterul înlocuiește adresa IP sursă a acestora cu adresa sa IP (publică) deoarece în cazul în care ar trimite pe internet pachete folosind adrese private, serverele destinație nu ar putea ști unde trebuie să trimită răspunsul.

Când serverul destinație răspunde, pachetul se supune aceleiași secvențe de pași.

Aplicații

VLC este un player multimedia (<http://www.videolan.org/vlc/>) gratuit și open-source care are numeroase alte facilități care pot fi folosite pentru a coda, a primi și a trimite fluxurile video în rețea. În continuare se va configura și se va testa transmiterea de fluxuri video bazate pe protocolul TCP și protocolul UDP folosind aplicația VLC.

Meniul „Media” permite transmiterea unui flux în rețea dacă se selectează opțiunea „Stream...” sau recepționarea unui flux din rețea dacă se selectează opțiunea: „Open Network Stream”.

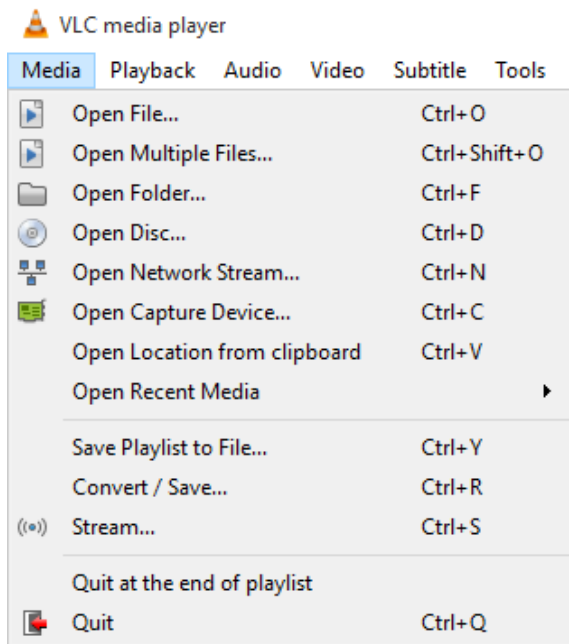


Figura 2 Menul Media al aplicației VLC

Deoarece este necesar ca în testele efectuate sa modificam latența sau sa eliminam pachete din traficul efectuat între sursă și destinație, va trebui sa configuram pe sistemul care este sursă a fluxului de date un utilitar care sa limiteze traficul. În acest sens pentru Windows poate fi utilizată aplicația: NetBalancer (<https://seriousbit.com/netbalancer>) care în versiunea neînregistrată permite limitarea a maxim 3 procese, lucru suficient pentru testare.

În situația în care se dorește testarea complexă a comportamentului fluxului se va plasa un ruter între sursă și destinația traficului (pe care este configurat utilitarul netem corespunzător), însă trebuie să se ia în calcul modificarea valorii TTL la 2 astfel încât sa fie permis traficului sa traverseze ruterul.

Pentru situația în care trimitem trafic cu valoarea TTL=1, deoarece ruterele decrementează valoarea TTL cu 1, pachetele vor fi picate pentru ca TTL va ajunge la 0. De exemplu pentru a adăuga o întârziere se folosește comanda:

```
tc qdisc add dev eth0 root netem delay 200ms
```

Pentru a simula pierderea de pachete se folosește comanda:

```
tc qdisc change dev eth0 root netem loss 0.2%
```

Flux UDP

Se va configura VLC pentru a transmite mai întâi un flux UDP. Se va observa la sursa dacă prezenta sau absenta unui client în rețea afectează transmiterea fluxului de date UDP (Traficul este devine zero? Crește/scade?). Se va observa comportamentul fluxului video în cazul în care lățimea de bandă este limitată (Se pierde cadre din fluxul video în mod vizibil? Se întrerupe redarea fluxului video?).

Configurare sursă flux UDP

- Se selectează din meniul Media opțiunea „Stream...”
- În fereastra nou apărută se selectează fișierul (sau fișierele) video care vor fi transmise în secțiunea *File Selection*.

- Se selectează meniul Stream și se selectează Next pentru a ajunge la meniul „Destination Setup”

- Se alege tipul de stream „UDP – legacy” și se apasă butonul ADD. Dacă se dorește vizualizarea video și pe calculatorul local trebuie bifată selecția din această pagina. În caz contrar nu va fi afișată local decât o bară de progres.

- Se selectează adresa destinație către care va fi trimis fluxul. Dacă se dorește trimiterea la mai multe calculatoare se introduce o adresa de multicast, de exemplu: 224.0.0.1. Se va nota portul pe care se trimit datele pentru a completa corespunzător la destinație. Implicit este portul 1234.

- În fereastra următoare se poate alege tipul de transcodare, de la calitate maximă (H.264+MP4) până la calitate redusă (WMV+WMA), sau se poate alege să se transmită numai partea audio a fișierului video.

- În fereastra următoare, în cazul în care se dorește modificarea TTL pentru a trimite fluxul dincolo de primul ruter (TTL implicit este 1), dacă această opțiune nu apare în interfața grafică, se poate adăuga în stringul rezultat această opțiune sub forma „:ttl=2” în care 2 este valoarea TTL dorită:

String fără TTL:

```
:sout=#transcode{vcodec=h264,acodec=mpga,ab=128,channels=2,samplerate=44100}:udp{dst=224.0.0.1:1234}:sout-keep
```

String cu TTL:

```
:sout=#transcode{vcodec=h264,acodec=mpga,ab=128,channels=2,samplerate=44100}:udp{dst=224.0.0.1:1234}:sout-keep:ttl=12
```

- În final se selectează butonul Stream.

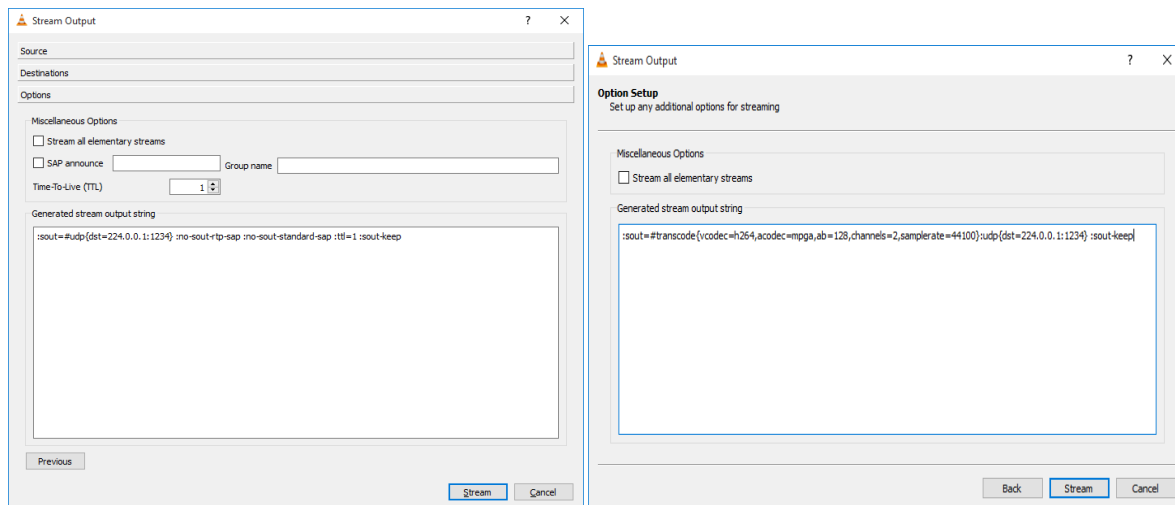
Configurare destinație flux UDP

- Se selectează opțiunea: „Open Network Stream”.

- Se introduce în câmpul text: udp://@224.0.0.1:1234

- Se apasă butonul pentru deschiderea fluxului și se așteaptă până acesta devine vizibil.

În cazul în care nu se deschide fluxul video se va verifica setarea firewall de pe cele două calculatoare. Setările firewall trebuie să permită trecerea protocolului și a portului pe care se face transmisia video (UDP, respectiv portul 1234).

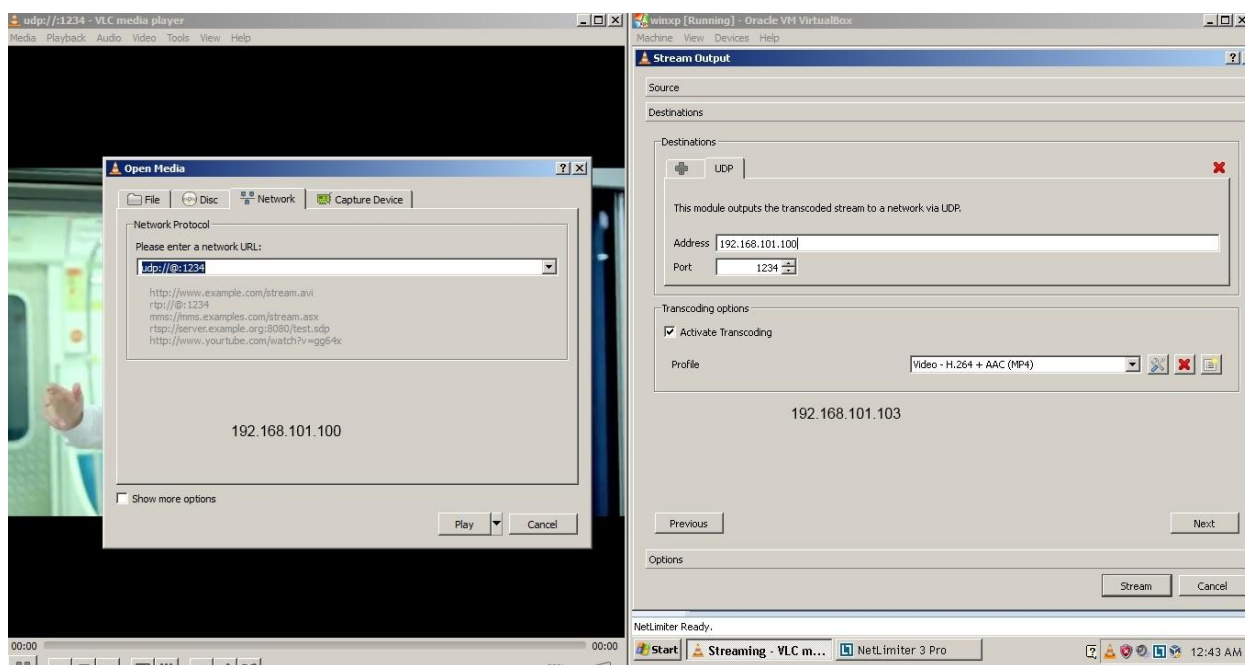


(a)

(b)

Figura 3 Interfața veche a aplicației VLC (a) care are setare grafică pentru TTL în vreme ce în interfața nouă (b) setarea se poate face doar în text

Exemplul din figura următoare este pentru situația în care transmiterea datelor se face folosind o adresa unicast. În mod normal se preferă transmisiile multicast deoarece o singură transmisie poate fi recepționată de mai multe dispozitive, însă și transmisiile unicast pot fi utilizate în cazul în care transmisiile multicast sunt restricționate în rețea.



(a)

(b)

Figura 4 Configurare sursă (a) și destinație (b) flux video bazat pe UDP

Concluzii:

- Fluxul video este transmis cu aceeași viteză indiferent de limitarea sau nu a lățimii de bandă a conexiunii, rezultând pierderi de pachete....
- Se folosește un buffer de recepție. În situația în care videoclipul pierde cadre din fluxul video în mod vizibil, el nu se oprește ci continuă să redea cadre noi care sosesc.
- Recomandat pentru transmisii în timp real

Flux TCP

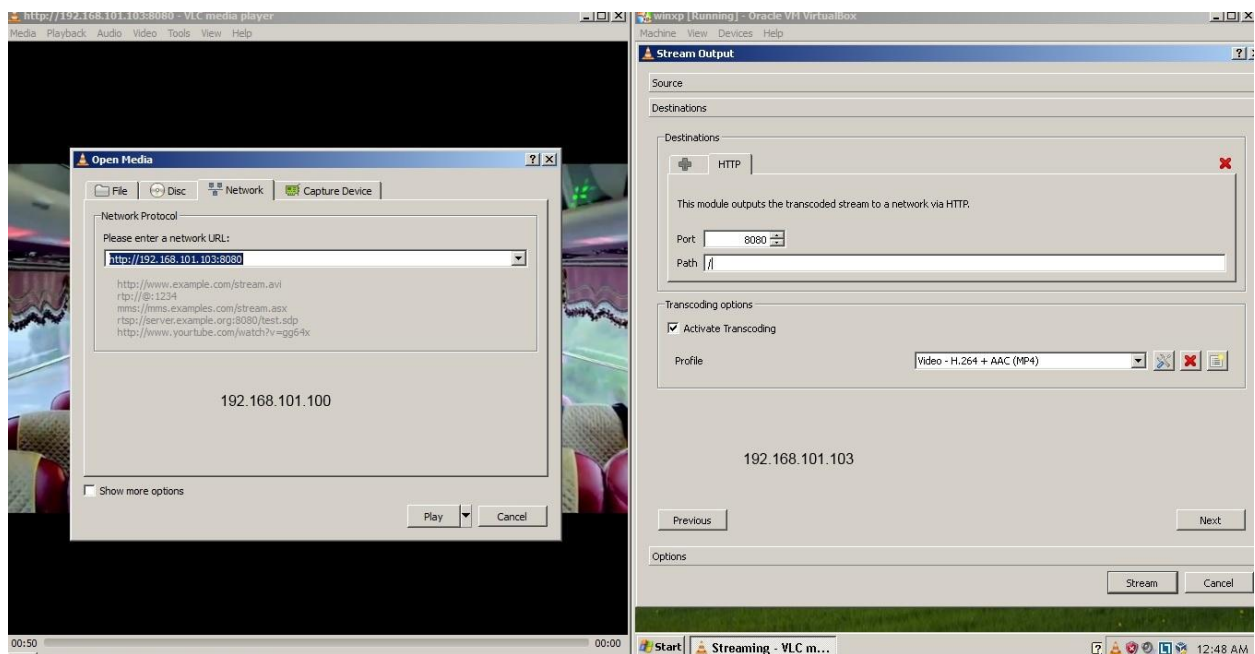
Se va configura VLC pentru a transmite în continuare un flux TCP. Se va observa la sursă dacă prezența sau absența unui client în rețea afectează transmiterea fluxului de TCP (Traficul este devine zero dacă nu sunt clienți conectați?). Se va observa comportamentul fluxului video în cazul în care lățimea de bandă este limitată (Se pierde cadre din fluxul video în mod vizibil? Se întrerupe redarea fluxului video?).

Configurarea sursei flux TCP este similară cu a fluxului UDP cu excepția pasului în care se alege tipul streamului, în loc de „UDP – legacy” alegându-se „HTTP”. Portul implicit de această dată este 8080.

Configurare destinație flux UDP:

- Se selectează opțiunea: „Open Network Stream”.
- Se introduce în câmpul text: <http://IPsursa:8080> în care *IPsursa* este IP-ul calculatorului care trimite traficul în rețea.
- Se apasă butonul pentru deschiderea fluxului și se așteaptă până acesta devine vizibil.

Exemplul din figura următoare este configurat pentru o transmisie tip unicast. Pentru multicast se vor folosi adrese corespunzătoare (ex. 224.0.0.1).



(a) (b)
Figura 5 Configurare sursă (a) și destinație (b) flux video bazat pe UDP

Concluzii:

- Se folosește un buffer de recepție. Limitarea lățimii de banda a conexiunii conduce la golirea buffer-ului de recepție, după care redarea fișierului video se întrerupe, se va afișa un mesaj pe ecran în care se va indica încărcarea buffer-ului de recepție și după ce bufferul este încărcat, se reia redarea fluxului TCP.
- Nu sunt pierdute cadre din fluxul video.
- Recomandat în cazul când dorim recepționarea întregului flux, fără pierderi

Port forwarding folosind iptables

Aceasta secțiune presupune că există un sistem Linux configurat drept gateway și este configurată rutarea pe acesta. Pentru rețeaua de mai jos vom configura iptables pentru a realiza port forwarding.

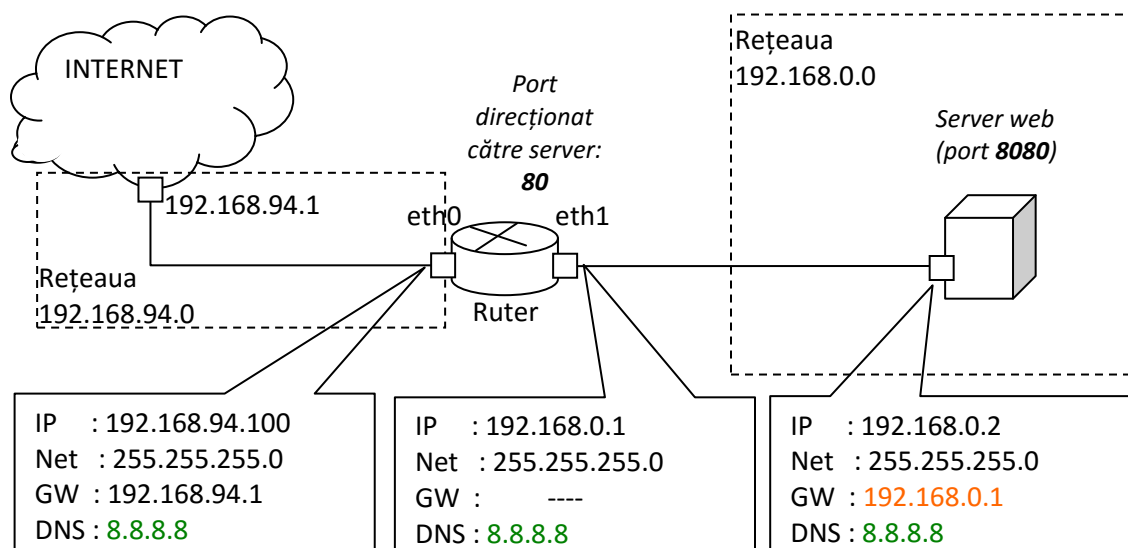


Figura 6 Rețeaua de test folosită în acest exemplu

În exemplul din figura serverul web este situat în rețeaua locală 192.168.0.0, având adresa IP 192.168.0.2. Routerul va deschide portul 80 către internet și va trimite toate cererile primite pe acest port către serverul web.

În mod normal, toate conexiunile de intrare la o mașină gateway sunt ignorate în mod implicit, deoarece deschiderea tuturor serviciilor și porturilor ar putea reprezenta un risc de securitate. Vom deschide prin urmare doar porturile pentru serviciile pe care le vom folosi. În acest exemplu, vom deschide portul 80 pentru serviciul HTTP.

Mai întâi ne vom asigura că opțiunea de IP forwarding este activată pe sistemul Linux:

```
echo 1 > /proc/sys/net/ipv4/ip_forwarding
```

Următoarele sunt comenzile folosite pentru a transmite conexiunile primite pe portul 80 al gateway-ului către portul 8080 de pe mașina din rețeaua privată(192.168.0.2):

```
# iptables -A PREROUTING -t nat -i eth0 -p tcp --dport 80 -j DNAT --to 192.168.0.2:8080
# iptables -A FORWARD -p tcp -d 192.168.0.2 --dport 8080 -j ACCEPT
```

Prima comandă specifica faptul ca toate conexiunile TCP de intrare la portul 80 ar trebui sa fie trimise către portul 8080 al mașinii din rețeaua locală 192.168.0.2. În continuare, prin a doua regulă adăugată în lanțul FORWARD, vom accepta conexiunile de intrare la portul 80 de la eth0 care se conectează la Internet cu IP-ul public.

Schimbarea de port nu este necesara, însa a fost folosita pentru a nu face confuzie între parametrii comenzii iptables.

În exemplu am setat regulile iptables pentru transmiterea portului web (80). Pentru alt serviciu, metoda este similara cu cea a serviciului HTTP.

Observație: Nu uitați ca filtrele se procesează în ordinea în care au fost introduse. Primul filtru în care se încadrează pachetul este cel procesat. De exemplu, dacă avem două filtre care vizează portul 80, primul care permite traficul și al doilea care îl blochează, primul dintre cele doua care a fost introdus este cel care va fi utilizat.

Specificarea unei game de adrese în filtrele iptables

În cazul în care mai multe adrese IP trebuie incluse într-un filtru, iptables suporta opțiunea de a specifica intervalul de adrese IP sau porturi prin iprange. Parametrii sunt:

- src-range ip-ip: verifică potrivirea IP-ului sursă în intervalul specificat.
- dst-range ip-ip: verifică potrivirea IP-ului destinație în intervalul specificat.

Sintaxa:

- m iprange --src-range IP-IP -j ACTION
- m iprange --dst-range IP-IP -j ACTION

De exemplu, pentru a permite cererea de intrare pe un port 22 pentru IP-uri sursă doar în intervalul 192.168.1.100- 192.168.1.200, este nevoie sa adăugam filtrul:

```
iptables -A INPUT -p tcp --destination-port 22 -m iprange --src-range 192.168.1.100-192.168.1.200 -j ACCEPT
```

În mod similar se poate realiza forwarding pentru un grup de porturi. Astfel, dacă este specificat: --protocol tcp (-p tcp), se poate preciza:

- intervalul portului sursă cu următoarea sintaxa:
 - source-port port:port
 - sport port:port
- intervalul portului destinație prin următoarele opțiuni:
 - destination-port port:port
 - dport port:port

De exemplu, pentru a bloca toate intrările către serverul SSH (portul 22) cu IP-ul 192.168.0.1, pentru intervalul de porturi sursă 500:1000 se poate introduce :

```
iptables -A INPUT -p tcp -s 0/0 --sport 500:1000 -d 192.168.0.1 --dport 22 -m state --state NEW,ESTABLISHED -j DROP
```

Pe de altă parte, pentru a permite doar cererea de intrare către serverul SSH cu următorul interval de porturi:

```
iptables -A INPUT -p tcp -s 0/0 -d 192.168.0.1 --sport 500:1000 --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT
```

```
iptables -A OUTPUT -p tcp -s 192.168.0.1 -d 0/0 --sport 22 --dport 500:1000 -m state --state ESTABLISHED -j ACCEPT
```

Dacă este vizat tabelul NAT, se folosesc opțiunile --to-source și --to-destination. De exemplu interval de adrese IP se specifica astfel:

```
iptables -t nat -A POSTROUTING -j SNAT --to-source 192.168.0.100-192.168.0.200
```

Alternativ, pentru un interval de porturi se specifică astfel:

```
iptables -t nat -A POSTROUTING -j SNAT --to-source 192.168.0.100:1700-2000
```

Crearea unui server SSH

Pentru acest exemplu vom crea un server SSH în sistemul de operare Windows. Aplicația folosită este KpyM Telnet/SSH Server (<http://www.kpym.com>).

Pentru început se va descărca și se va instala aplicația. Aplicația va solicita în etapa de instalare dacă să se configureze ca server nesecurizat (23, Telnet) sau securizat (port 22, SSH). Vom alege opțiunea SSH.

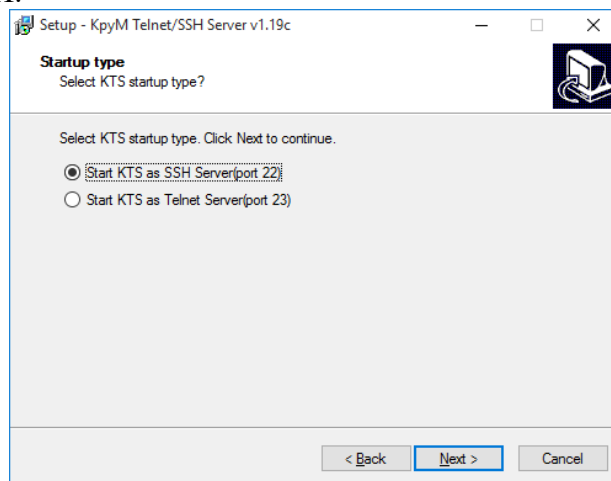


Figura 7 Alegerea tipului de server în aplicația KpyM Telnet/SSH Server

După instalare se va lansa "Setup KPyM Telnet SSH Server". Aceasta pornește o interfață de consolă prin care puteți naviga cu ajutorul tastaturii.

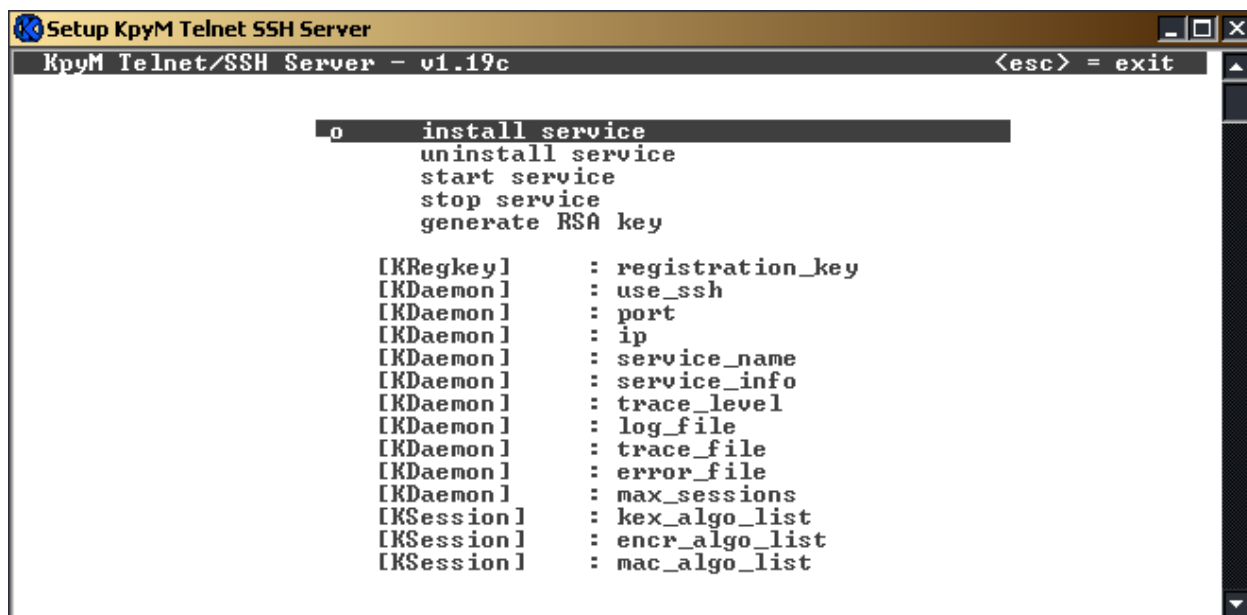


Figura 8 Interfața text a aplicației KPyM Telnet SSH Server

Deși sunt prezentate multe opțiuni, cele mai importante sunt:

- **use_ssh** care trebuie să fie 1 pentru SSH și 0 pentru Telnet
- **port** care trebuie să fie 22 pentru SSH și 23 pentru Telnet
- **Stop service / start service** după fiecare schimbare a configurației (SSH sau Telnet) trebuie oprit (stop service) apoi repornit (start service) serviciul pentru a fi aplicate configurările.

Se verifică **use_ssh** și **port** pentru a vedea ce serviciu este activ și se modifică dacă este nevoie.

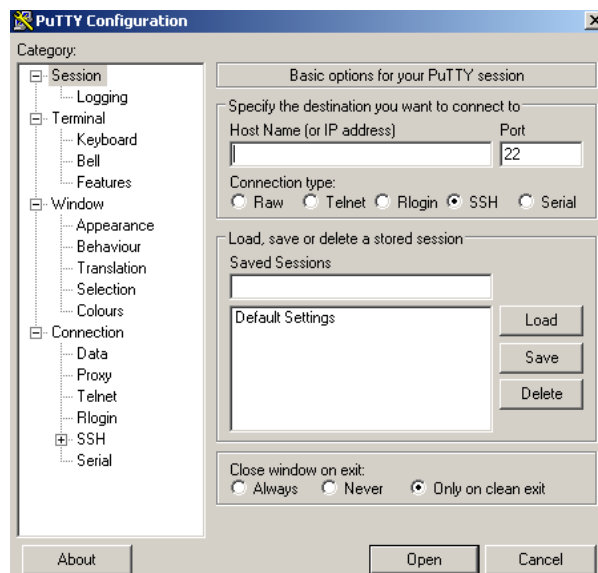


Figura 9 Interfața aplicației Putty pentru conectare la SSH

Observație: serverul funcționează doar cât timp aplicația de setup este pornită. Dacă închideți fereastra serverul se închide și el. Instalarea aplicației drept serviciu și pornirea ei

automata necesita repornirea sistemului de operare. Conectarea la server se face cu aplicația Putty.

Se introduce adresa IP și se alege SSH sau Telnet. Pentru SSH este necesar să aprobați cheile serverului SSH.

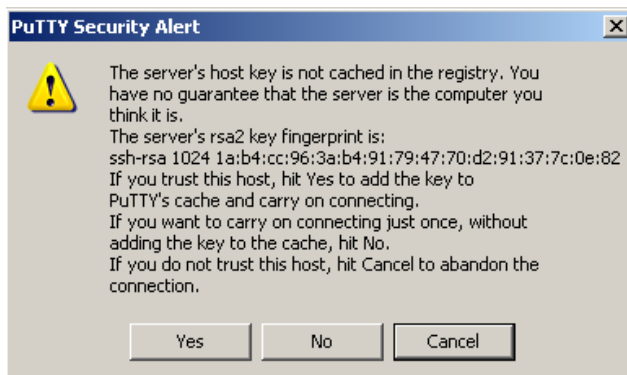


Figura 10 Pentru conectarea la un server SSH este necesară acceptarea cheii serverului (opțiunea Yes)

Crearea unui tunel SSH

Un tunel SSH este necesar uneori pentru a putea accesa o rețea din internet atunci când există un firewall între calculatorul utilizatorului și aceasta destinație care restricționează accesul direct. Un tunel SSH are nevoie de un alt calculator pe care rulează un server SSH, sistem care are acces nerestricționat către destinația pe care dorim să o accesăm.

Există două metode prin care se poate crea un tunel SSH:

- Port forwarding pentru porturile locale
- Port forwarding pentru porturile remote

Pentru exemplele următoare se va folosi aplicația ssh inclusă în sistemul de operare Linux. Alternativ se poate folosi aplicația Putty.

Port forwarding pentru porturile locale

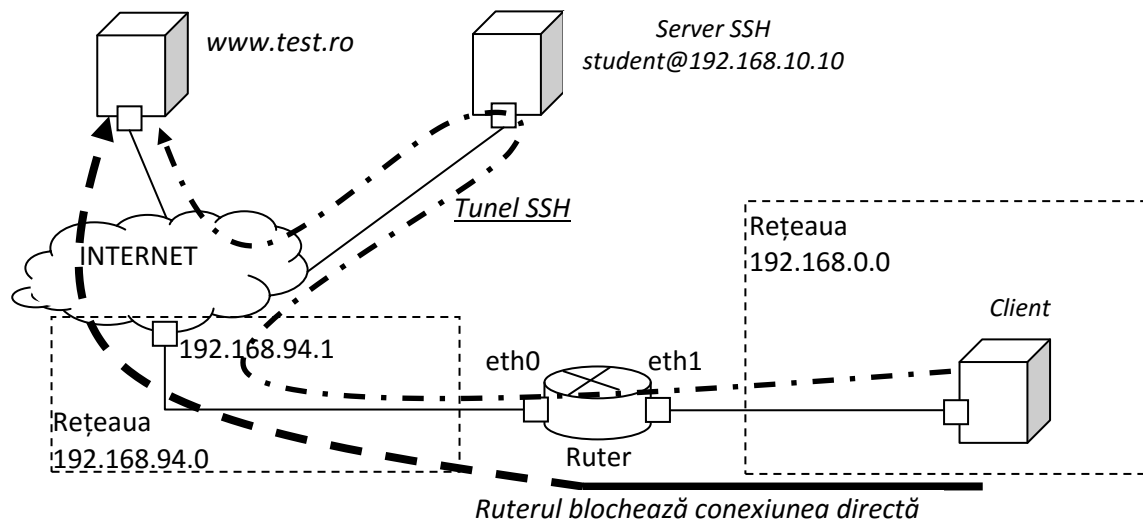


Figura 11 Port forwarding pentru porturile locale

Sa presupunem că putem accesa un server SSH cu IP-ul: 192.168.10.10 dar nu putem accesa site-ul www.upit.ro (portul 80). Comanda care permite crearea unui tunel prin serverul SSH către site-ul dorit este:

```
#ssh -L 8000: www.upit.ro:80 student@192.168.10.10
```

Parametrul `-L` specifică faptul că orice cerere trimisa pe portul local 8000 va fi trimisă se serverul SSH către destinație www.upit.ro pe portul 80.

Port forwarding pentru porturile remote

Aceasta funcție este utilă în cazul în care un utilizator de la distanta dorește sa acceseze un serviciu de pe calculatorul local și nu poate face acest lucru în mod direct (deoarece sistemul nostru se află într-o rețea privată în spatele unui ruter care nu are activat port forwarding pentru IP-ul calculatorului nostru).

Sa presupunem ca dezvoltam un site web pe calculatorul local și dorim ca un utilizator extern sa acceseze acest site în mod indirect prin intermediul calculatorului pe care rulează serverul SSH.

Comanda care permite crearea acestui tunel este:

```
#ssh -R 8000:localhost:80 student@192.168.10.10
```

Parametrul `-R` specifica faptul ca orice cerere primita pe portul 8000 al serverului SSH va fi trimisă către calculatorul nostru pe portul 80, prin tunelul SSH.

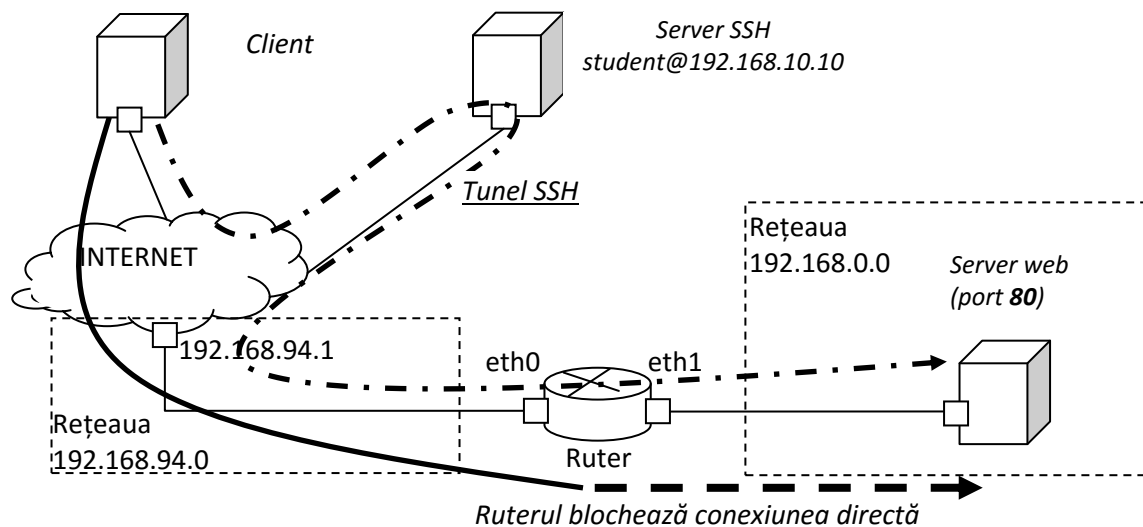


Figura 12 Port forwarding pentru porturile remote

Desfășurarea lucrării

1. Se va studia breviarul teoretic.

2. Se vor testa fluxurile TCP și UDP și comportamentul acestora pentru condiții precum modificarea latentei sau pierderea de pachete. Se va urmări evoluția traficului de rețea în tot acest timp.
3. Se va realiza port forwarding pentru fluxurile multimedia prezentate anterior, astfel încât acestea să poată fi accesate de către un calculator care nu este prezent în rețeaua locală ca serverului multimedia. Se va seta valoarea TTL=2 pentru a permite traversarea routerului.
4. Se vor configura parametrii fluxurilor video și audio care sunt transmise în rețea și se va observa impactul acestora asupra calității imaginilor dar și asupra traficului care este trimis.
5. Se va monitoriza traficul de date cu aplicația Wireshark și se va observa transferul informațiilor criptate.

Capitolul 7. Virtualizarea sistemelor de operare

Obiectivul lucrării

Virtualizarea este procesul de transformare a resurselor hardware în resurse software. Virtualizarea este prezentă în toate activitățile legate de funcționarea calculatoarelor și conectivitatea la internet. Laboratorul va prezenta noțiunile de bază privind instalarea și configurarea unui sistem de virtualizare pe un sistem de calcul și a unui sistem de operare pe această platformă.

Breviar teoretic

Virtualizarea se refera la crearea unei versiuni software a unui sistem hardware, în acest mod se creează posibilitatea rulării simultane a mai multor sisteme de operare pe un singur computer fiind posibilă o utilizare mai bună a resurselor sistemului de calcul decât în cazul în care ar exista un număr similar de mașini fizice. Virtualizarea crește productivitatea muncii, permițând instalarea automată a softului și a aplicațiilor pe mașinile virtuale, protejează datele critice și aplicațiile prin redundanța la nivel hardware, permite actualizări dinamice și managementul centralizat al serverelor, precum și reducerea la zero a timpului în care serverele sunt oprite pentru mentenanță.

Virtualizarea a apărut din anii '60 dar numai în prezent are suportul hardware și software pentru a deveni o tehnologie disponibilă pentru utilizatorii obișnuiți la un preț accesibil. În acest moment toate firmele de vârf din domeniu sunt implicate în implementarea virtualizării, atât la nivel hardware cât și la nivel software, fiind considerată tehnologie de vârf în acest moment în IT.

În mod ideal se dorește ca o singură aplicație sau serviciu să ruleze pe un server, deoarece mai multe aplicații rulând simultan pe același sistem crește riscul de blocare a acestuia în cazul în care una are probleme. Servicii uzuale care pot partaja aceeași mașină pot include, de exemplu: servicii pentru contabilitate, servicii web (server e-mail, server web, server de fișiere, etc.), servicii pentru securitate și monitorizare, pentru integrarea serviciilor de telefonie (Asterix PBX), etc. Intricată puterea sistemelor de calcul a crescut, această practică a dus la sub-utilizarea procesorului fiecărui server.

Apariția virtualizării a schimbat fundamental această situație. Astfel, aplicațiile rulează pe mașini virtuale (logice) separate, chiar dacă ele se află pe același hardware fizic. Adicional, chiar dacă hardware-ul pe care este instalat un server virtual încetează să funcționeze, virtualizarea

permite trecerea transparenta pentru utilizator și în scurt timp (sau chiar în timp real) a tuturor informațiilor pe un alt hardware, iar întreaga activitate a firmei nu este deloc afectată.

Scopul virtualizării este reducerea dependentei de resursele hardware, ușurarea sarcinilor de administrare și nu în ultimul rând reducerea costurilor. De exemplu, un sistem de calcul (desktop) obișnuit consuma zeci de wați și în starea idle. Dacă 10 astfel de calculatoare ar fi pornite și fiecare este utilizat în procent de 5%, ele ar avea un consum mult mai mare comparativ cu situația în care un singur calculator ar fi pornit și ar fi ocupat în procent de 50%. Suplimentar spațiul fizic necesar pentru stocarea serverelor va fi mare.

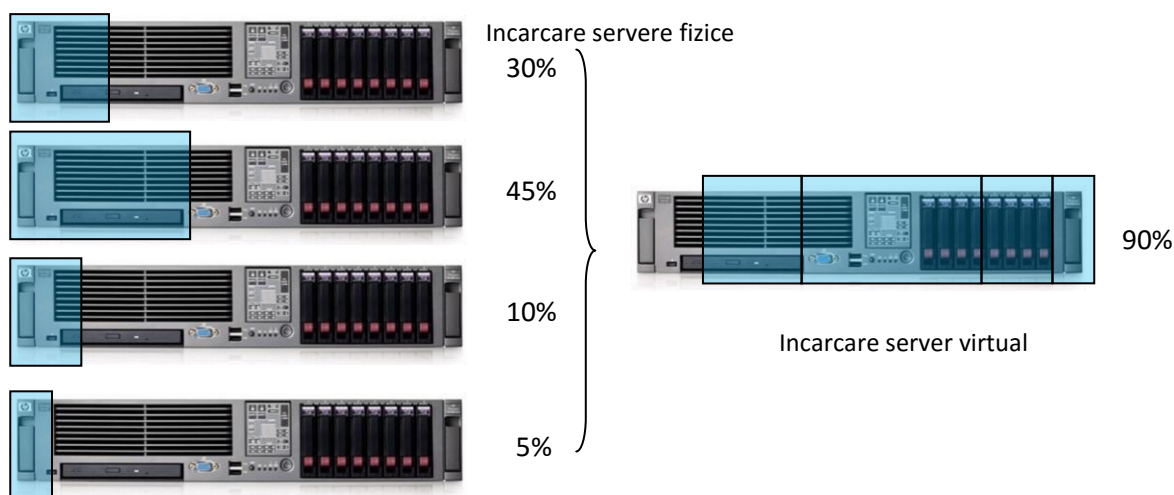


Figura 1 Un server cu mașini virtuale poate înlocui sistemele de operare ale mai multor mașini fizice, reducând spațiul de stocare și consumul de energie

Virtualizarea îmbunătățește eficiența dar și disponibilitatea resurselor și aplicațiilor dintr-o organizație, iar lucrul acesta se face în mod automat. În plus se salvează din costurile IT prin consolidarea gamei de resurse și livrarea de mașini cu disponibilitate ridicată.

Există mai multe tehnologii considerate suport pentru implementarea virtualizării: RAID care combina multe disk-uri într-unul singur; Virtualizarea stocării se referă la procesul complet de abstractizare a memoriei logice de memoria fizică și este folosită de obicei în sistemele Storage Area Network; Virtual Private Network (VPN) și Network Address Translation (NAT) sunt tehnologii de networking care creează rețele virtuale.

Virtualizarea software oferă o serie de avantaje majore, precum:

- Vor fi reduse semnificativ investițiile pentru achiziționarea de noi resurse hardware prin implementarea de servere virtuale pe mașinile deja existente. Numărul serverelor virtuale este de obicei fiind mai mare decât cel al mașinilor fizice.
- Indiferent de volumul de date administrate, operațiunile de backup sau restore sunt mult mai ușor de efectuat pe mașinile virtuale, asigurând planul de continuitate al unei afaceri prin soluții avansate de recuperare în cazul apariției erorilor.
- Folosind același nivel de consum, facturile de energie vor fi reduse, având astfel implicații majore la scăderea costurilor de administrare.

- Timpul de implementare finală pentru noile versiuni de software este redus drastic deoarece nu mai trebuie luate în calcul particularitățile hardware. Suplimentar, se pot testa mai ușor și mai sigur versiuni noi ale software-ului.

- Operațiunile de upgrade sau depanare sunt efectuate cu ușurință, administratori IT acționând de la distanță asupra terminalelor din rețea sau a serverelor din datacenter de la o simplă consolă.

De multe ori unificarea aplicațiilor care rulează pe mai multe servere pe un singur echipament fizic nu este posibilă în acest mod deoarece sistemele de operare trebuie să fie diferite datorită constrângerilor diferitelor aplicații (o aplicație necesită anumite versiuni ale sistemului de operare). În acest moment intervine virtualizarea, care permite rularea mai multor sisteme de operare diferite în interiorul mașinii virtuale.

Desigur și virtualizarea are dezavantaje, cum ar fi creșterea complexității sistemului și o performanță scăzută comparativ cu soluția hardware.

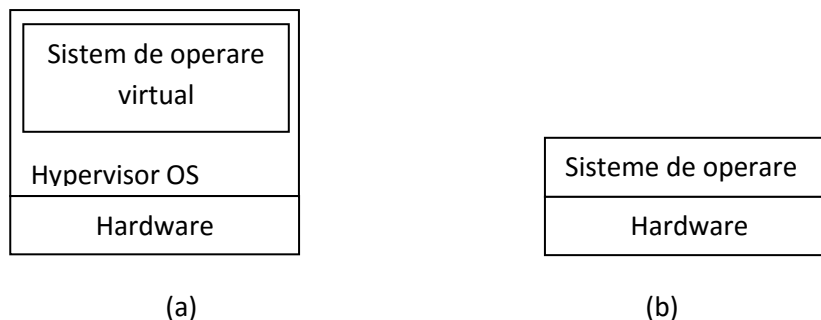


Figura 2 Comparare între un sistem de virtualizare (a) și un sistem tradițional (b)

Sistemele de operare virtuale sunt administrate de o aplicație numită hypervisor.

În continuare vom folosi noțiunile de:

- sistem de operare gazda pentru a identifica sistemul de operare pe care este instalată mașina virtuală. Acest sistem de operare are rolul să ofere aplicației hypervisor resursele hardware ale sistemului real (fizic). Sistemul este dependent de configurația specifică a hardware-ului pe care este instalat.

- sistem de operare invitat pentru a identifica sistemul de operare virtual care este administrat de hypervisor și care rulează în interiorul sistemului de operare gazdă. Sistemul este independent de hardware și poate fi deplasat și pornit pe oricare sistem gazdă cât timp resursele hardware virtuale îi sunt asigurate.

Aplicațiile hypervisor se pot clasifica în două grupuri:

- Hypervisor de tip 1: acest tip (numit și bare metal hypervisor) rulează ca o aplicație software pe un sistem de operare dedicat exclusiv administrării mașinilor virtuale. Sistemul de operare gazdă nu are interfața grafică cu utilizatorul ci interfața text, lucru care reduce numărul

de resurse necesare, însă crește configurarea instalării și configurării sistemului. Administrarea aplicației hypervisor se face fie în linia de comandă, fie la distanță, printr-o interfață grafică accesată intermediul rețelelor de calculatoare. Performanța mașinilor virtuale este foarte apropiată de performanța unui sistem de operare clasic. Exemple sunt: Microsoft Hyper-V, VMWare Esxi, Xen.

-Hypervisor de tip 2: acest tip rulează ca o aplicație software obișnuită într-un sistem de operare. De obicei aplicația instalează drivere pentru anumite componente (USB, adaptor de rețea, etc.) pentru a asigura accesarea și comunicarea cu dispozitivele periferice. În mod normal administrarea se face dintr-o interfață grafică, necesitând un sistem de operare care are interfața grafică cu utilizatorul. Acest lucru duce la ocuparea unui număr important de resurse de către sistemul de operare gazdă. Performanța mașinii virtuale în acest caz va fi semnificativ redusă comparativ cu un sistem de operare clasic însă sistemul de operare gazdă va putea fi folosit de utilizator în paralel cu mașina virtuală. Exemple de hypervisor de tip 2 sunt: VirtualBox, Microsoft Virtual PC, VMWare workstation.

În acest moment nu există o standardizare a aplicațiilor hypervisor, însă majoritatea dezvoltatorilor acestor aplicații au dezvoltat unelte software care permit trecerea de la o soluție de virtualizare la alta. Suplimentar, toți producătorii oferă aplicații pentru transformarea unui sistem real (fizic) într-o mașină virtuală.

Aplicații

În acest laborator va fi prezentată soluția de virtualizare de tip 1 Microsoft Hyper-V Server 2012 R2. Aceasta este o versiune Core a sistemului de operare Windows Server de la Microsoft. Versiunile Core sunt disponibile gratuit pentru descărcare și evaluare și pot implementa multiple roluri precum: server DNS, server Hyper-V pentru virtualizare, server de Fișiere, DHCP, etc.

Microsoft Hyper-V Server 2012 R2 poate fi administrat din linie de comandă, din acest motiv utilizarea uneltelor de control avansat și scripting din PowerShell este recomandată.

Powershell este o consolă inclusă de Microsoft în versiunile Windows mai noi de Windows 7 și disponibilă ca o descărcare separată pentru versiunile mai vechi de Windows. Aceasta consolă suportă un număr mai mare de comenzi decât consola Windows clasică precum și funcții speciale pentru rularea scripturilor. Pentru a porni PowerShell, într-o consolă obișnuită Windows se execută comanda *powershell*.

Pentru a obține ajutor în legătură cu folosirea comenzilor în PowerShell, se poate tasta: *get-help* urmat de comanda despre care se dorește obținerea de informații. De exemplu: *get-help network* afișează o listă de rezultate conținând acest cuvânt cheie.

În cazul în care o comandă nu poate fi executată deoarece trebuie lansată în modul Administrator, va fi afișat un mesaj de eroare. În cazul în care documentația de ajutor nu este instalată, se tastează comanda: Update-Help.

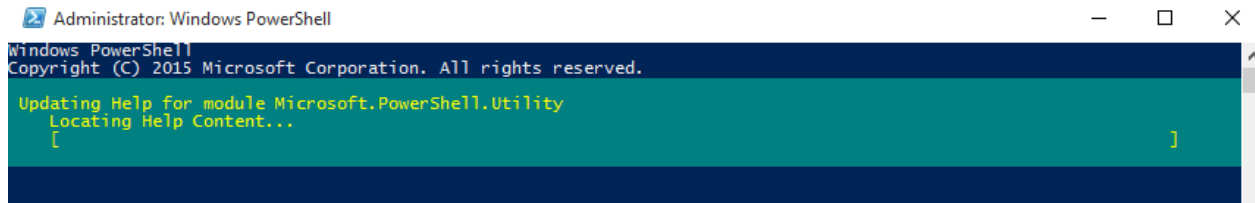


Figura 3 PowerShell pornit în modul Administrator indica acest lucru în titlul ferestrei

Comenzile din consola clasică Windows sunt suportate: dir, ls, cd, del, md, cls, echo.

Rezultatul acestor comenzi este nu numai un text afișat în consola ci și un obiect care poate fi procesat ulterior de alte comenzi. În acest fel mai multe comenzi pot fi înlănțuite pentru a executa comenzi complexe. Comenzile sunt separate prin caracterul |.

De exemplu:

```
dir | get-itemproperty | format-list | out-file info.txt
```

Comanda permite listarea conținutului unui director (dir), obținerea informațiilor despre obiectele găsite (get-itemproperty) și afișarea de informații adiționale (format-list). Rezultatele vor fi salvate într-un fisier (out-file) numit info.txt

Instalarea și configurarea unui sistem de virtualizare de tip 1: Microsoft Hyper-V

Compania Microsoft a intrat în domeniul virtualizării de mulți ani și a propus soluții de virtualizare atât de tipul 1 (Hyper-V) cât și de tipul 2 (Virtual PC). În ultimii ani însă au întrerupt dezvoltarea Virtual PC (ultima versiune a apărut în 2011) și au dezvoltat exclusiv Hyper-V pe care o oferă atât ca un sistem de operare gratuit, numit Hyper-V Server, cât și în soluțiile comerciale de Windows Server.

În exemplul următor este prezentat sistemul de operare gratuit: Hyper-V Server. Acest sistem de operare nu include interfața grafică. După instalare, utilizatorul poate accesa două ferestre:

- O fereastră cu un meniu text care permite configurarea ușoară a setarilor principale ale sistemului de operare (data, setari de limba, adresa IP, workgroup, actualizari software, repornire sistem, etc.)
- O fereastră consola în care se pot executa comenzi obisnuite sau PowerShell pentru a administra în profunzime sistemul.

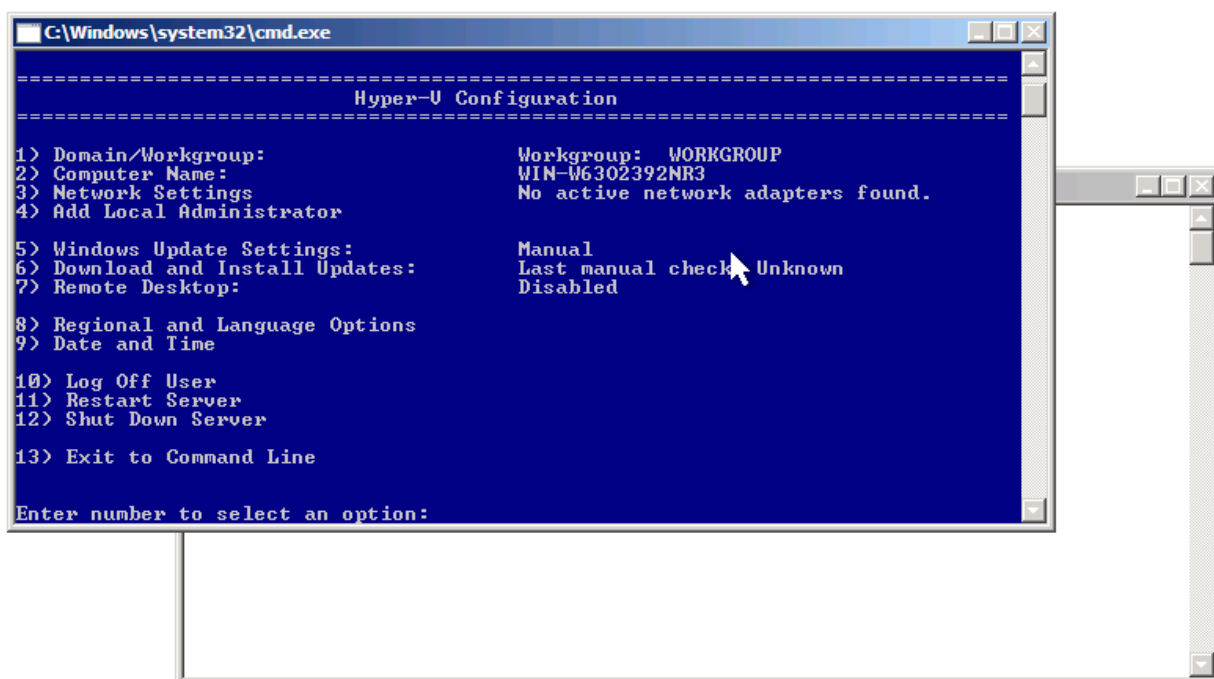


Figura 4 Sistemul de operare Hyper-V Server nu prezinta interfata grafica obisnuita cu ferestre a sistemelor Windows

Administrarea acestui sistem are nevoie de un calculator pe care îl vom numi *Client*, pe care rulează cel puțin sistemul de operare Windows 8.

Pașii necesari pentru instalarea sistemului de operare Hyper-V Server

În majoritatea cazurilor este recomandată instalarea unei noi versiuni a sistemului de operare și să se evite opțiunea de upgrade a unei versiuni mai vechi a sistemului de operare. Motivul este că procedura de upgrade poate să nu ia în calcul personalizările sistemului de operare rezultând un sistem de operare instabil sau să lase funcționale servicii care nu mai sunt folosite de sistemul nou de operare rezultând un sistem de operare încărcat cu software redundant chiar de la momentul instalării.

În continuare vom prezenta situația instalării ca un nou sistem de operare.

La instalare se va selecta partiția pe care o va folosi sistemul de operare (se recomandă instalarea pe un disk SSD) și se va cere stabilirea parolei pentru utilizatorul Administrator. Este important de reținut această parolă pentru conectarea de la distanță de pe calculatorul *Client*.

Observație: În secțiunea următoare, de fiecare dată când modificările necesită/solicita o repornire se va face acest lucru!

După ce sistemul s-a instalat se vor face setările din consola:

1. configurare adresa IP manual sau automat (DHCP) și configurarea server DHCP. Se recomandă o setare manuală a adresei IP sau alocarea de către serverul DHCP a unei anumite adrese IP pentru acest sistem deoarece este necesar să știm IP-ul la care trebuie să ne conectăm.
2. configurarea unui administrator local: acesta cont trebuie să aibă același nume și parolă ca și sistemul de la distanță de la care se va face ulterior administrarea;
3. activare Remote Desktop (dacă se dorește conectarea într-o rețea care nu are *Active Directory* configurat se va alege opțiunea 2 (*less secure*);

4. conectarea la un domeniu dacă acesta există;
5. configurare remote management: se selectează opțiunile „Allow MMC Remote Management” și „Allow Windows PowerShell”;
6. stabilire nume server. De exemplu DEV;
7. În cazul în care calculatorul nu face parte dintr-un Active Directory trebuie executate următoarele comenzi în PowerShell:

Enable-NetFirewallRule -DisplayGroup “Windows Remote Management”

Enable-NetFirewallRule -DisplayGroup “Remote Event Log Management”

Enable-NetFirewallRule -DisplayGroup “Remote Volume Management”

Set-Service VDS -StartupType Automatic

În acest moment serverul este configurat și se poate trece la configurarea clientului.

Configurare sistem Client pentru controlul de la distanta al Hyper-V Server

Pe sistemul Client se va instala din Control Panel -> Programs and Features -> Turn Windows features on and off, functia de Hyper-V. Aceast feature va instala pe Client uneltele de administrare la distanta pentru sistemul Hyper-V Server.

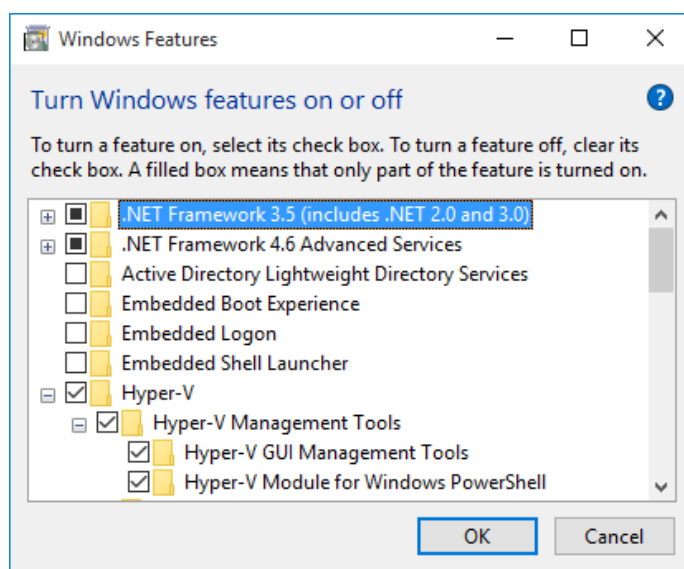


Figura 5 Unelele de administrare la distanta Hyper-V din sectiunea Windows Features

Mai este necesar ca pe calculatorul client de la care se va face administrarea:

- să se adauge o regulă la firewall (în PowerShell pornit în modul Administrator):
Enable-NetFirewallRule -DisplayGroup “Remote Volume Management”
- să se modifice fișierul *hosts* (găsit în folderul c:\windows\system32\drivers\etc dacă Windows a fost instalat în calea implicită) astfel încât să asocieze numele serverului

(ales în acest exemplu anterior drept DEV) cu adresa IP acestuia, de exemplu o linie noua cu:

192.168.0.2 DEV

Clientul este în acest moment configurat și se pot porni unelte de administrare la distanță a serverului. Se va realiza conectarea la adresa serverului Hyper-V.

În exemplul de mai jos cele două sisteme sunt în aceeași rețea locală, conectate printr-un switch, pentru a preveni problemele de filtrare a traficului dacă s-ar conecta printr-un ruter.

Nu uitați să folosiți același nume și parola pe sistemul Client la momentul conectării ca și cele configurate de pe serverul Hyper-V. În cazul în care una dintre cele două diferă, managerul de rețea nu se va putea conecta la sistemul de virtualizare.

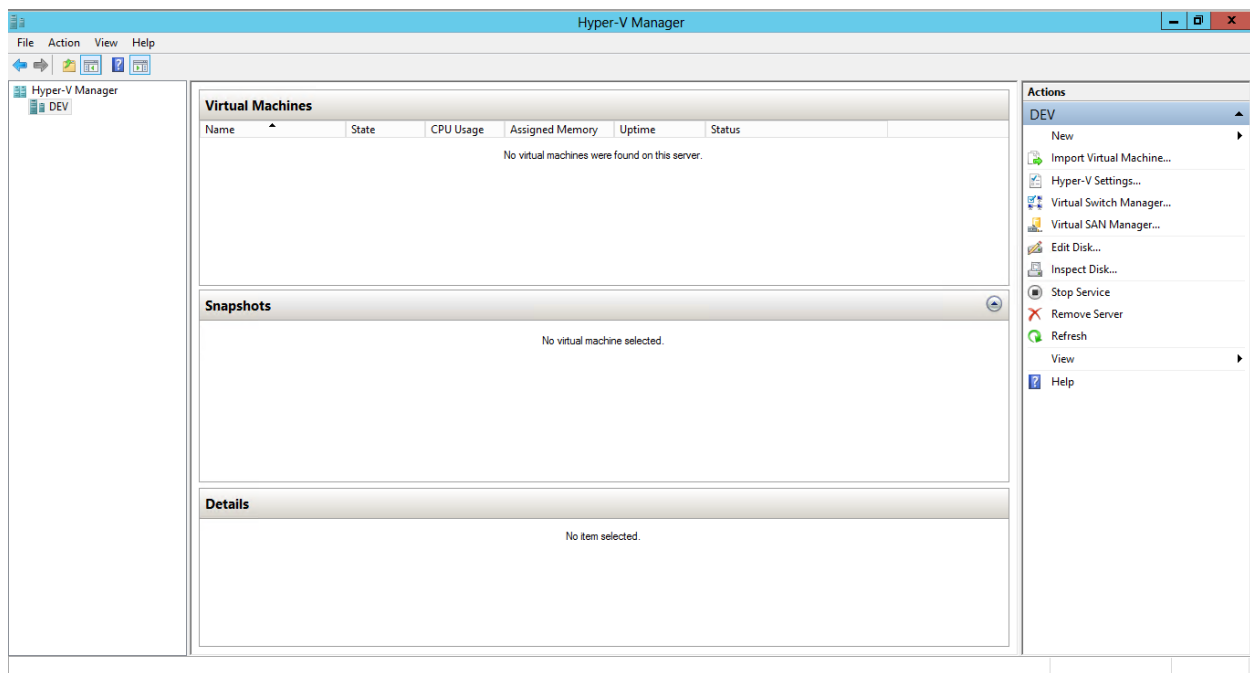


Figura 6 Interfața aplicației Hyper-V Manager

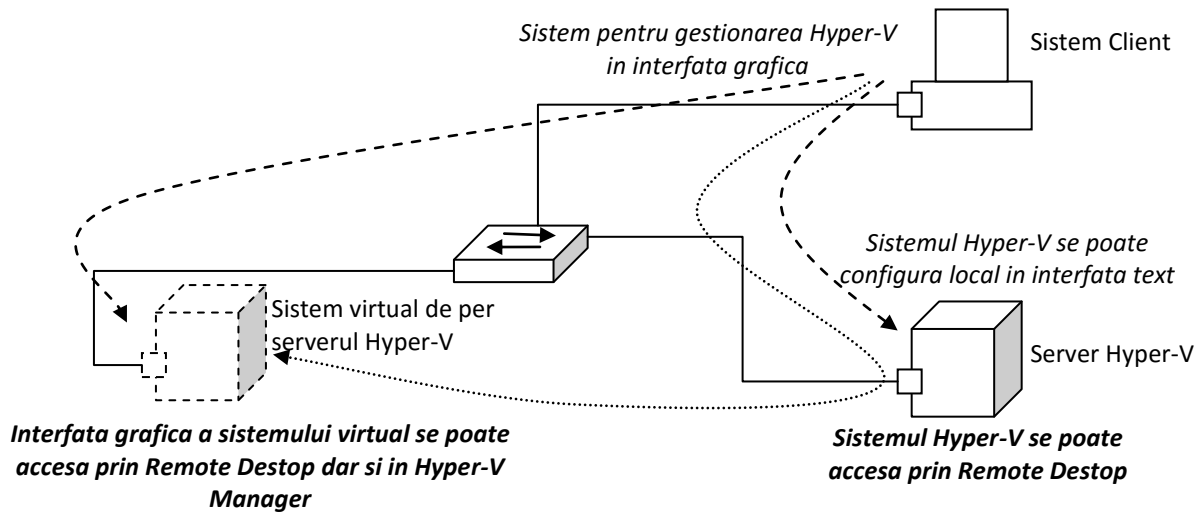


Figura 7 Administrarea serverului Hyper-V și a mașinii virtuale: este posibilă conectarea directă prin Remote Desktop sau prin Hyper-V Manager

Crearea unei mașini virtuale

Pentru a crea o mașină virtuală nouă alegem New -> Virtual Machine.

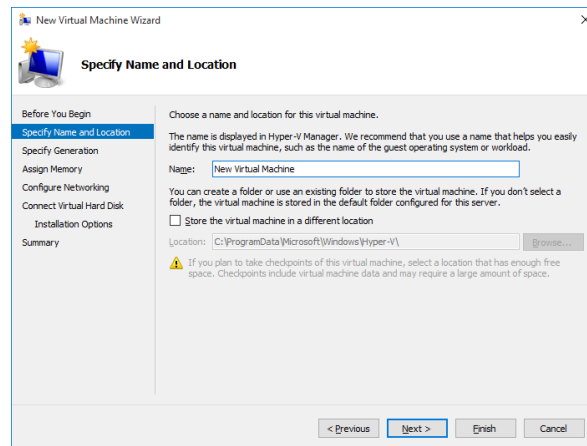


Figura 8 Selectarea numelui și a locației mașinii virtuale.

În mod implicit va fi aleasă locația sistemului de operare. În cazul în care se dorește o altă locație, se va selecta acest lucru în acest pas al instalării sau la sfârșit în momentul în care se pot revizui opțiunile selectate. În continuare se va selecta versiunea mașinii virtuale dacă, din motive de compatibilitate, dorim să creăm o mașină virtuală compatibilă cu versiunile anterioare de Hyper-V.

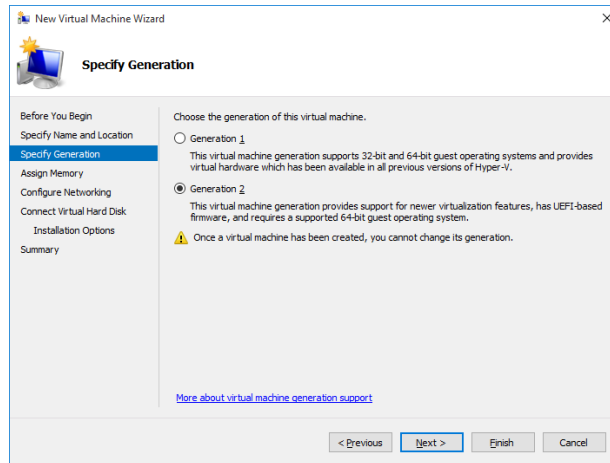


Figura 9 Alegerea versiunii mașinii virtuale Hyper-V

Alegerea dimensiunii memoriei este un pas important în configurarea unei mașini virtuale. În cazul alocării unei memorii insuficiente (în special pentru serverele de baze de date), performanta sistemului va avea de suferit.

Memoria poate fi alocata static sau dinamic. Memoria alocata static este disponibila tot timpul pentru mașina virtuala, în vreme ce alocarea dinamica permite redistribuirea memoriei către alte mașini virtuale în cazul în care sistemul de operare virtual nu are nevoie de aceasta memorie.

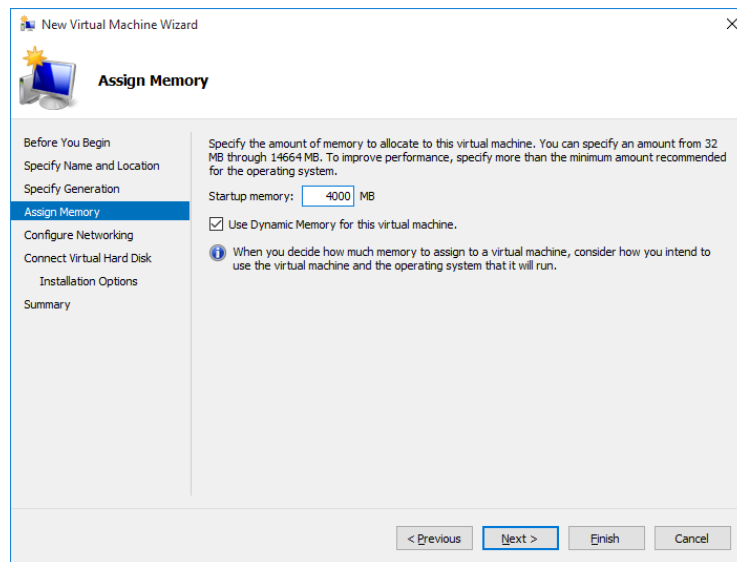


Figura 10 Selectarea memoriei alocate

Un sistem de operare virtual este un sistem software căruia ii trebuie alocata o placa fizica de rețea prin care sa poată trimite și primi pachete din mediul de transmisie, numit „*Virtual Switch*”.

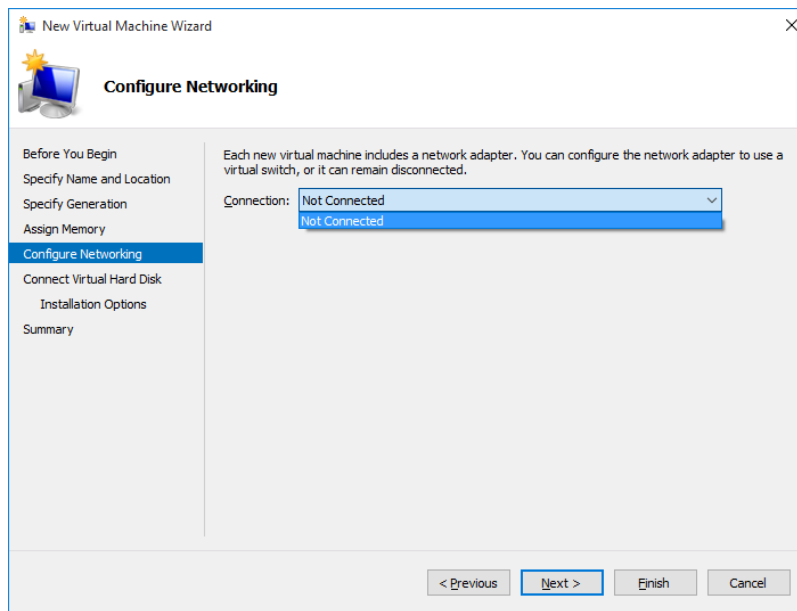


Figura 11 Interfața pentru selectarea unei conexiuni la rețea nu are implicit un Virtual Switch creat

În acest pas al configurării deoarece nu este creata nici o configurație pentru rețea în Hyper-V, în lista nu apar opțiuni pentru sistemul de operare. Configurațiile de rețea se vor specifica din meniul Virtual Switch Manager.

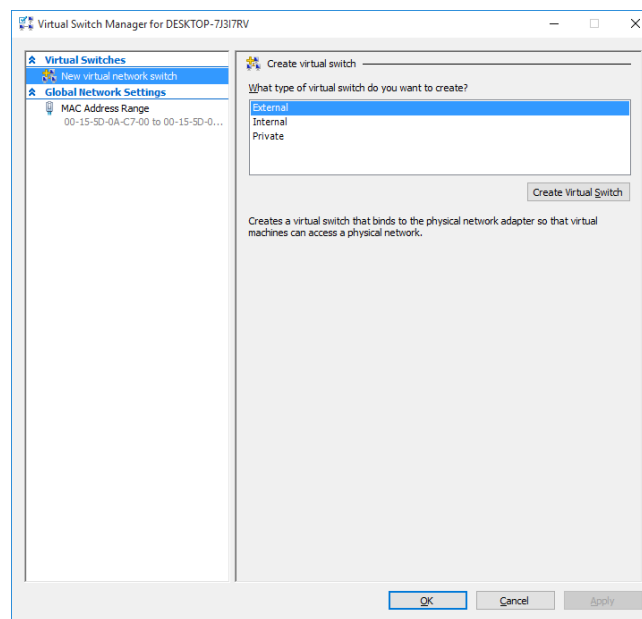


Figura 12 Opțiunile pentru crearea unui Virtual Switch

Exista trei opțiuni posibile pentru crearea unui Switch virtual:

- Pentru a crea o rețea care să se conecteze la internet prin intermediul plăcii de rețea a calculatorului fizic, se alege opțiunea External. Trebuie specificata placa

de rețea pe care vor fi trimise pachetele în cazul în care sistemul dispune de mai multe plăci de rețea.

- Opțiunea Internal creează o rețea care poate fi folosită doar între mașinile virtuale și mașina fizică.
- A treia opțiune, rețeaua Internal permite comunicarea numai între mașinile virtuale. Aceasta rețea poate fi folosită în teste în care conectarea la mașini reale sau la internet poate afecta securitatea sistemului.

În cazul în care sistemul face parte dintr-un VLAN, se poate selecta în acest moment identificatorul de VLAN care va fi folosit.

Odată creat switch-ul virtual, el poate fi modificat ulterior din același Virtual Switch Manager. Pentru exemplul acesta am ales opțiunea de conectare External. Dacă revenim la configurarea sistemului de operare, în cazul în care switchul virtual a fost creat, îl vom regăsi în lista de opțiuni.

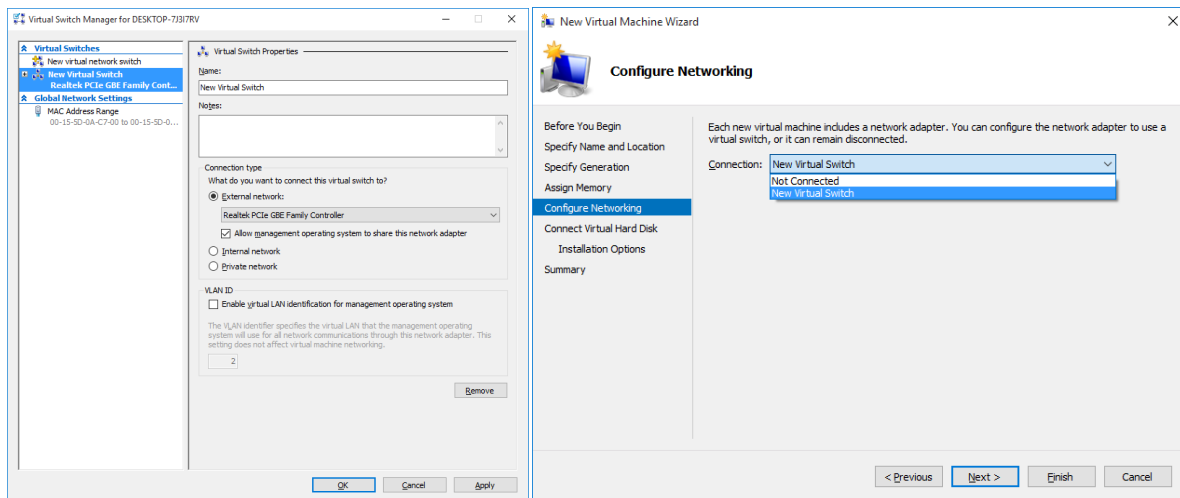


Figura 13 Odată creat un Virtual Switch, el apare în lista de selecție pentru mașina virtuală

Următorul pas este specificarea hard-disk-ului care va fi folosit. Opțiunile permit crearea unui hard-disk în acest moment, folosirea unui disk virtual creat anterior (de exemplu de la o mașina virtuală mai veche) sau adăugarea ulterioară a unei unități de stocare.

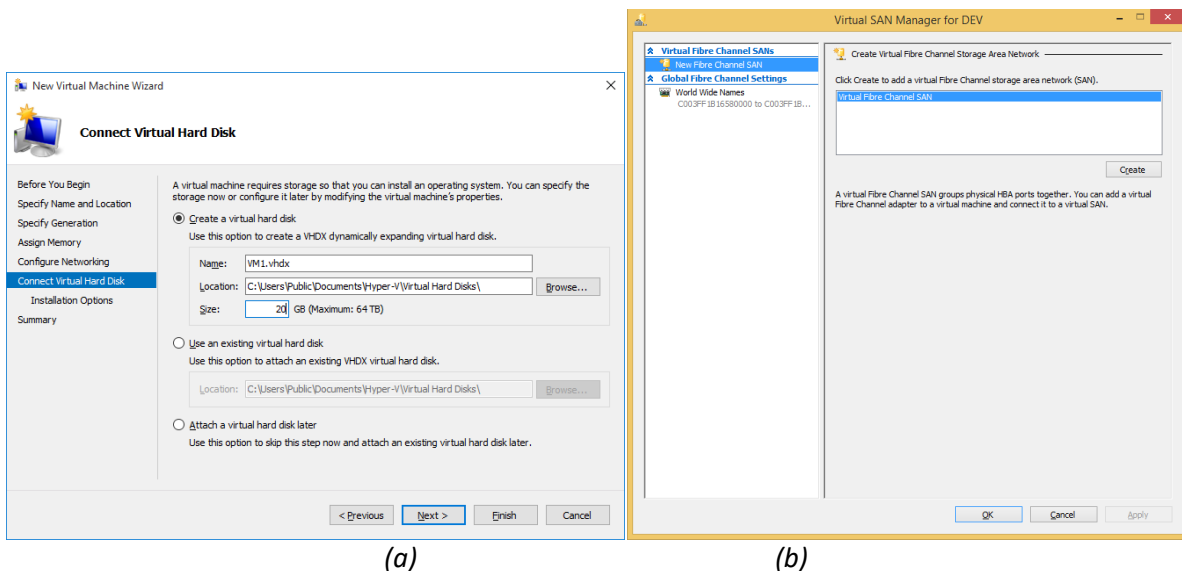


Figura 14 Selectare locație hard-disk pentru mașina virtuală (a). Este posibilă configurarea și selectarea unei locații de stocare în rețea prin Virtual SAN Manager (b)

În ultimul pas al configurării se va preciza locația CD/DVD de instalare a sistemului de operare, sau se poate preciza acest lucru ulterior.

Se recomanda ca în acest pas sa fie precizata locația și să se înceapă instalarea, deoarece unii utilizatori au semnalat apariția de probleme în cazul în care specificarea se face la un moment de timp ulterior.

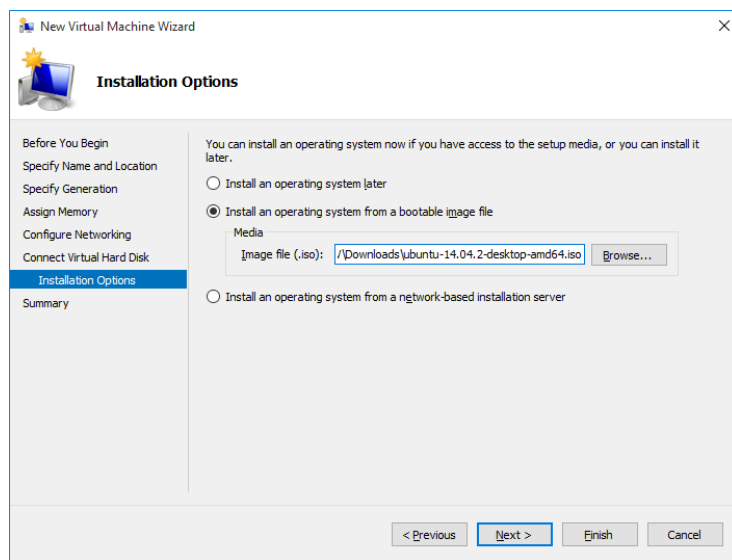


Figura 15 Selectare locație imagine ISO necesara pentru instalarea sistemului de operare (in acest caz Ubuntu Linux)

În final configurația sistemului poate fi revizuita într-o fereastră unică de configurare. Este indicat sa parcurgem fiecare selecție pentru ca sunt oferite setări avansate (adăugarea mai multor hard-disk-uri la un sistem sau eliminarea unor periferice care nu sunt necesare pentru mașina virtuală – de exemplu floppy disk).

Este important de știut ca anumite setări (ex. numărul de procesoare, hard-disk, etc.) nu pot fi modificate cât timp mașina virtuala este pornita și necesita pornirea acesteia pentru a putea fi aplicate.

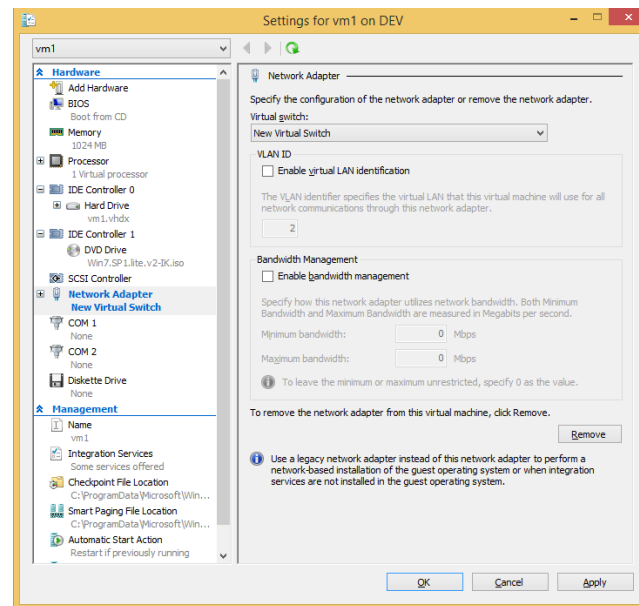


Figura 16 Setările mașinii virtuale pot fi vizualizate și modificate

În acest moment mașina este instalata pe sistemul de virtualizare și ne putem conecta la ea fie din managerul pentru server cât și direct prin rețea.

Instalarea și configurarea unui sistem de virtualizare de tip 2: VirtualBox

Exista o gamă largă de produse de virtualizare, de la aplicații software pentru virtualizarea serverelor și desktop-urilor, până la soluții complexe pentru firme, cum ar fi platformele automatizate de virtualizare pentru optimizarea centrelor de date și a infrastructurii IT.

Soluțiile de virtualizare a serverelor și optimizează infrastructura și ajută organizațiile să exploateze la maxim resursele hardware existente/achiziționate cât și să reducă considerabil costurile de investiție (serve, de stocare, de retelistică) și de operare (administrare, alimentare cu energie electrică, răcire), simplificând totodată managementul prin automatizarea proceselor IT și asigurarea unui maxim de disponibilitate, performată și scalabilitate.

Cu ajutorul virtualizării putem rula pe același sistem mașini virtuale multiple, fiecare având alta structură și alt sistem de operare, independent de celelalte mașini virtuale.

Anumite solutii de virtualizare sunt optimizate pentru anumite sisteme de operare (de obicei Windows sau Linux) sau pentru anumite platforme hardware (PC), altele pot emula inclusiv arhitecturi pentru echipamente mobile (arhitectura ARM). Pentru arhitectura ARM mai pot fi precizate Andy the Android Emulator, Genymotion, ambele bazate pe VirtualBox, Android emulator din Google SDK pentru sisteme Android, etc.



(a)

(b)

Figura 17 (a) Windows executat pe o platforma Linux; (b) Linux executat pe o platforma Windows.

VirtualBox a fost creată de Innotek GmbH și a devenit software gratuit și open-source în 2007. În 2008 Innotek a fost achiziționat de firma Sun Microsystems iar în 2010, deoarece Oracle a achiziționat Sun Microsystems, VirtualBox a devenit Oracle VM VirtualBox.

VirtualBox este una dintre cele mai folosite aplicații de virtualizare, datorită simplității de instalare și utilizare.

VirtualBox are și suport pentru interacțiunea cu aplicațiile scrise de dezvoltatorii de software prin VirtualBox API.

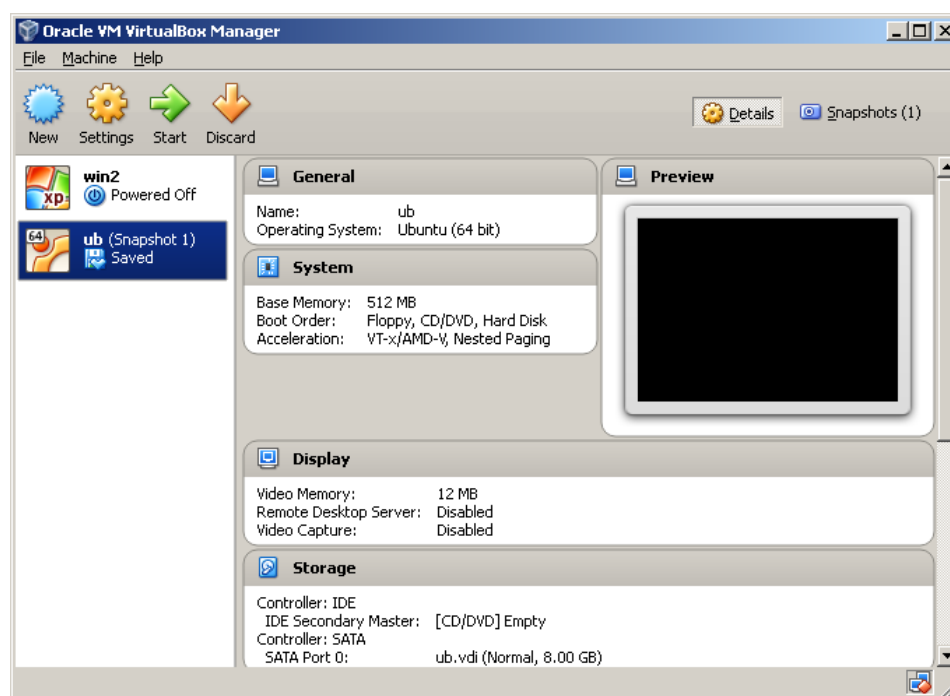


Figura 18 Interfața VirtualBox cu doua sisteme virtuale instalate.

VirtualBox poate fi folosit la testarea software dar și pentru a rula aplicații incompatibile cu sistemul de operare folosit în mod curent.

Pentru a verifica un software sau un sistem de operare, îl vom instala într-un sistem de operare virtual și a urmări dacă funcționează cum trebuie. În plus, este posibil să folosim mai multe sisteme virtuale în același timp, dacă resursele hardware sunt suficiente, și să le interconectăm în rețele virtuale, lucru care extinde posibilitățile de testare.

VirtualBox rulează ca o aplicație obișnuită în sistemul de operare și nu necesită dedicarea mașinii pentru virtualizare. Din acest motiv aplicația se recomandă să fie folosită pentru execuții ocazionale, de exemplu pentru testarea aplicațiilor, nu pentru sistemele de operare de producție care vor funcționa non-stop.

Aplicația are suport pentru salvarea stării mașinii virtuale. Acesta permite întreruperea unui test și reluarea acestuia la un moment de timp ulterior fără ca sistemul de operare virtual să perceapă această întrerupere.

În VirtualBox se pot instala multe sisteme de operare (Linux, Windows, Solaris, Max OS X și sisteme de operare mai vechi precum DOS sau OS/2) iar numărul acestora crește de la o versiune la alta, fiind actualizat cu noile sisteme de operare apărute.

Instalare sistem de operare Linux în VirtualBox

Instalarea sistemului de operare este un proces ghidat, în care trebuie să alegem dimensiunea memoriei alocate și spațiul de stocare alocat.

Procesul este similar cu cel de instalare a unui sistem de operare pe un sistem de virtualizare Hyper-V, cu următoarele excepții:

- aplicația de virtualizare este doar o altă aplicație în sistemul de operare Windows, în vreme ce soluția Hyper-V este una care rulează pe un sistem de operare dedicat virtualizării. Dacă se dorește aproximarea cât mai bună a performanței unui sistem real, soluția Hyper-V este soluția recomandată. Dacă se dorește obținerea unui sistem de operare în cel mai scurt timp posibil și în cel mai facil mod, VirtualBox este cel recomandat.

- Spre deosebire de Hyper-V, VirtualBox suportă mai multe tipuri de hard-disk, specifice altor soluții de virtualizare:

- (a) VDI: este un format specific VirtualBox (VirtualBox Disk Image);
- (b) VMDK: este un format deschis folosit în special de sistemele de virtualizare ale VMWare. Un disc virtual se include mai multe fișiere VMDK.
- (c) VHD: este formatul utilizat de Microsoft pentru soluțiile de virtualizare (actualizat la VHDX pentru Hyper-V). Permite pornirea în VirtualBox a acestor mașini virtuale, însă pentru formatul nou VHDX este necesară transformarea discului virtual cu ajutorul utilitarului `VBoxManage` după care fișierul VDI rezultat poate fi utilizat:

```
VBoxManage clonehd <filename>.vhdx <filename>.vdi --format VDI
```

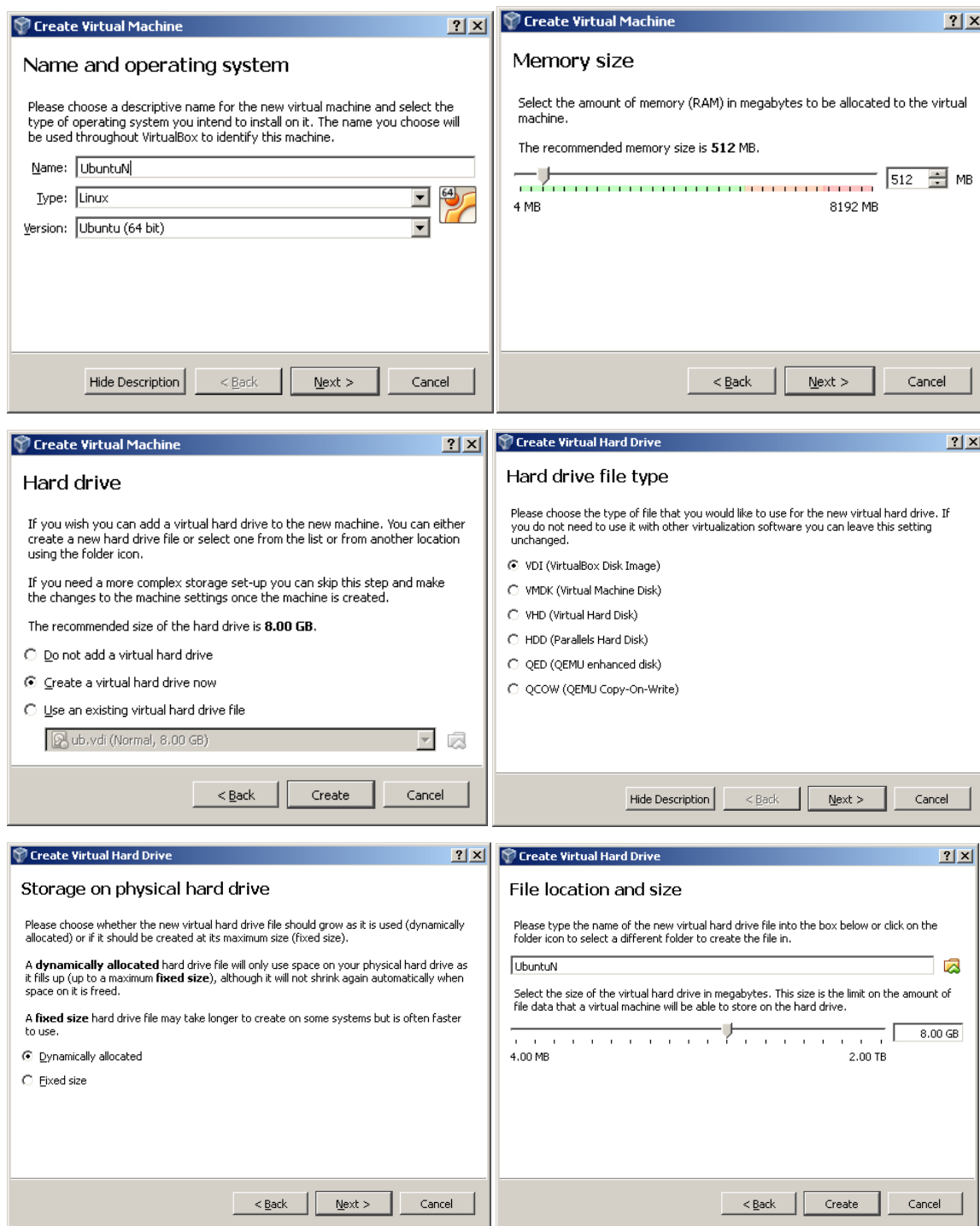


Figura 19 Pași de instalare a sistemului de operare Linux în VirtualBox

- la repornirea sistemului de operare Hyper-V Server, toate masinile virtuale active sunt repornite automat. În cazul VirtualBox, este necesara scrierea unui script care sa faca acest lucru.

În final se apăsă butonul de start pentru a porni mașina virtuală. VirtualBox afișează un meniu în care putem alege fișierul .iso sau drive-ul care conține imaginea sistemului de operare Ubuntu Linux descărcată de pe internet.

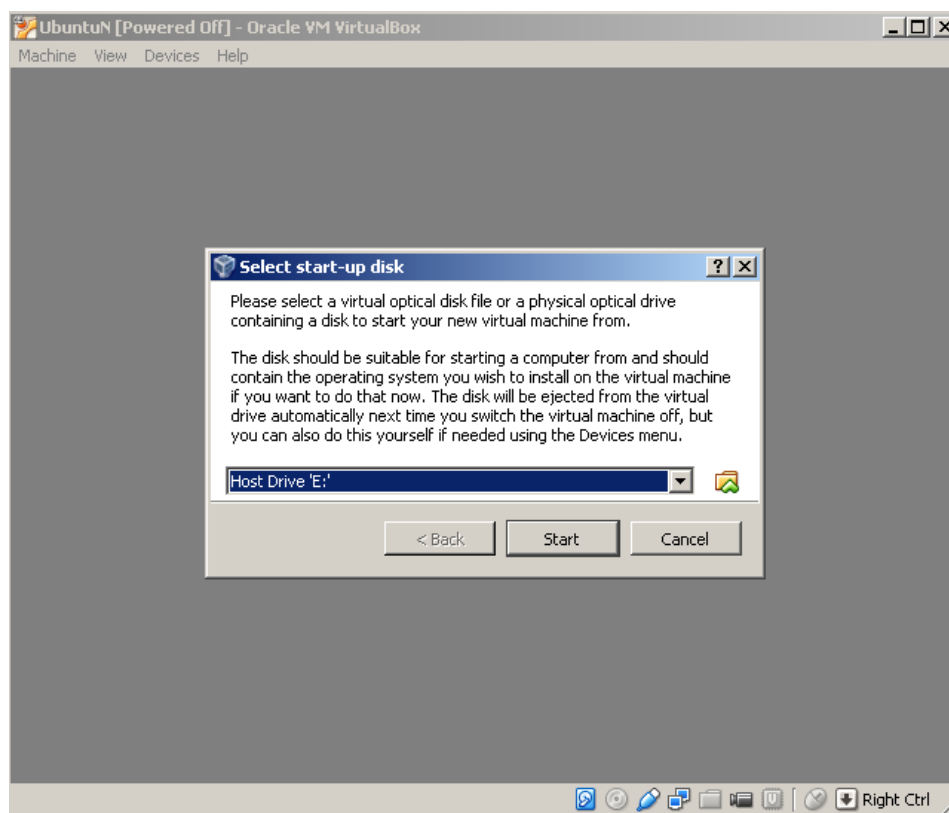


Figura 20 VirtualBox solicită precizarea locației sau a discului de unde poate încărca kit-ul de instalare al sistemului de operare

Desfășurarea lucrării

1. Se va studia breviarul teoretic.
2. Se va crea un sistem de virtualizare folosind Hyper-V Server.
3. Se va crea o mașină virtuală Ubuntu Linux pe sistemul de virtualizare din pasul anterior. Se va monitoriza ulterior activitatea acestei mașini virtuale.
4. Se va instala aplicația Oracle VirtualBox pe un sistem Windows și se va instala în VirtualBox un sistem de operare Ubuntu Linux. Se va monitoriza ulterior activitatea acestei mașini virtuale și încărcarea sistemului de calcul pe care se execută atât sistemul de operare virtual cât și cel fizic.

Capitolul 8. Virtualizarea echipamentelor de rețea

Obiectivul lucrării

Lucrarea si propune sa introducă tipurile de conectivitate la rețea disponibile pentru un sistem de operare virtual. Fiecare dintre opțiuni are avantaje și dezavantaje și se pretează pentru o anumita utilizare a sistemului de operare virtual. Aplicațiile vor urmări instalarea unui sistem de operare Linux pentru aceste tipuri de conectivitate la rețea iar sistemul de virtualizare utilizat este Oracle VM VirtualBox.

Breviar teoretic

Configurarea conectării la rețea a sistemului de operare virtual

Deși virtualizarea rețelelor oferă libertate în configurarea sistemului, numai anumite plăci de rețea care pot fi emulate. Acestea au fost însă alese pentru o compatibilitate maxima cu cât mai multe sistem de operare. De exemplu VirtualBox poate virtualiza nu mai puțin de 6 tipuri de hardware pentru adaptor de rețea: AMD PCNet PCI II (Am79C970A); AMD PCNet FAST III (Am79C973, the default); Intel PRO/1000 MT Desktop (82540EM); Intel PRO/1000 T Server (82543GC); Intel PRO/1000 MT Server (82545EM); adaptor de rețea paravirtualizat.

Oricare dintre acestea poate sa coexiste pe aceeasi mașina celelalte tipuri de plăci de rețea, fiecare având avantaje cum ar fi compatibilitatea sau performanta maxima. PCNet FAST III este suportat de aproape toate sistemele de operare și de catre managerul de boot Linux GNU GRUB bootmanager. Intel PRO/1000 MT de tip Desktop lucreaza cu Windows Vista și versiunile ulterioare și este recunoscuta de catre clientii Windows XP fără a instala alte drivere.

Adaptoarele de rețea pot și configurate într-unul din modele următoare:

NOT ATTACHED (adaptorul nu este atașat la rețea)

În acest mod, sistemul de virtualizare raportează sistemului de operare client prezenta cardului de rețea, dar ca nu există conexiune - ca și când cablul de rețea nu este introdus în adaptor. În acest fel este posibil sa informam sistemul de operare client ca nu este vizibila nici o conexiune.

NETWORK ADDRESS TRANSLATION (NAT)

Network Address Translation (NAT) este cea mai simpla cale de a accesa o rețea externa dintr-o mașina virtuala, fiind modul de lucru implicit în sistemele de virtualizare. În mod normal, nu necesita o alta configurare între rețeaua gazda și sistemul gazda.

O mașina virtuala cu NAT acționează asemeni unui calculator real care se conectează la Internet printr-un ruter. "Ruterul virtual" creează traficul de la și pana la mașina virtuala în mod transparent. Acest ruter este plasat între fiecare mașina virtuala și gazda. Aceasta separare

maximizează securitatea sistemelor virtuale. Mașina virtuala primește adresa de rețea și configurarea de la un server DHCP integrat în mașina virtuala. Adresa IP astfel atribuită mașinii virtuale este, într-o rețea diferită de cea a gazdei.

Dezavantajul modului NAT, asemenea unei rețele private din spatele unui ruter, este ca mașina virtuala nu este vizibilă în internet iar un server nu poate fi accesat din exteriorul rețelei în acest mod decât dacă este configurat port forwarding.

În acest mod cadrele de rețea trimise de către sistemul de operare client sunt primite de către software-ul mașinii virtuale, din care sunt extrase datele TCP/IP și retrimise sistemului de operare gazda. Mașina virtuala asculta replicile pachetelor trimise, le reîmpachetează și retrimite mașinii client care este rețeaua privată.

BRIDGED NETWORKING

Acesta este un mod avansat de lucru în rețea. Când este activat, mașina virtuala se conectează la unul din adaptoarele de rețea instalate și cere pachete din rețea, evitând stiva sistemului de operare gazda.

Cu bridged networking, mașina virtuala folosește un driver de dispozitiv pe sistemul gazdă care filtrează datele de la adaptorul de rețea fizică. Acest driver este numit driver "filtru net" și permite ca mașina virtuala să intercepteze date din rețeaua fizică și să injecteze date în aceasta. Atunci când un sistem de operare oaspete folosește o astfel de interfață software, sistemul gazda va vedea conexiunea cu acesta ca și când ar fi o legătură fizică. Acest lucru înseamnă că se poate seta rutarea între oaspete și restul rețelei.

INTERNAL NETWORKING

Acest mod de lucru este similar cu cel bridged cu excepția faptului că mașina virtuala poate comunica direct doar la alte mașini virtuale pe aceeași gazda care sunt conectate la aceeași rețea internă.

Chiar dacă tehnic tot ce se poate face folosind rețeaua internă poate fi făcut folosind rețeaua bridge, există avantaje de securitate în cadrul folosirii rețelei interne. Cu modul rețea bridge tot traficul trece prin interfața fizică a sistemului gazda, prin urmare este posibilă atașarea unui software de ascultare (precum Wireshark) la interfața gazda și înregistrarea traficului care trece prin ea. Dacă se prefera ca mașinile virtuale să comunice invizibil pentru rețeaua reală, ascunzând date atât față de sistemul gazda cât și de oaspete, trebuie aleasă conectivitatea *internal networking*.

Rețelele interne sunt create automat, de aceea nu există o configurare centrală. Fiecare rețea internă este identificată după numele acesteia. Odată ce sunt active mai multe rețele virtuale cu același identificator de rețea internă, driverul suport al mașina virtuala face automat conectarea și acționează ca un switch de rețea. Driverul suport al mașinii virtuale implementează un switch Ethernet complet și suporta atât broadcast/multicast cât și modul promiscuous.

Modul promiscuous este cel în care adaptorul de rețea asculta la tot traficul de rețea, nu numai la cel care îi este destinat. De exemplu în Ethernet un adaptor își da seama că traficul îi este adresat în funcție de adresa MAC destinație. În modul promiscuous, orice trafic este capturat, indiferent de adresa MAC.

HOST-ONLY NETWORKING

Rețea de tip host-only (doar gazda) este un alt mod de rețea și poate fi considerat un hybrid între modurile de rețea bridge și intern. Ca și în cazul rețelei bridge, mașinile virtuale pot comunica între ele dar și cu sistemul gazda ca și cum ar fi conectate fizic printr-un switch de rețea. Similar cu rețeaua internă, interfața de rețea fizică nu este necesară și mașina virtuală nu poate să comunice cu lumea exterioară gazdei.

Atunci când se utilizează rețele *host only*, mașina virtuală creează o nouă interfață software pe gazda care apare alături de interfețele de rețea existente. În timp ce rețea bridged folosește o interfață fizică existentă pentru a atașa mașinile virtuale, rețeaua host-only folosește o interfață loopback pe mașina gazda. În concluzie traficul dintre mașinile virtuale nu pot fi văzut din exterior însă mașina gazdă poate comunica cu acestea.

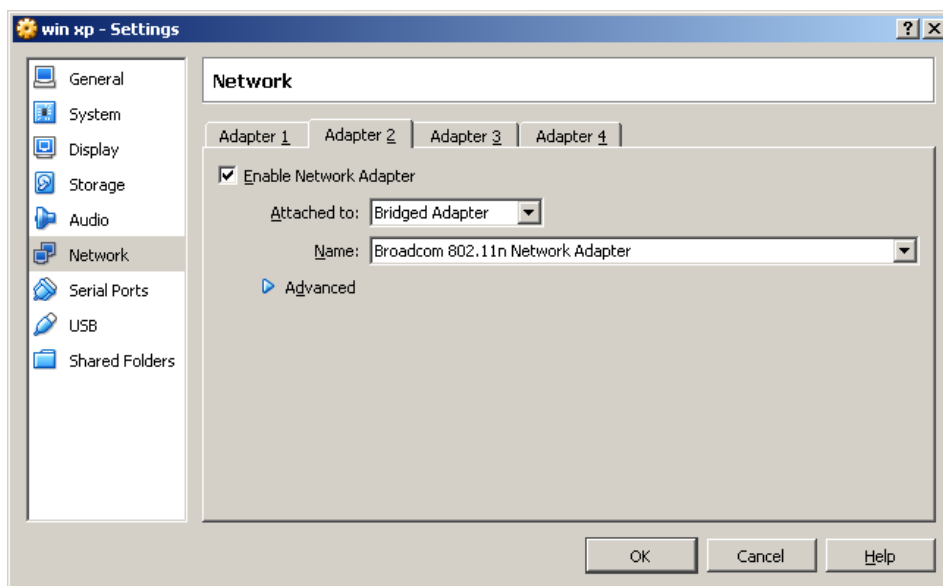


Figura 1 VirtualBox permite mai multe adaptoare de rețea pentru fiecare mașină virtuală

Modul implicit de conectare a sistemului virtual în VirtualBox la rețea este NAT.

Aplicatii

Conectarea unei mașini virtuale la rețea în modul Network Address Translation (NAT)

Se va utiliza VirtualBox pentru a porni o mașină virtuală cu o placă de rețea în modul NAT (modul implicit) și sistemul de operare Linux.

Se va observa că sistemul de operare si-a configurat automat adresa IP, gateway, DNS, etc. Este posibilă fără alte setări accesarea site-urilor internet fără alte setări suplimentare.

Dacă se trimite ping de pe sistemul gazda și de pe sistemul oaspete către aceeași adresă de internet (ex. ping 8.8.8.8), valoarea TTL este mai mică cu 1 pentru sistemul virtual, indicând prezența unui ruter suplimentar (asigurat de mașina virtuală) între sistemul virtual și rețeaua fizică.

Se observă suplimentar că latența rețelei este mai mică pentru sistemul real ($<1\text{ms}$) și puțin mai mare ($=1\text{ms}$) pentru mașina virtuală. Acest fenomen este cu atât mai vizibil cu cât mașina pe care rulează VirtualBox este mai lentă.

Adresa IP a mașinii virtuale nu poate fi accesată de alte calculatoare din rețeaua sistemului real (gază) deoarece nu este configurat port forwarding pe ruterul virtual.

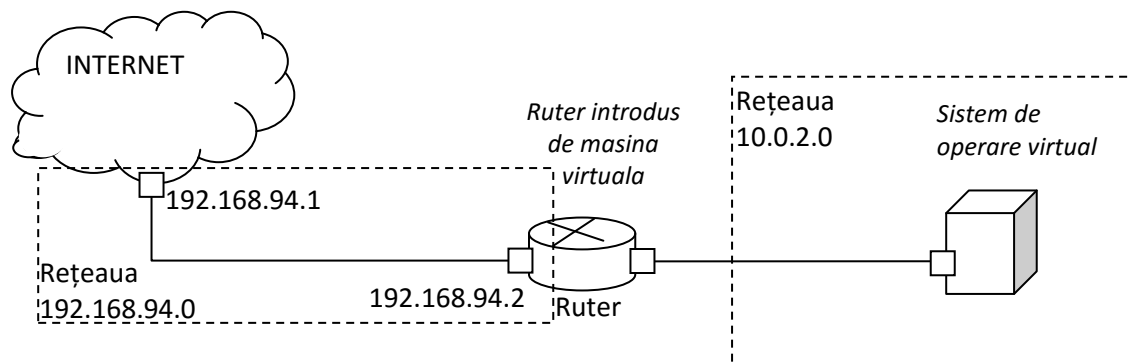


Figura 2 Configurația rețelei pentru modul NAT

Conectarea unei mașini virtuale la rețea în modul Bridge

Se va utiliza VirtualBox pentru a porni o mașină virtuală cu o placă de rețea în modul NAT (modul implicit) și sistemul de operare Linux.

Se observă că după pornirea sistemului de operare adresa IP nu este configurată automat (presupunem că rețeaua calculatorului gazdă 192.168.94.0 nu are activ un server DHCP) și trebuie configurată adresa IP în modul de root (considerăm configurația din figura):

```
ifconfig eth0 192.168.94.3
```

Dacă se vrea acces în rețele externe se configurează gateway:

```
route add default gw 192.168.94.1
```

În acest moment se poate testa conectivitatea cu exteriorul (ping 8.8.8.8)

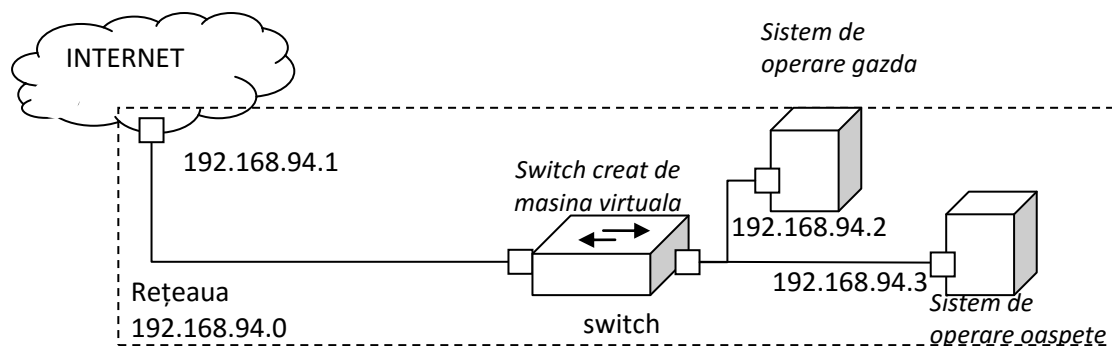


Figura 3 Configurația rețelei pentru modul Bridge

Se observă totodată că TTL are aceeași valoare pe sistemul gazdă cât și pe sistemul oaspete și latența către un sistem extern este aproximativ egală pentru cele două sisteme.

Sistemul gazda poate fi accesat de toate calculatoarele din rețeaua calculatorului oaspete și traficul de broadcast al mașinii virtuale este vizibil în această rețea. Aceasta ne arată că avem de a face cu o conectivitate realizată printr-un switch creat de mașina virtuală.

Desfasurarea lucrării

1. Se va studia breviarul teoretic.

2. Se va urmări ghidul pentru instalarea și configurarea unui sistem de operare virtual în VirtualBox și se va instala sistemul de operare Ubuntu Linux.

3. După instalare se vor efectua următoarele modificări asupra sistemului de operare instalat și se vor nota care modificări pot fi aplicate în timpul funcționării și care necesită repornirea mașinii virtuale pentru a fi aplicate:

- se va modifica dimensiunea memoriei sistemului de operare virtual;
- se va modifica modul de conectare la rețea din NAT în Bridged și se va reconfigura rețeaua;
- se va adăuga un disk suplimentar la sistemul de operare virtual și se va asigura accesarea acestuia de către mașina virtuală;
- se vor testa diferite metode de oprire a mașinii virtuale și efectele acestora asupra sistemului de operare instalat în VM.

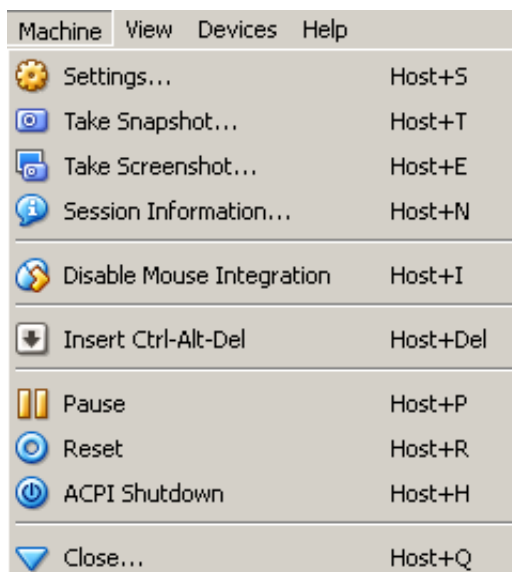


Figura 4 Modalități de repornire mașină virtuală

- se va salva un Snapshot al mașinii virtuale, se vor efectua apoi modificări (crearea unui nou fișier, etc.) după care se va restaura mașina virtuală la starea salvată în Snapshot. Cum apreciați viteza acestui proces comparativ cu reinstalarea sistemului de operare? Ce avantaje vedeți pentru această funcționalitate?

- se vor modifica perifericele conectate la masina virtuală și modul de partajare a memoriei cu sistemul de operare gazda

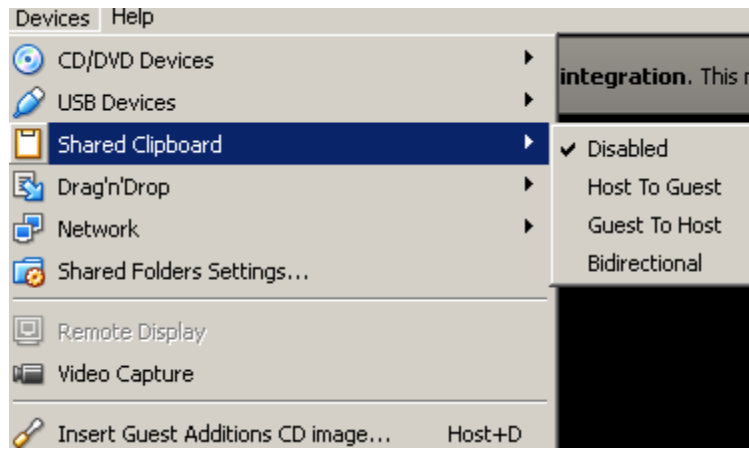


Figura 5 Modalități de interacțiune între clipboard-ul mașinii reale și cel al mașinii virtuale

4. Se va încerca monitorizarea traficului de rețea cu Wireshark pentru adaptoarele de rețea disponibile pe sistemul de virtualizare. Se vor observa problemele și limitările existente.

Capitolul 9. Securizarea rețelelor locale. VLAN. Interfețe virtuale și subinterfețe

Obiectivul lucrării

Lucrarea își propune studierea câtorva metode pentru securizarea comunicațiilor în rețele de calculatoare în general și în rețele locale în particular. Se va studia rolul subinterfețelor și al interfețelor virtuale (numite și IP alias) în rețelele locale de calculatoare. Subinterfețele permit comunicarea între VLAN-uri în rețelele locale. Configurarea interfețelor virtuale permite unui sistem de calcul să răspundă la mai multe adrese IP chiar dacă acesta este dotat cu o singură placă fizică de rețea. Aplicațiile investighează instalarea și configurarea rețelelor VLAN.

Breviar teoretic

Subinterfețe

În telecomunicații și rețele de calculatoare, o subinterfață este divizarea unei interfețe fizice în mai multe interfețe logice. Ruterele folosesc subinterfețele pentru o varietate de scopuri, cele mai frecvente dintre acestea sunt pentru rutarea traficului între VLAN-uri și în rețele precum frame relay sau ATM.

Subinterfețele sunt necesare deoarece un router nu are de obicei un număr de interfețe fizice egal cu numărul de rețele pe care trebuie să le gestioneze.

Router-ul prezentat în figura poate fi achiziționat inițial doar două porturi FastEthernet, însă are mai multe module care pot fi adăugate ulterior în cazul în care rețeaua se extinde sau este necesară conectarea la alte medii de transmisie (serial, ISDN, etc.).



Figura 1 Router Cisco 2800 un număr redus de porturi de rețea și un număr limitat fizic de module în care se pot instala porturi adiționale

Subinterfețele sunt necesare pentru găzduirea mai multor servicii diferite pe același calculator fizic, pentru încărcarea echilibrată a rețelei și orice aplicație care are nevoie de mai multe IP-uri pe aceeași interfață fizică.

Interfețe virtuale

Interfețele virtuale sunt numite și IP alias și permit calculatoarelor să aibă un număr mai mare de adrese IP decât numărul de interfețe fizice. În acest mod este posibil ca un site web, de exemplu, să ofere pagini distincte pentru interfețele virtuale configurate pentru a utiliza mai bine resursele serverului.

În figură de mai jos se observa că în cazul în care un server mai puternic preia sarcinile mai multor servere, el poate face acest lucru păstrând adresele IP ale calculatoarelor inițiale.

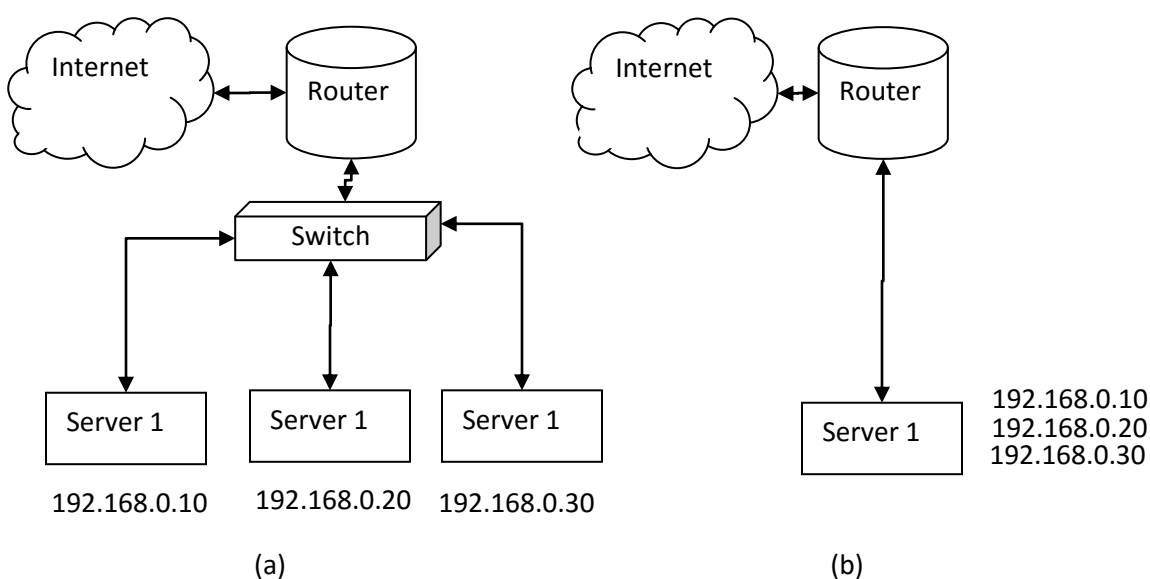


Figura 2 Prin folosirea interfețelor virtuale un singur sistem de calcul poate avea mai multe servere fără modificări la structura IP

Virtual Local Area Network (VLAN)

VLAN - Virtual Local Area Networks sunt utilizate pentru a împărți o rețea fizică în mai multe domenii de difuzare, izolate reciproc, fără a ține cont de gruparea lor fizică. Motivația pentru a utiliza VLAN-uri este pentru a diviza o rețea și pentru a separa calculatoarele astfel încât, deși ele se conectează la același echipament fizic, să nu poată să se acceseze unul pe altul în mod direct.

În mod uzual însă numărul rețelelor locale pe care un router le poate gestiona este mult mai mare decât numărul porturilor fizice pe care acesta le are. Fiecare rețea necesită însă o interfață separată.

Pentru a rezolva această problemă se conectează la fiecare interfață a routerului unul sau mai multe switch-uri, care permit concentrarea traficului provenind de la mai multe calculatoare către interfața routerului. Acest model de interconectare are avantajul unei siguranțe bune deoarece un switch este alocat unei anumite rețele însă nu este foarte flexibil.

Sa ne imaginam ca utilizatorii nu sunt grupati în functie de rețeaua în care fac parte și gasim frecvent mai multe grupuri de utilizatori la aceeași locație. Fiecare grup ar necesita propriul switch, lucru care ar duce la costuri ridicate de implementare.

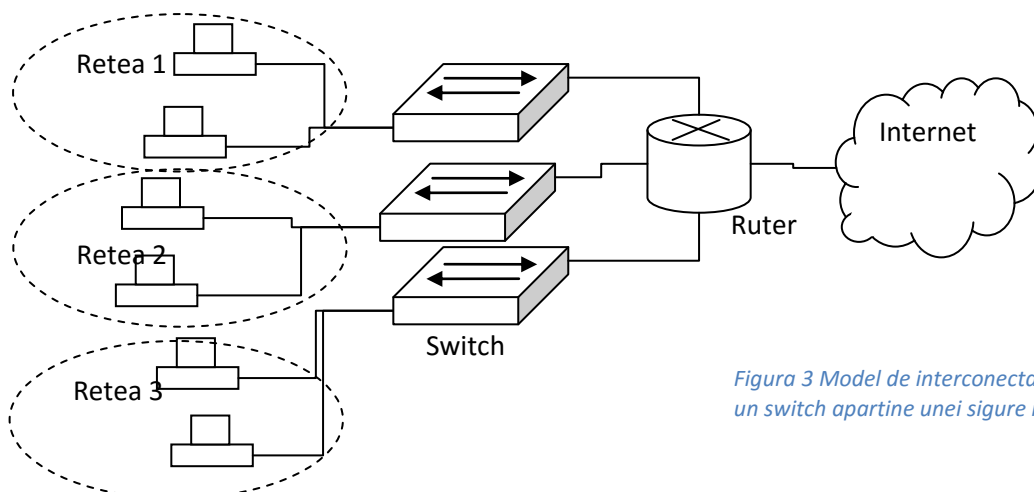


Figura 3 Model de interconectare în care un switch aparține unei singure rețele

Soluția în acest caz este utilizarea unui switch cu suport pentru VLAN pe care poate specifica rețele virtuale și care poate alocă porturile sale acestor rețele. În acest mod traficul intrat pe porturile unui VLAN nu poate trimite trafic din porturile altui VLAN. Către router vor porni atâtea conexiuni câte VLAN-uri sunt pe switch. Singura cerință în acest caz pentru router este să aibă suficient de multe interfețe fizice pentru conexiunile care vin de la switch.

Routerul este cel care va permite trecerea traficului dintr-o rețea în alta.

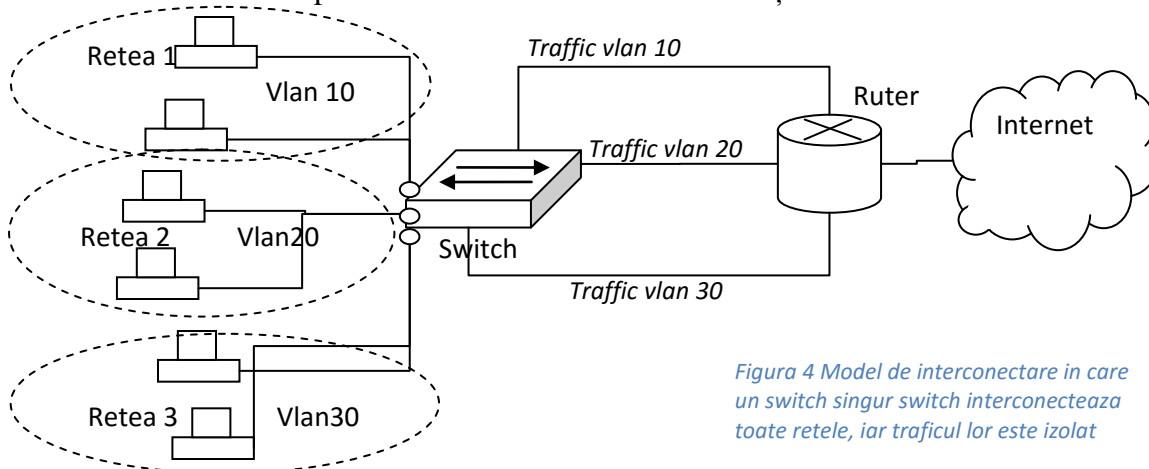


Figura 4 Model de interconectare în care un switch singur interconectează toate rețele, iar traficul lor este izolat

În mod uzual însă numărul porturilor fizice ale unui router este foarte mic, fiind multe care au doar o singură interfață fizică. Pe de altă parte numărul VLAN-urilor este foarte mare (de la 1 la 4094).

Pentru a rezolva situația în care routerul are un număr mic de interfețe fizice se conectează rețeaua prin intermediul switch-urilor cu suport de VLAN, în care traficul din rețelele aparținând unor VLAN-uri diferite este transmis printr-o singură interfață fizică la server numită Trunk. Routerul, având configurate subinterfețe pentru conexiunea de trunk, direcționează traficul din fiecare VLAN către subinterfața corespunzătoare.

În figura următoare se observă două situații:

- Cazul în care numărul de rețele este mai mic decât numărul porturilor ruterului: în aceasta situație ruterul nu are nevoie de suport pentru VLAN, decât switch-ul trebuie să poată separa rețelele astfel încât să ajunga la ruter pe interfețe diferite.
- Cazul în care numărul de rețele este mai mare decât numărul porturilor ruterului: în aceasta situație ruterul are nevoie de suport VLAN și anumite să înțeleagă încapsularea 802.1q specifică liniilor de trunk prin care circula amestecat fluxuri din VLAN-uri diferite (fiecare flux fiind însă marcat corespunzător cu ID-ul VLAN-ului din care face parte).

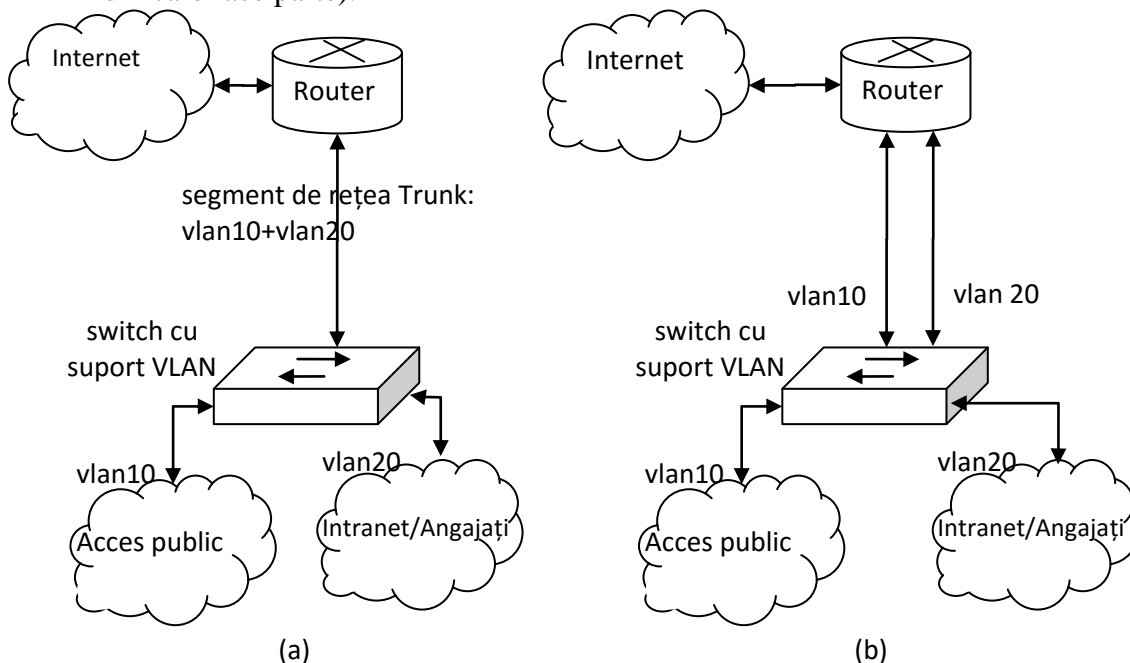


Figura 5 Structura unei rețele cu 2 VLAN-uri și o singură legătură fizică la router (a) respectiv două interfețe fizice la ruter (b)

Standardul IEEE 802.1q care realizează transmiterea informațiilor de VLAN este utilizat doar pe liniile de trunk (prin care circula traficul de la mai multe VLAN-uri) și are un pachet care utilizează la baza formatul unui cadru Ethernet, în care sunt inserați 32 de biți. Uneori providerii de internet folosesc standardul 802.1ad pentru eticheta suplimentară traficul de VLAN astfel încât să poată transmite prin rețeaua internă traficul clienților care poartă deja eticheta 802.1q. Implementarea acestui standard realizează o dublă marcare a pachetelor: marcarea inițială cu VLAN-ul clienților și marcarea providerului cu VLAN-urile acestora. Pachetele pentru cele trei încapsulări sunt prezentate în continuare.

Ethernet

Preambul	MAC destinație	MAC sursă	Tip/Dimensiune	Date	CRC
----------	----------------	-----------	----------------	------	-----

Ethernet + Marcare 802.1q

Preambul	MAC destinație	MAC sursă	Antet 802.1q	Tip/Dimensiune	Date	CRC
----------	----------------	-----------	--------------	----------------	------	-----

Ethernet+marcare dubla 802.1ad

Preambul	MAC destinație	MAC sursă	Antet Provider 802.1q	Antet Client 802.1q	Tip/Dimensiune	Date	CRC
----------	----------------	-----------	-----------------------	---------------------	----------------	------	-----

Antetul 802.1q conține 32 de biți, din care 16 biți identifica tipul cadrului Ethernet drept 802.1q iar următorii 16 biți sunt rezervați pentru identificatorul de VLAN și diferiți indicatori.

Aplicații

Hardware

Pentru a fi capabil să facă uz de VLAN-uri va avea nevoie de un switch care acceptă standardul IEEE 802.1q. Veți avea nevoie, de asemenea, de un sistem de operare și o placă de rețea capabilă de a transmite și primi pachete etichetate 802.1q.

Software

Ubuntu Linux suportă VLAN-uri în mod implicit dar este nevoie de un instrument suplimentar care trebuie să fie instalat pentru a crea interfețe care să suporte VLAN.

```
sudo apt-get install vlan
```

Este necesară în continuare încărcarea modulului 8021q în kernel:

```
sudo modprobe 8021q
```

Acum se va crea o nouă interfață, care este un membru al unui anumit VLAN. În exemplu este utilizat VLAN ID-ul 10:

```
sudo vconfig add eth1 10
```

Se realizează configurarea unei adrese pentru noua interfață:

```
sudo ip addr add 10.0.0.1/24 dev eth1.10
```

Pentru a face aceste modificari permanente, trebuie adăugate informații la unele fișiere de configurare. Adăugarea modulului la kernel în momentul pornirii sistemului de operare se face cu:

```
sudo su -c 'echo "8021q" >> /etc/modules'
```

Suplimentar este necesar ca subinterfața să fie disponibilă la momentul pornirii sistemului de operare. Aceasta trebuie realizată prin modificarea fișierului /etc/network/interfaces:

```
auto eth1.10
iface eth1.10 inet static
    address 10.0.0.1
    netmask 255.255.255.0
    vlan-raw-device eth1
```

În final se poate configura rutarea între interfețele nou create așa cum se realizează rutarea între oricare alte două interfețe pe un ruter Linux.

Configurarea interfețelor virtuale în Linux

Crearea interfețelor virtuale poate fi o sarcină dificilă în Linux, deoarece în majoritatea distribuțiilor lipsesc uneltele grafice destinate acestui scop. Deoarece serverele Linux au de

obicei interfața în modul consola, în continuare sunt prezentate metodele pentru configurarea interfețelor virtuale în Linux, atât individual cât și pentru game de adrese IP.

Crearea interfețelor virtuale temporare se realizează executând comanda:

```
ifconfig eth0:0 192.168.100.25
```

Comandă va crea o interfață virtuală de rețea bazată pe interfață fizică eth0. Condiția pentru crearea interfeței virtuale este că interfața fizică pe care esteasta se bazează să existe.

```
# ifconfig eth0
```

```
eth0 Link encap:Ethernet HWaddr 1a:27:1c:42:99:e1
```

```
inet addr:192.168.100.25 Bcast:192.168.100.255
```

```
Mask:255.255.255.0
```

În acest moment interfața virtuală poate fi configurată.

```
# ifconfig eth0:0
```

```
eth0:0 Link encap:Ethernet HWaddr 1a:27:1c:42:99:e1
```

```
UP BROADCAST MULTICAST MTU:1500 Metric:1
```

```
Interrupt:20 Memory:f1600000-f1620000
```

```
# ifconfig eth0:0 192.168.90.2
```

```
# ifconfig eth0:0
```

```
eth0:0 Link encap:Ethernet HWaddr 1a:27:1c:42:99:e1
```

```
inet addr: 192.168.90.2 Bcast: 192.168.100.255 Mask: 255.255.255.0
```

Odată terminată configurarea, interfața virtuală poate fi utilizată imediat.

```
# ping 192.168.90.2
```

```
PING 192.168.90.2 (192.168.90.2) 56(84) bytes of data.
```

```
64 bytes from 192.168.90.2: icmp_req=1 ttl=64 time=0.122 ms
```

```
64 bytes from 192.168.90.22: icmp_req=2 ttl=64 time=0.105 ms
```

Dezactivarea interfeței virtuale se realizează prin utilizarea parametrului down pentru interfața virtuală respectivă.

```
# ifconfig eth0:0 down
```

Dezactivarea interfeței virtuale se realizează prin utilizarea parametrilor up pentru interfața virtuală respectivă.

```
# ifconfig eth0:0 up
```

Configurarea permanentă a interfețelor virtuale este necesară deoarece configurările anterioare sunt valabile doar până în momentul în care mașina va fi repornită. În acest scop este necesară modificarea fișierului de configurare din /etc/network/interfaces pentru distribuția Ubuntu Linux. Deschideți fișierul /etc/network/interfaces cu editorul de text.

```
$ mcedit /etc/network/interfaces
```

```
iface eth0:0 inet static
address 192.168.90.2
netmask 255.255.255.0
broadcast 192.168.90.255
```

Este posibilă folosirea interfețelor virtuale cu DHCP. în acest caz este necesară modificarea /etc/network/interfaces în modul următor.

```
iface eth0:0 inet dhcp
```

Aplicarea modificărilor la configurarea rețelei se realizează cu:

```
# /etc/init.d/networking restart
```

Puteți adăuga întotdeauna mai multe subinterfețe în Ubuntu de exemplu: eth0:1, eth0:2 , eth0:3, etc.

Trebuie precizat că adresa IP a subinterfeței trebuie să **nu** fie din aceeași rețea cu adresele rețelei originale sau ale celorlalte subinterfețe.

Configurarea interfețelor virtuale în Windows

Pentru suportarea mai multor adrese IP în Windows pentru o singură interfață fizică este necesară crearea unui adaptor de rețea virtual. În acest mod este posibil să înlocuim cu o singură mașină fizică mai multe alte servere fără a face alte modificări în rețea.

Interfețele virtuale Windows permit unui adaptor de rețea să poată răspunde cererilor mai multor clienți adresate adreselor IP care au fost configurate.

Acesta se realizează pentru IPv4 prin accesarea proprietăților rețelei locale:

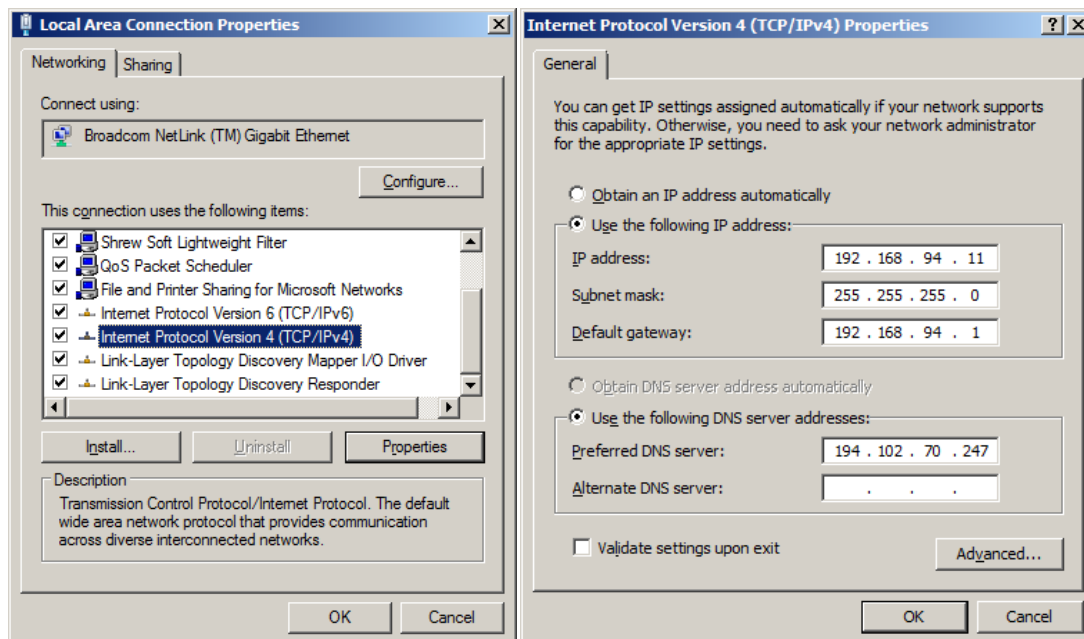


Figura 6 Accesarea proprietăților adaptorului de rețea pentru accesarea proprietăților avansate

Se apasă în continuare butonul "Advanced..." și în fereastra care se deschide se pot adauga mai multe adrese IP pentru un singur adaptor de retea.

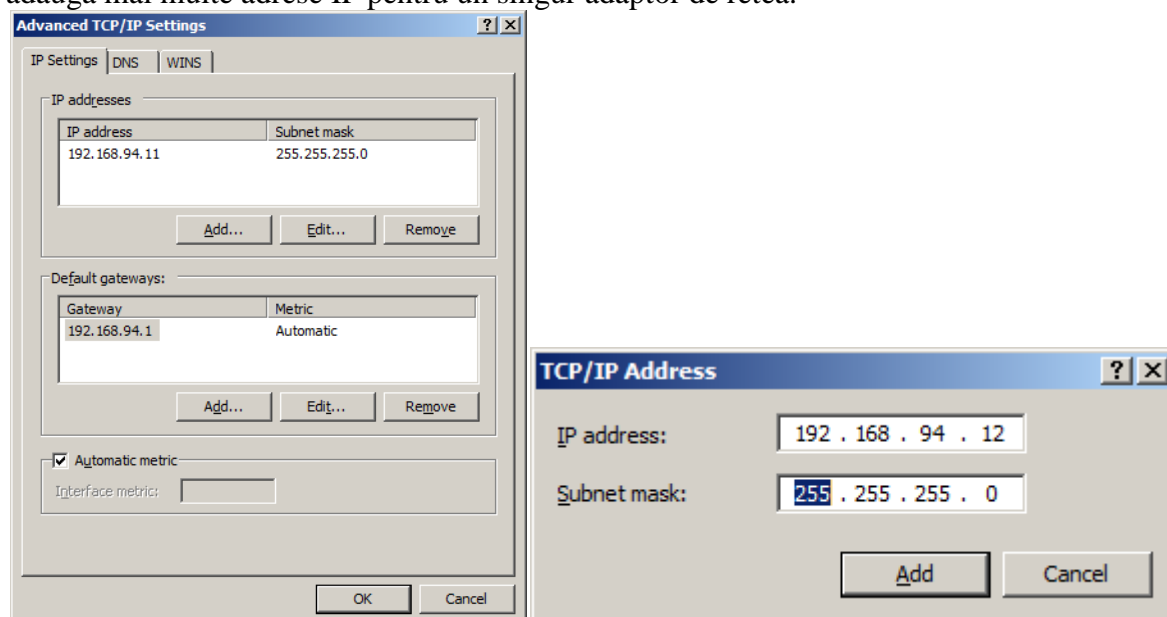


Figura 7 Adaugarea adreselor IP multiple pentru un singur adaptor de rețea

În final se apasă "Add" respectiv OK pentru a salva modificările făcute la configurația adaptorului de rețea. Același lucru se poate realiza și pentru adresa de gateway, pentru configurarea adreselor de backup.

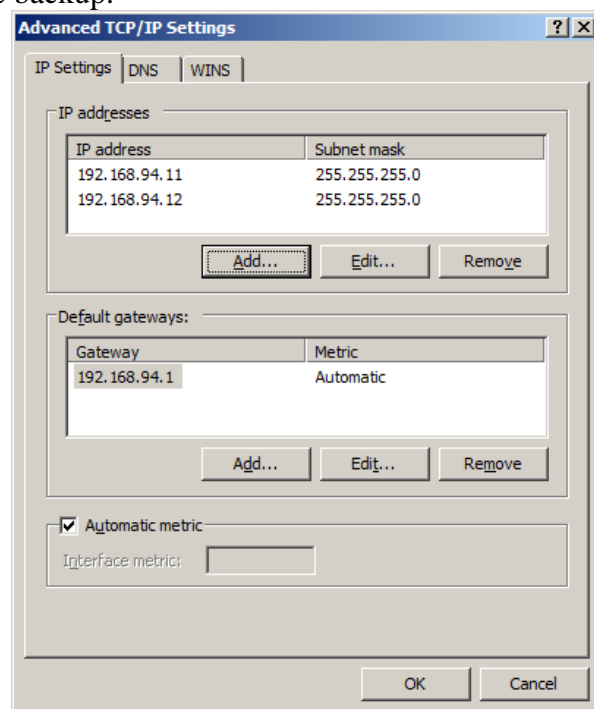


Figura 8 Adaptor de rețea cu mai multe adrese IP configurate

Interfața virtuală astfel creată poate fi vizualizată în consola cu ajutorul comenzii `ipconfig /all` și poate fi utilizată ca oricare din adresele configurate anterior.

```
Wireless LAN adapter Wireless Network Connection:

Connection-specific DNS Suffix . : 
Description . . . . . : Broadcom 802.11n Network Adapter
Physical Address. . . . . : 38-59-F9-9C-AA-86
DHCP Enabled. . . . . : No
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::552b:fcf:1da2:75dd%12<Preferred>
IPv4 Address. . . . . : 192.168.94.110<Preferred>
Subnet Mask . . . . . : 255.255.255.0
IPv4 Address. . . . . : 192.168.94.111<Preferred>
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.94.254
DHCPv6 IAID . . . . . : 305682937
DHCPv6 Client DUID. . . . . : 00-01-00-01-17-ED-ED-B1-B8-70-F4-E9-7B-E1

DNS Servers . . . . . : 192.129.4.1
NetBIOS over Tcpip. . . . . : Enabled
```

Figura 9 Adaptor de rețea cu mai multe adrese IP configurate – rezultatul comenzii `ipconfig/all`

Se vor testa interfețele virtuale prin trimiterea de pachete ICMP folosind aplicația `ping`. Atât 192.168.94.110 cât și 192.168.94.111 vor răspunde.

Desfășurarea lucrării

1. Se va studia breviarul teoretic.
2. Se va studia modul în care a fost instalată și configurată aplicația și se vor configura pe calculator subinterfețe în sistemul de operare Ubuntu Linux, respectiv IP alias pentru sistemul de operare Windows.
3. Se vor utiliza aplicațiile de monitorizare a traficului în rețea (Wireshark) pentru a verifica schimbul de date între calculatoare.
4. Se va configura un server HTTP în Windows folosind pachetul de aplicații XAMPP pentru a răspunde diferit la cereri destinate diferitelor interfețe virtuale (alias).
5. Se vor instala două VLAN-uri care schimbă informațiile prin același switch. Se va observa comportamentul rețelei la încercarea de a comunica cu echipamentele din alt VLAN fără prezența unui ruter.
6. Se va monitoriza comunicația din rețea cu aplicația Wireshark.

Capitolul 10. Virtualizare stocare. Configurare Storage Area Network folosind protocolul iSCSI

Obiectivul lucrării

Lucrarea își propune studierea protocolului iSCSI și configurarea unui server de stocare de date, pentru sistemul de operare Linux, care folosește acest protocol. Clienții vor fi utiliza sistemul de operare Windows XP și Windows 7.

Breviar teoretic

Stocarea datelor este o problema datorita cantității conținutului multimedia care este transmis prin internet.

Internet Small Computer System Interface - iSCSI este un protocol folosit în rețelele SAN (Storage Area Network) pentru stocarea datelor bazat pe standardul de rețea IP pentru conectarea facilităților de stocare a datelor. iSCSI a fost elaborat de IBM și Cisco la sfârșitul anilor '90 și introdus publicului larg în 2000. Spre deosebire de alte tehnologii similare (precum rețelele Fiber Channel, care necesită de obicei cablare dedicată), iSCSI poate fi utilizat pe distanțe lungi, folosind infrastructura de rețea existentă.

Prin transmiterea de comenzi SCSI prin rețelele IP, iSCSI este folosit pentru a facilita transferurile de date prin intranet și pentru a gestiona spațiul de stocare la distanță. iSCSI pot fi folosit pentru a transmite date prin rețele locale (LAN), rețelele de arie largă (WAN) sau prin Internet și poate permite poziționarea stocării datelor independent de locația în care aceasta este accesată.

Protocolul permite clienților (numiți inițiatori - iSCSI Inițiator) să trimită comenzi SCSI la dispozitive de stocare SCSI (numite ținte - iSCSI Target) situate pe serverele de la distanță. Prin acest protocol se poate crea o Storage Area Network (SAN), permițând organizațiilor să-și implementeze stocarea în centre de date specializate, oferind clienților (cum ar fi servere de baze de date și web) impresia că unitățile de stocare a datelor sunt atașate local. Viteza unei rețele trebuie să fie mare (1Gbps) pentru a nu exista întârzieri și limitări în accesarea spațiului de stocare accesat prin iSCSI, dar funcționează perfect și pe rețele mai lente. În rețelele iSCSI se recomandă conectarea unui singur inițiator să un singur target pentru a preveni coruperea resurselor.

Suportul la nivelul sistemului de operare pentru protocolul iSCSI este larg, majoritatea sistemelor de operare au posibilitatea sa fie atât inițiator iSCSI cât și ținta iSCSI. iSCSI nu poate fi folosit în mod normal ca partiție de boot a unui sistem de operare.

iSCSI utilizează TCP (de obicei porturile TCP 860 și 3260) drept protocol pentru transmiterea datelor și comenzilor și nume de nivel înalt folosite pentru adresarea obiectelor în interiorul protocolului. iSCSI definește trei tipuri de nume (iSCSI Qualified Name (IQN)) care pot fi folosite într-o rețea iSCSI:

- formatul IQN (iSCSI Qualified Name) definit în RFC 3720, cu alte exemple de nume în RFC 3721. Este cel mai folosit format. Exemplu de nume:

Tip	Data	Autoritatea de nume	Text definit de autoritatea de nume
iqn.	1992-01.	com.example:	storage:diskarrays-sn-a8675309

- formatul Extended Unique Identifier (EUI). Utilizat de IEEE Registration Authority. Format: . Eui {EUI-64 de biți de adresa}. Exemplu: eui.02004567A425678D)
- formatul T11 Network Address Authority (NAA) adăugat în RFC 3980 pentru compatibilitate cu numele folosite în standardele Fiber Channel și SAS (Serial Attached Storage) Format: naa.{ identificator NAA de 64 sau 128 de biți }. Exemplu: naa.52004567BA64678D)

Aplicații

Aplicația prezentată va acoperi pașii necesari pentru a configura un target iSCSI pe sistemul de operare Ubuntu pe 64 de biți sau pe 32biți și cei necesari un client Windows.

Actualizarea surselor sistemului Linux

Înainte ca un target iSCSI să poată fi instalat trebuie verificata actualizarea surselor de instalare pentru sistemul de operare.

```
sudo apt-get update
```

Uneori este necesara în continuare și actualizarea sistemului de operare cu:

```
sudo apt-get upgrade
```

Instalare target iSCSI

Pachetul de baza necesar este iscsitarget însă este necesar și pachetul iscsitarget-dkms pentru a putea porni modulul *iscsi_trgt* necesar funcționarii iSCSI. Următoarea comanda executata în modul root instalează pachetele necesare pentru targetul iSCSI:

```
sudo apt-get install iscsitarget iscsitarget-dkms
```

Configurarea targetului iSCSI

Configurarea se realizează prin editarea fișierului: `/etc/default/iscsitarget`

Se poate folosi în acest scop `mcedit`, `nano` sau alt editor de fișiere text. Se modifică în aceasta linia `ISCSITARGET_ENABLE = false` în `ISCSITARGET_ENABLE = true`

Configurarea stocării

În terminologia iSCSI un LUN este un dispozitiv SCSI adresabil în mod individual care este parte a unui dispozitiv target SCSI. Un inițiator negociază cu un target pentru a stabili conectivitatea la un LUN. Rezultatul acestui proces este o conexiune iSCSI care emulează o conexiune cu un hard-disk de tipul SCSI. În acest mod un LUN este tratat ca un hard-disk obișnuit de către sistemul de operare inițiator care poate de exemplu să îl formateze, nu doar să îl acceseze sub formă de directoare ca în cazul sistemului de fișiere în rețea NFS. Numerele LUN trebuie să înceapă de la 0.

Pentru stocarea în fișiere se folosește utilitarul dd. În Linux unitățile de disc apar drept fișiere obișnuite în sistemul de fișiere sau drept fișiere dispozitiv precum `/dev/zero` sau `/dev/random`.

Fișierele dispozitiv sunt interfețe pentru un driver astfel încât acesta să apară sistemului de operare ca un fișier obișnuit. Scrierea și citirea în aceste fișiere se face cu apeluri de sistem obișnuite și este direcționată către dispozitivul conectat prin acel driver. Fișierul special `/dev/zero` este folosit în inițierea unui flux de caractere pentru inițializarea stocării datelor și asigură atâtea caractere nule (0x00) câte sunt solicitate de operația de citire din acest fișier.

Utilitarul dd poate citi și scrie în aceste fișiere dispozitiv dacă funcția este permisă de driver, deci poate fi folosit pentru a realiza un backup pentru un sector de boot sau alte operații precum conversia datelor. De exemplu formatarea (umplerea cu 0x00) a unei partiții `/dev/sdb1` se realizează cu:

```
dd if=/dev/zero of=/dev/sdb1
```

Crearea unui fișier de 1MB numit test, umplut cu biti de 0 se realizează cu:

```
dd if=/dev/zero of= test count=1024 bs=1024
```

Într-un LUN stocarea se poate realiza în fișiere sau poate fi reprezentată de o partiție. În cazul în care se dorește utilizarea unei partiții, se trece direct la pasul următor de creare a țintei iSCSI. Pentru cazul în care LUN-ul creat este stocat într-un fișier, utilitarul dd este folosit pentru a genera un fișier gol în care vor fi salvate datele partiției iSCSI:

```
dd if=/dev/zero of=/media/volume0/storlun0.bin count=0 obs=1 seek=5G
```

Crearea țintei iSCSI

Aceasta etapă se realizează prin modificarea fișierului:

```
/etc/iet/ietd.conf
```

Acesta conține numeroase exemple de configurare comentate. Se poate adăuga la sfârșitul fișierului configurarea conform cerințelor țintei iSCSI pe care dorim să o realizăm.

Pentru configurarea fișierului din pasul anterior drept LUN ținta al iSCSI se adăugă:

```
Target iqn.2012-05.local.mynet:storage.sys0
```

```
Lun 0 Path=/media/volume0/storlun0.bin,Type=fileio,ScsiId=lun0,ScsiSN=lun0
```

În cazul în care se dorește utilizarea de partiții (de exemplu `/dev/sda1`) pentru ținta iSCSI se folosește:

Target iqn.2012-05.local.mynet:storage.sys0

Lun 0 Path=/dev/sda1,Type=fileio

Prima linie precizează numele țintei iSCSI în conformitate cu notarea IQN iar a doua linie indica localizarea LUN.

Repornire target iSCSI

Pentru a citi configurarea realizata pentru ținta iSCSI este necesara repornirea serviciului iscsitarget astfel:

```
service iscsitarget restart
```

Testarea țintei iSCSI

Se va folosi în acest sens oricare versiune de Windows mai veche de Windows XP. Pentru Windows XP este necesară instalarea unui kit online (se va verifica dacă acesta este sau nu instalat pe sistem).

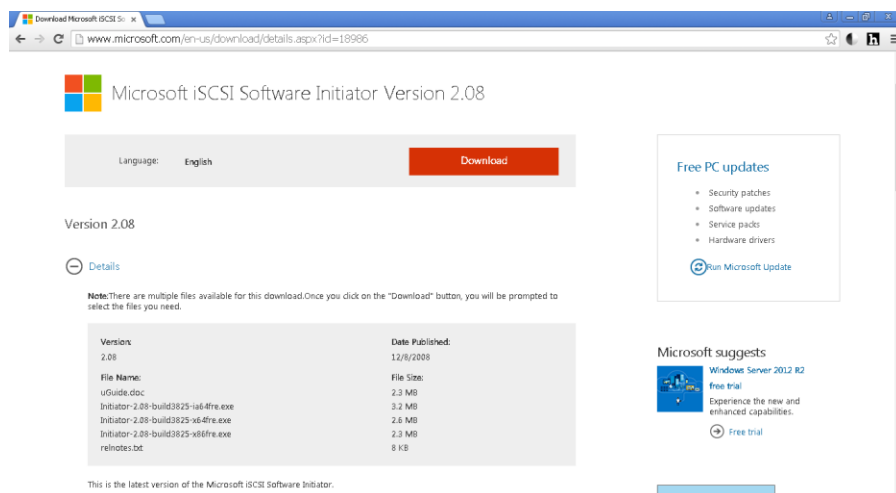


Figura 1 Windows XP are nevoie de o aplicație suplimentară pentru a accesa stocarea iSCSI

În versiunile de Windows mai noi decât Windows Vista utilitarul este integrat în sistemul de operare. Numele acestuia este iSCSI Initiator.

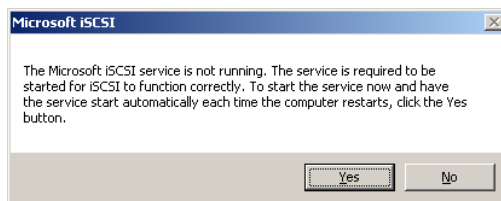


Figura 2 Serviciul iSCSI Initiator nu pornește în mod automat în versiunile noi de Windows deoarece nu este unul folosit în mod uzual

La pornirea pentru prima dată a acestui utilitar suntem atenționați ca serviciul Windows corespunzător și se solicita pornirea a acestuia.

Pornirea serviciului este necesară pentru funcționarea inițiatorului iSCSI astfel încât la repornirea sistemului partițiile LUN să fie conectate automat atât la pornirea sistemului cât și de fiecare dată când acestea devin disponibile în rețea.

Tabelul de control al serviciilor se accesează pornind aplicația *services.msc* (se apasă **Win+R** apoi se scrie *services.msc* după care se confirmă cu **Enter**).

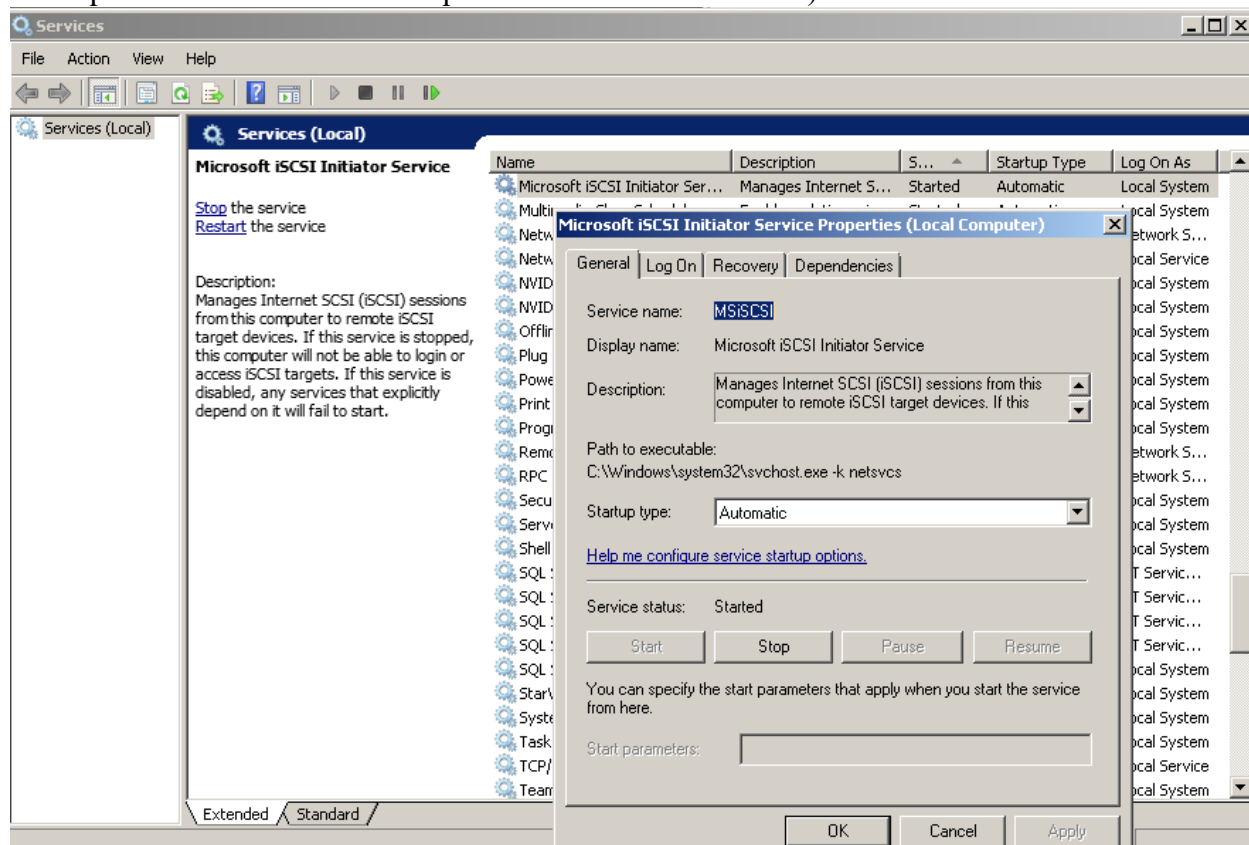


Figura 3 Pornirea serviciului în Fereastra de configurare servicii

Se introduce adresa IP a sistemului Ubuntu configurat drept țintă iSCSI apoi se apasă *Quick Connect...*

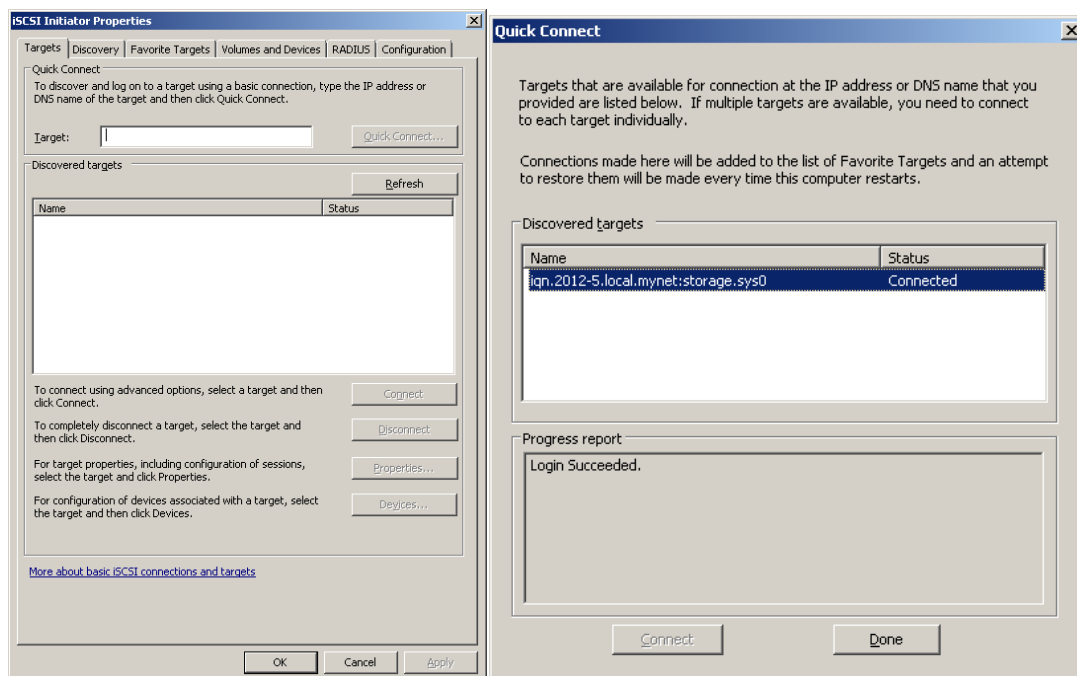


Figura 4 Identificarea unui target iSCSI

În *Computer manager* partiția este acum vizibilă însă nu este formatată. Se poate formata aceasta partiție pentru a fi vizibilă în Windows Explorer. Dacă vizualizăm proprietățile acestei partiții, se observă că este văzută drept unitate SCSI.

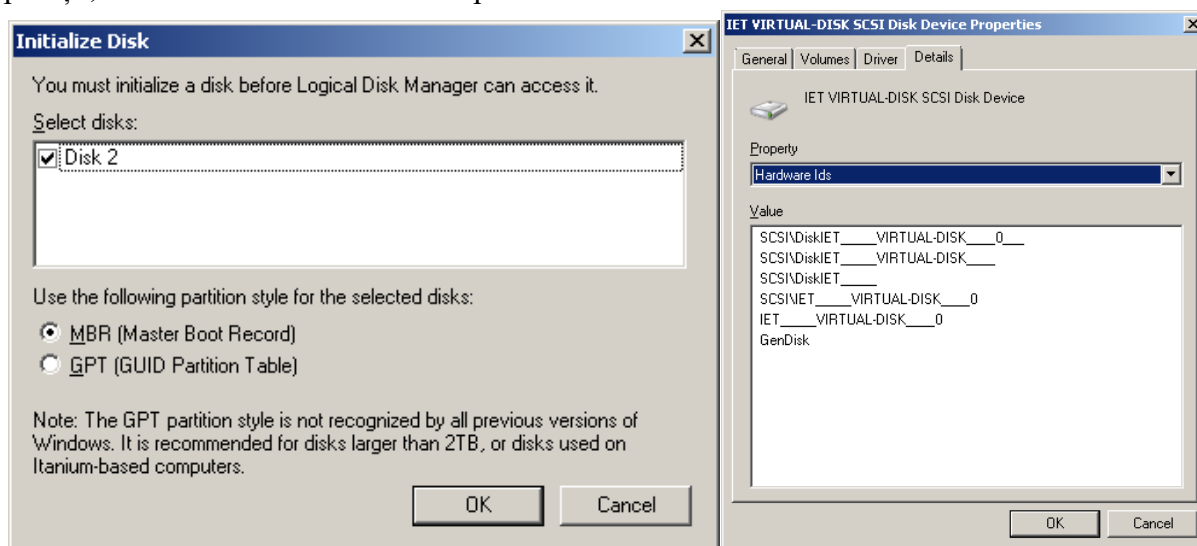


Figura 5 Un disk iSCSI trebuie initilizat inainte de a putea fi folosit

Securizarea unei partiții iSCSI

În mod normal o partiție iSCSI trebuie securizata prin separarea acesteia de restul rețelei. Suplimentar se pot folosi mecanisme în care se solicita un nume de utilizator și o parola pentru conectarea la ținta iSCSI.

Challenge-Handshake Authentication Protocol (CHAP) este un protocol care include un mecanism de prevenire a transmiterii parolelor în clar prin rețea dar și pentru prevenirea atacurilor de tip replay (retransmiterea datelor de autentificare capturate la un moment dat din rețea). CHAP este specificat de RFC 1994. CHAP este folosit și în alte mecanisme de autentificare, de exemplu *Point to Point Protocol* (PPP). Verificarea datelor de autentificare se bazează pe un secret partajat (cum ar fi parola utilizatorului clientului) și se desfășoară în următoarele etape:

- după finalizarea etapei de stabilire legăturii, autentificatorul trimite un mesaj de "provocare" pentru la egal la egal.
- clientul răspunde cu o valoare calculată folosind o funcție hash one-way bazată pe provocarea primită (challenge) și secretul cunoscut .
- autentificatorul verifică răspunsul prin compararea cu propriul calcul al valorii așteptate. Dacă valorile coincid, autentificatorul recunoaște autentificarea; în caz contrar va pune capăt conexiunii .
- la intervale aleatorii autentificatorul trimite o nouă provocare pentru către client și repetă pașii anteriori.

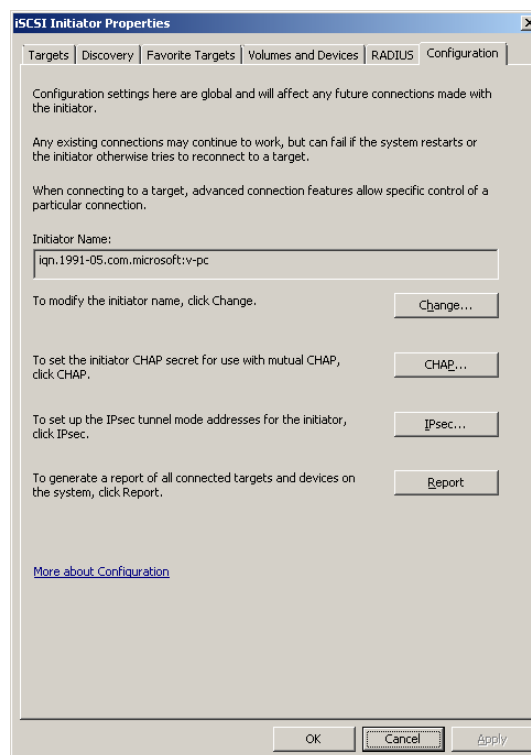


Figura 6 Interfața care permite configurarea modului în care sunt accesate resursele iSCSI

În mod normal toate calculatoarele sunt acceptate. Dacă se dorește restricționarea adreselor IP de la care este permisă conectarea la un anumit LUN, se modifică fișierul: `/etc/iet/initiators.allow` ca în rândul de mai jos, în care 192.168.0.2 este IP-ul permis:

```
iqn.2012-05.local.mynet:storage.sys0 192.168.0.2
```

Alternativ se poate introduce o plajă de adrese IP prin precizarea adresei de rețea și a măștii dorite:

ALL 192.168.0.0/16

Procesarea restricțiilor se termina când fișierul `/etc/iet/initiators.allow` a fost parcurs complet sau s-a ajuns la linia ALL ALL care permite accesul tuturor calculatoarelor.

Pentru activare CHAP se vor adăuga la ținta iSCSI în `/etc/iet/ietd.conf` liniile care vor stabili activarea CHAP, numele de utilizator și parolă:

```
Target iqn.2012-05.local.mynet:storage.sys0
```

```
IncomingUser user1 secret
```

```
OutgoingUser
```

```
Lun 0 Path=/dev/sda1,Type=fileio
```

Pentru inițiatorul iSCSI, la momentul conectării vor trebui precizate numele de utilizator și parola CHAP. În Windows 7 acest lucru se face astfel: în tab-ul Discovery se apasă "Discover Portal", apoi se adaugă IP-ul și portul țintei iSCSI, după care se selectează în meniul "Advanced"

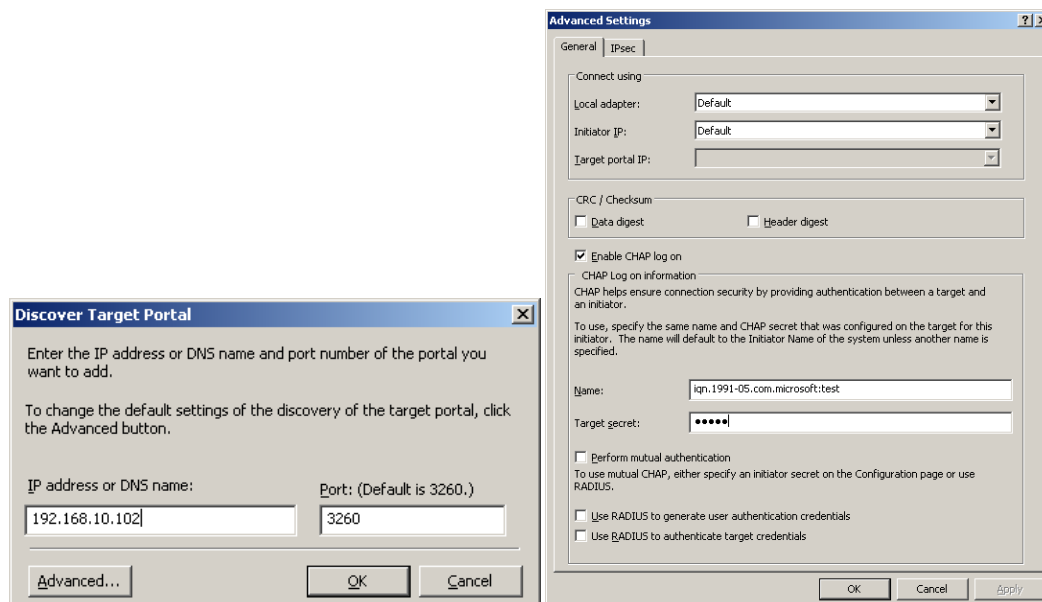


Figura 7 Accesarea resurselor iSCSI cu mecanismul CHAP necesita o parola

Se observă formatul special în care trebuie introdus numele de utilizator: `iqn.1991-05.com.microsoft:test`

Pentru ca un sistem să încerce să se conecteze la un LUN iSCSI acesta trebuie adăugat în lista de target-uri iSCSI favorite în Windows.

Configurare inițiator Linux

Configurarea conectării la partiția iSCSI din alt sistem Linux se face în modul următor:

Se instalează pachetul pentru inițiator iSCSI:

```
apt-get install open-iscsi
```

Dacă se dorește pornirea automată a conexiunii la target se modifica în fișierul: `/etc/iscsi/iscsid.conf` modul de pornire a serviciului, din manual în automatic, astfel:

```
node.startup = automatic
```

Se repornește serviciul de inițiator:

```
service open-iscsi restart
```

Acum putem identifica tinta (în acest caz s-a considerat că are IP-ul 192.168.0.12):

```
iscsiadm -m discovery -t st -p 192.168.0.12
```

Aici:

m: determina modul în care iscsiadm este executat

t: specifica tipul de descoperire folosită.

p: indică adresa IP țintă.

Putem verifica nodurile disponibile pe aceasta:

```
iscsiadm -m node
```

În cazul în care s-a folosit CHAP pentru autentificare, se va introduce comanda:

```
iscsiadm -m node --login
```

dupa care se vor introduce datele de autentificare.

În continuare se folosește comanda `dmesg` care analizează lista de mesaje ale kernelului sistemului de operare. În aceasta se pot observa mesajele legate de partițiile detectate, printre care și cele iSCSI.

```
dmesg | grep sd
```

În rezultatul obținut se poate observa **sdb** ca fiind un nou disc iSCSI.

În cazul în care partiția a fost deja formatata în Windows (de exemplu într-o accesare anterioara a acestei resurse), pasul următor nu mai este necesar!

Formatare partiție în Linux

În continuare se va crea o partiție (cu utilitarul *fdisk*), se va accesa sistemul de fișiere și se va formata partiția iSCSI. În terminal se introduce pentru a crea o partiție pe dispozitivul **sdb** creat în pasul anterior:

```
sudo fdisk /dev/sdb
```

fdisk are următoarele opțiuni (se scrie litera și se apasă *Enter*):

p afișare tabela de partiții

n creare partiție nouă

d ștergere partiție

q ieșire din *fdisk* fără a salva modificările

w ieșire din *fdisk* după ce s-au salvat modificările la tabela de partiții.

După alegerea opțiunilor în dialogul care apare în ordinea: **p** (afișare partiții curente); **n** (creare partiție nouă); **p** (se alege partiție primară, apoi se selectează numărul partiției ca fiind 1 pentru ca este singura prezentă, `First cylinder` ca având valoarea default și `Last cylinder` ca având valoarea default pentru a ocupa tot disk-ul cu partiția); **w** pentru ieșire din fdisk după ce s-au salvat modificările la tabela de partiții.

La terminarea procesului de creare a partiției, este necesară formatarea partiției. Vom alege sistemul de fișiere ext4 care este folosit frecvent în Linux. Utilitarul pentru formatarea în ext4 este numit `mkfs.ext4` și se folosește astfel:

```
sudo mkfs.ext4 /dev/sdb1
```

Alternativ putem folosi orice alt sistem de fișiere prin folosirea utilitarului `mkfs` (de exemplu FAT32 ce poate fi accesat și în Windows se formatează astfel: `sudo mkfs.vfat -F 32 /dev/sdb1`.)

Accesarea partiției `/dev/sdb1` poate fi dificilă dacă avem mai multe partiții pentru că ne putem încurca în valoarea numerică de la sfârșit, deci este recomandat să se facă prin accesarea unui folder cu denumire mai ușor de reținut, în acest caz `/media/partitie` astfel (*Observație:* în cazul în care folderul `/media/partitie` nu este deja creat, se va realiza acest lucru cu `sudo mkdir /media/partitie`):

```
sudo mount /dev/sdb1 /media/partitie
```

În acest mod accesarea `/media/partitie` va deschide partiția iSCSI.

Desfășurarea lucrării

1. Se va studia breviarul teoretic.
2. Se va studia modul în care a fost instalată și configurată aplicația și se vor configura pe calculator ținte iSCSI în sistemul de operare Ubuntu Linux, respectiv inițiatori pentru sistemul de operare Windows.
3. Se vor utiliza aplicațiile de monitorizare a traficului în rețea (Wireshark) pentru a verifica schimbul de date între calculatoare.
4. Se vor studia RFC 1994, RFC 3720 și RFC 3721 .

Anexa 1 – Codul integral al filtrului de animatie „Whirl”

```
1 (define (script-fu-whirl-anim img
2       drawable
3       whirl
4       pinch
5       radius
6       num-frames
7       )
8   (let* ((num-frames (max 1 num-frames))
9         (remaining-frames num-frames)
10        (whirl-shift (/ whirl num-frames))
11        (whirl whirl-shift)
12        (cadre '())
13        (image (car (gimp-image-duplicate img)))
14        (source-layer (car (gimp-image-get-active-layer image))))
15
16     (gimp-image-undo-disable image)
17
18     (while (> remaining-frames 1) ; cat timp mai raman cadre de prelucrat
19       (let* (
20         (whirl-layer (car (gimp-layer-copy source-layer TRUE))) ; copiaza cadrul sursa intr-un nou cadru
21         (layer-name (string-append "Frame " ; denumirea cadrului nou este inregistrata intr-un sir de caractere
22                                (number->string ; introducerea in numele cadrului a unui numar de ordine
23                                (- (+ num-frames 2)
24                                remaining-frames) 10
25                                )
26                                " (first)")) ; introducerea sirului first semnifica sensul rasucirii: crestere
27         )
28         (gimp-layer-set-lock-alpha whirl-layer FALSE) ;
29         (gimp-image-insert-layer image whirl-layer 0 -1) ; urcarea noului cadru in stiva de cadre a imaginii pe prima pozitie (-1)
30         (gimp-item-set-name whirl-layer layer-name) ; botezarea noului cadru folosind sirul de caractere layer-name
31
32         (plug-in-whirl-pinch RUN-NONINTERACTIVE ; apelarea plugin-ului whirl-pinch
33           image ; parametru general: imaginea
34           whirl-layer ; al doilea parametru general: stratul
35           whirl ; parametru special: unghiul de rasucire
36           pinch ; parametru special: strangularea
37           radius) ; parametru special: raza efectului
38
39         (set! remaining-frames (- remaining-frames 1)) ; scaderea numarului de cadre ramase pentru a fi procesate
40         (set! whirl (+ whirl whirl-shift)) ; modificarea unghiului de rasucire
41         (set! cadre (cons (car (gimp-layer-copy whirl-layer TRUE)) cadre)) ; adaugarea cadrului prezent intr-o stiva virt
42       )
43     )
44
45     (set! remaining-frames (- num-frames 1)) ; reinitializarea numarului de cadre ramase pentru a fi procesate
46
47     (while (> remaining-frames 1) ; cat timp mai raman cadre de prelucrat
48       (set! cadre (cdr cadre)) ; scoaterea unui cadru din stiva virtuala de cadre
49       (let* (
50         (whirl-layer (car cadre)) ; citirea primului cadru din stiva
51
52         (layer-name (string-append "Frame " ; denumirea noului cadru va fi inregistrata intr-un sir de caractere
53                                (number->string
54                                remaining-frames
55                                10
56                                )
57                                " (second)")) ; introducerea sirului second semnifica sensul rasucirii: descrest
58       )
59     )
```



```

60 (gimp-layer-set-lock-alpha whirl-layer FALSE) ;
61 (gimp-item-set-name whirl-layer layer-name) ; botezarea noului cadru folosind sirul de caractere layer-name
62 (gimp-image-insert-layer image whirl-layer 0 -1) ; urcarea noului cadru in stiva de cadre a imaginii pe prima po:
63
64 (set! remaining-frames (- remaining-frames 1)) ; scaderea numarului de cadre ramase pentru a fi procesate
65 )
66 )
67
68 (gimp-item-set-name source-layer "Frame 1") ; botezarea cadrului sursa folosind sirul de caractere Frame 1
69 (gimp-image-undo-enable image)
70 (gimp-display-new image) ; afiseaza noua imagine
71
72 )
73 )
74
75 (script-fu-register "script-fu-whirl-anim"
76   "whirl"
77   "Creates a multi-layer image with a whirling effect"
78   "Adrian Iordachescu <adi_iord@yahoo.com>"
79   "Adrian Iordachescu"
80   "2014/12/01"
81   "RGB* GRAY*"
82   SF-IMAGE      "Image" 0
83   SF-DRAWABLE    "Drawable" 0
84   SF-ADJUSTMENT "Whirl"      '(10 0 720 1 10 2 0)
85   SF-ADJUSTMENT "Pinch"      '(0 -1 1 0.1 1 3 0)
86   SF-ADJUSTMENT "Radius"     '(1 0 2 0.1 1 3 0)
87   SF-ADJUSTMENT "Number of frames" '(6 1 512 1 10 0 1)
88 )
89
90 (script-fu-menu-register "script-fu-whirl-anim"
91   "<Image>/Filters/Animation/Adrian")

```

Anexa 2 – Configurarea unui ruter Linux

Vom configura în continuare un ruter care folosește sistemul de operare Linux. Aceste rutere sunt deseori folosite de firmele mici deoarece costă foarte puțin (de obicei se utilizează calculatoare obișnuite).

Sistemul de operare Linux are multe distribuții care prezintă diferențe minimale în ceea ce privește configurarea interfeței de rețea.

În exemplele din aceasta carte se folosesc distribuțiile bazate pe Debian: Knoppix și Ubuntu. Dacă nu se folosesc sisteme bazate pe Debian, se va consulta documentația pentru aceste versiuni de Linux.

În sistemul Linux adaptorul de rețea are denumiri speciale în funcție de tipul conexiunii. Avem următoarele denumiri folosite pentru identificarea adaptorului de rețea:

Loopback – prescurtarea **lo**.

Ethernet – eth0, eth1, lan0, lan1.

Wireless – wlan0, wlan1, ...

Token Ring – tr0, tr1.

Seriale – ppp0, ppp1 (ppp – point to point protocol – folosit pentru comunicațiile prin liniile analogice de telefon (de ex. modem-uri))

Descarcarea și pornirea sistemului de operare

Pentru exemplificare vom folosi o distribuție Live CD: KNOPPIX. Aceasta poate fi descărcată sub forma de imagine ISO de la adresa: <http://www.knoppix.org/>

După descărcare, imaginea se scrie pe un CD sau DVD (în funcție de versiunea aleasă) după care se pornește calculatorul de pe acest mediu de stocare.

Alternativ se poate instala ca un sistem de operare virtual într-un sistem de virtualizare precum VirtualBox. În acest caz:

- nu mai este necesară scrierea imaginii ISO descărcate
- orice modificare nu va putea afecta sistemul de fișiere al mașinii reale
- avem două sisteme de operare în care putem lucra în paralel (cel real și cel virtual)

Dezavantajul este că sistemul de operare virtual va avea o viteză mai mică decât în cazul în care rulează direct pe hardware.

Instalarea și configurarea mașinii virtuale este prezentată în această carte.

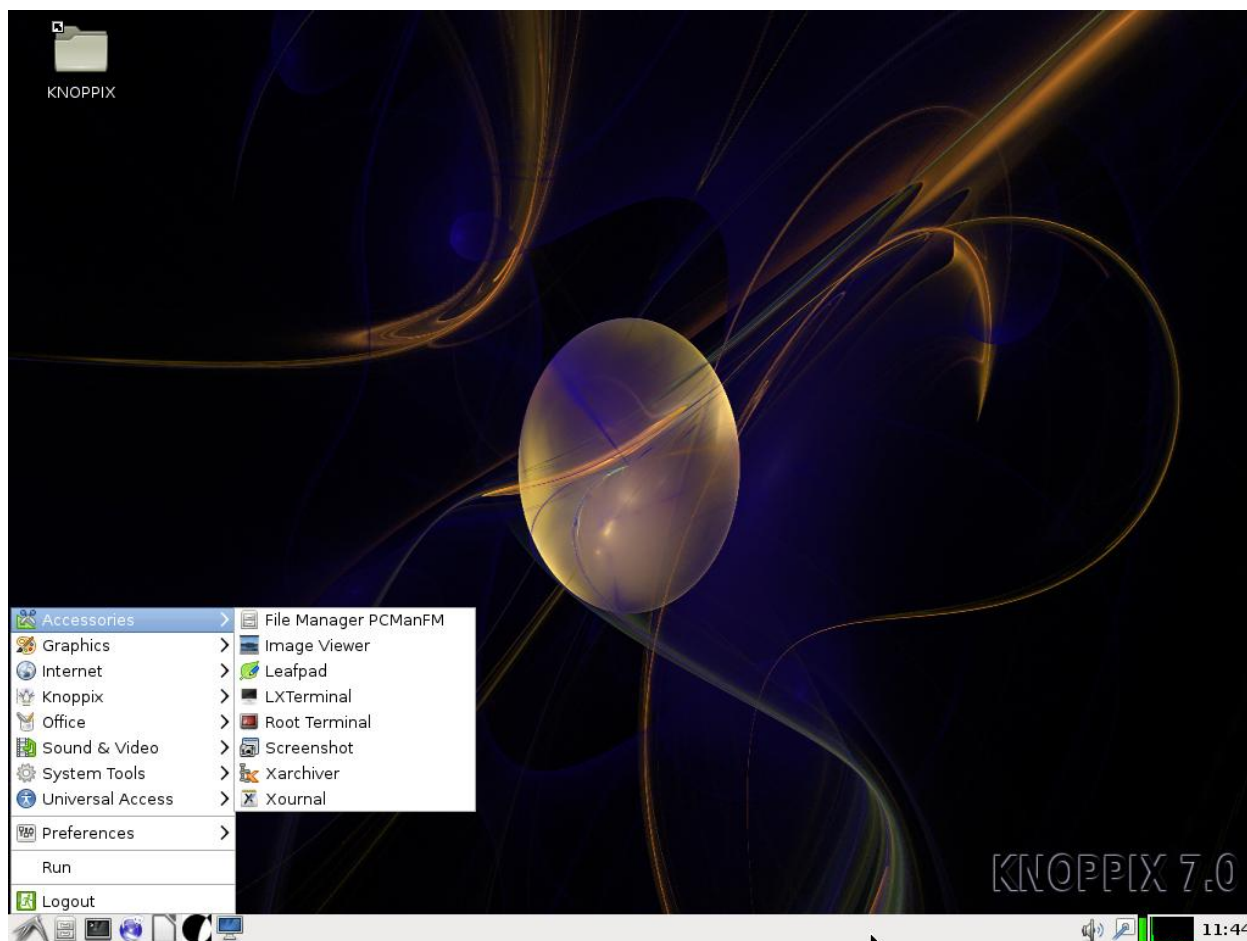


Figura 1 Interfața grafică a Knoppix Linux cu meniul de start

În cazul în care s-a descărcat o versiune de server (de exemplu Ubuntu Server), vom avea la pornirea sistemului de operare doar un terminal în care să putem scrie comenzi.

Sistemul de fișiere Linux

Linux are un sistem de fișiere arborescent care pornește de la o rădăcină unică, notată cu /. Separatorul între componentele căii către de fișier este caracterul /,

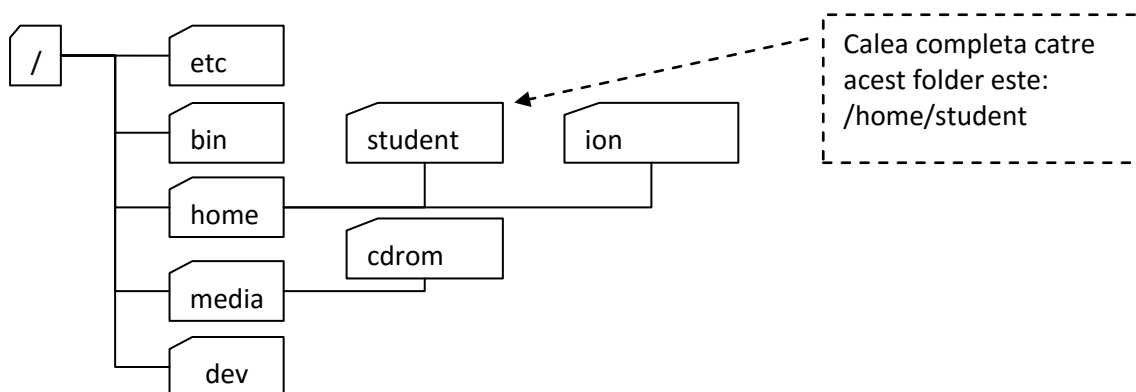


Figura 2 Structura arborescentă a sistemului de fișiere în Linux

De exemplu calea următoare arată că suntem în subfolderul *student* al folderului *home*, iar *home* este un folder în rădăcina unică */*:

`/home/student`

Câteva directoare/foldere importante sunt:

Directory	Content
<code>/bin</code>	Conține programe partajate de utilizatori, sistemul de operare și de administrator.
<code>/boot</code>	Conține fișierele citite la încărcarea sistemului de operare.
<code>/dev</code>	Conține referințe către toate dispozitivele periferice care sunt reprezentate drept fișiere cu proprietatea specială.
<code>/etc</code>	Conține fișierele de configurare ale sistemului
<code>/home</code>	Directorul home pentru utilizatorii obișnuiți
<code>/initrd</code>	Conține informații despre procesul de pornire (boot) al sistemului. Util pentru diagnosticarea problemelor de pornire ale sistemului
<code>/lib</code>	Diferite librării utilizate și utilizatori și de sistemul de operare.
<code>/lost+found</code>	Locul în care sunt salvate fișierele care sunt găsite de sistem în urma unei opriri incorecte.
<code>/mnt</code>	Punctul de accesare pentru sistemele de fișiere externe (CD-ROM, etc.)
<code>/net</code>	Punctul de accesare pentru sistemele de fișiere externe
<code>/root</code>	Directorul home pentru administrator (root)
<code>/sbin</code>	Programe utilizate doar de sistem și de administratorul sistemului
<code>/tmp</code>	Director pentru fișiere temporare.
<code>/usr</code>	Programe și librării pentru utilizatori.
<code>/var</code>	Stocare pentru toate variabilele și fișierele temporare

Execuția comenzilor

Comenzile prezentate în continuare se scriu într-un *terminal* (numit uneori Shell sau consolă) care este o fereastră text specială în care utilizatorul poate scrie comenzi. Pentru a porni un terminal în interfața grafică Linux, accesați meniul *Application* -> *Accessories* -> *LXTerminal*. Va apărea o fereastră în care puteți executa comenzi.

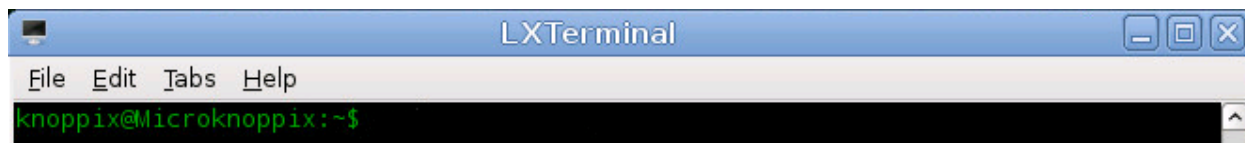


Figura 3 Prompterul implicit din Knoppix

Prompterul afișat are forma:

knoppix@Microknoppix:~\$

Acesta se compune din trei părți separate de @, : și \$

1. Zona dinainte de @ indică utilizatorul cu care sunteți logat în momentul respectiv. Implicit este utilizatorul Knoppix. Acest utilizator nu are drepturi depline pentru a putea configura setările de rețea.
2. Zona între @ și :, indică numele calculatorului pe care lucrați (Microknoppix în exemplu).
3. Zona de după , indică directorul curent în care lucrați. În exemplu este ~ care este o scurtătură către directorul Home al utilizatorului. Acesta este echivalent cu /home/student/. Pe măsură ce navigați prin directoare, a treia parte a prompterului se schimbă.

O comandă se execută tastând textul corespunzător comenzii, urmat de ENTER. Rezultatul execuției comenzii se va afișa în terminal.

Câteva comenzi de bază sunt:

Comanda	Explicație
<code>mkdir nume_director</code>	Creează un director/folder cu numele dat.
<code>cd [nume_director]</code>	Schimbă directorul curent, trecând în cel specificat ca parametru. Dacă vrem să ieșim în directorul părinte se scrie cd ..
<code>ls [cale]</code>	Listează fișierele din directorul dat. Ex: ls /bin
<code>cat fișier</code>	Afișează conținutul fișierului
<code>cp sursa destinatie</code>	Face o copie a lui <i>sursa</i> , cu numele <i>destinatie</i>
<code>mv sursa destinatie</code>	Redenumeste <i>sursa</i> , cu numele <i>destinatie</i>
<code>rm nume_fisier</code>	Șterge fișierul specificat

Fisiere importante în Linux pentru configurarea adaptorului de rețea

Setările legate de configurarea adaptorului de rețea sau a rutării sunt salvate în fișiere de configurare care sunt citite la repornirea sistemului, astfel încât modificările realizate să devină permanente. Câteva fișiere legate de configurarea rețelei sunt:

/etc/resolv.conf - fișierul de traducerea a numelor de calculatoare în adrese IP. Acesta permite Linux-ului să cunoască ce server DNS va soluționa numele de domenii în adrese IP. Dacă se folosește DHCP (protocol folosit pentru configurarea dinamică a calculatoarelor) aceste informații sunt trimise automat de către ISP (Internet Service Provider). Dacă se folosește adresarea statică (adresele sunt configurate manual) setările se vor efectua de către administratorul calculatorului. În acest exemplu noi vom face configurarea manuală.

/etc/hosts - translatează numele de calculatoare în adrese IP la nivel local.

/etc/modules.conf - acest fișier arată modulele care vor fi încărcate de sistemul de operare. În majoritatea cazurilor dispozitivele de rețea sunt încărcate drept module.

Configurarea setărilor de rețea

Configurarea setărilor de rețea se face în modul root (administrator). Pentru a intra în acest mod se pornește un *terminal* și se scrie comanda: **sudo -i** (sau: **su -**, dacă prima comandă nu este acceptată)

Dacă există o parolă pentru utilizatorul root, aceasta este solicitată. Promptul va indica modificarea rolului utilizatorului curent și deveni de exemplu:

root@Microknoppix:~#

```
LXTerminal
File Edit Tabs Help
knoppix@Microknoppix:~$ sudo -i
root@Microknoppix:~#
```

Figura 4 Trecerea in modul de root pentru a avea drepturi de configurare

Atenție! Sistemele de operare moderne încearcă configurarea automată a adreselor IP ale unui sistem pentru că acesta să fie conectat la internet automat. În cazul în care nu avem în rețea un server DHCP de adrese IP configurat sau dorim configurarea manuală a adreselor IP, este necesară oprirea serviciului de configurarea automată (Dacă nu se oprește acest serviciu, la câteva minute el reîncearcă configurarea adreselor IP și dacă eșuează, șterge configurația curentă introdusă manual):

service network-manager stop

În urma execuției acestei comenzi vom primi un mesaj că serviciul a fost oprit sau că este deja oprit. În cazul în care nu primim aceste mesaje, ci un mesaj de eroare, trebuie să verificăm dacă suntem logați ca utilizatorul root sau dacă am scris comanda corect.

Configurarea manuală a unui ruter cu sistemul de operare Linux

Presupunem configurația din figura în care **sistemul Linux cu rol de ruter este PC1** și are două adaptoare de rețea, unul conectat la internet (eth0), altul conectat la rețeaua locală (eth1). Calculatoarele PC2 și PC3 pot fi sisteme Linux sau Windows. Internetul sosește la ruterul Linux de la IP-ul 192.168.94.1. Adresele IP vor fi configurate în exemplu ca în figură.

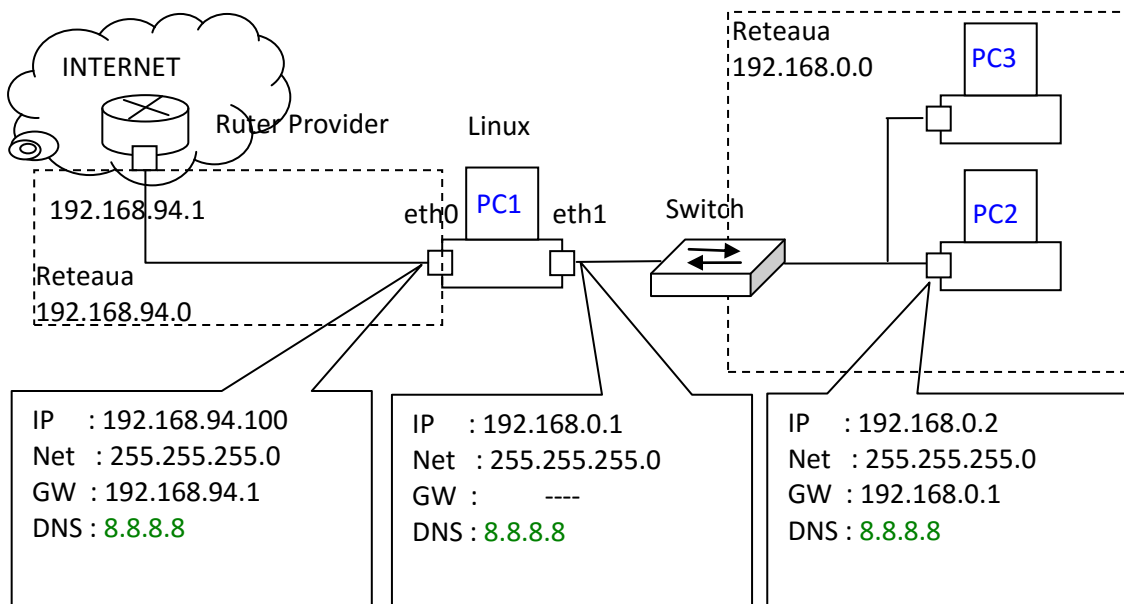


Figura 5 Structura rețelei din exemplu are două rețele

Etape pentru configurarea ruterului Linux:

1. Pe sistemul ruter PC1 se execută **ifconfig** (interface configuration) pentru a lista plăcile de rețea existente. Este posibil să se afișeze doar placa de rețea *lo* deoarece celelalte adaptoare de rețea nu sunt încă configurate.
2. Pe sistemul ruter PC1 se configurează, cele două plăci de rețea: **eth0** și **eth1** (în unele versiuni de Linux: **lan0** și **lan1**) pentru sistemul care asigură rutarea (aici PC1). În terminal se scrie:

```
ifconfig eth0 192.168.94.100 netmask 255.255.255.0
```

```
ifconfig eth1 192.168.0.1 netmask 255.255.255.0
```

3. Pe sistemul PC1 configurarea DNS se poate face în mai multe moduri alternative în funcție de versiunea de Linux utilizată (în exemple vom folosi serverul DNS oferit public de Google de la adresa IP **8.8.8.8**. Modificând direct în fișierul: **/etc/resolv.conf** (sau dacă în acest fișier găsim un mesaj că modificările de aici sunt generate automat, prin crearea unui fișier **/etc/resolvconf/resolv.conf.d/tail** și modificarea în acesta), adăugând la final pe o linie nouă:

```
nameserver 8.8.8.8
```

4. Pe sistemul ruter PC1 se configurează calea implicită (default gateway) către care sunt trimise pachetele care nu au destinația în rețeaua locală. Configurarea cii implicite nu este necesară pentru accesarea calculatoarelor din rețeaua locală însă este necesară dacă dorim să accesăm calculatoare din alte rețele. Se scrie în terminal:

```
route add default gw 192.168.94.1
```

În exemplul nostru **gw** (gateway) este placa conectată la internet, eth0.

Observație: dacă am introdus greșit o adresă în acest pas dar am executat însă comanda, pentru a adăuga o adresă nouă este necesară eliminarea celei vechi cu:

```
route del default gw 192.168.94.1
```

Dacă nu se face acest pas Linux va folosi ambele adrese însă în ordinea introducerii lor, deci a doua adresă, chiar dacă este prezentă, nu va fi folosită.

- În acest moment **ruterul** are acces la internet. Se testează pe PC1 prin accesare în browser a unui site internet. Dacă nu are conectivitate la internet se verifică etapele parcurse.

5. Pentru sistemele PC2, PC3 conectate la interfața eth1 a ruterului Linux se configurează: adresa pentru placa de rețea (în cazul nostru vor fi în rețeaua 192.168.0.0, de exemplu: 192.168.0.2), adresa gateway (IP ruter de pe interfața spre client, aici 192.168.0.1) și serverul DNS (8.8.8.8).

- În acest moment clientul poate contacta ruterul Linux, nu și internetul. Se testează în terminal pe PC2 sau PC3 prin trimitere ping către ruter (ping 192.168.94.1). Dacă nu primește răspuns (cu timp de ~1ms) se verifică etapele parcurse.

6. Pe sistemul ruter PC1 se activează funcția de rutare. Ea are două componente:

- Fișierul `/proc/sys/net/ipv4/ip_forward` conține valoarea care controlează procesul de rutare (implicit 0). Pentru a verifica conținutul acestui fișier se scrie: `cat /proc/sys/net/ipv4/ip_forward`

Pentru a activa rutarea trebuie să se scrie în fișier 1. Se scrie în terminal:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

- Pentru activarea funcției de traducere a adreselor între interfețele eth0 și eth1, se folosește funcția **iptables** în care `eth0` este adaptorul conectat la internet. Se scrie în terminal:

```
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

- În acest moment sistemul Linux PC1 are configurată rutarea între interfețele eth0 și eth1.
- În acest moment **clientul** are acces la internet. Se testează pe PC2 sau PC3 prin accesare în browser a unui site internet. Dacă nu are conectivitate la internet se verifică etapele parcurse.

Observatii:

Nu uitați să folosiți frecvent utilitarele disponibile pe sistemul Linux (executate în terminal în modul root) pentru a testa configurația realizată parțial. Este mult mai ușor să detectați o eroare și să o corectați imediat decât la finalul tuturor pașilor.

- **ping <IP>** pentru testare prezenta altui calculator în rețea: Ex. ping 192.168.94.1
- **route** pentru afișare tabelă de rutare din care se pot observa IP gateway (dacă este configurat) și adrese rețele configurate pe interfețele de rețea;
- **iptables -L** și **iptables -t nat -L** pentru a inspecta regulile de filtrare a traficului și activarea rutării;
- **iptraf** pentru a monitoriza traficul din rețea.
- **arp** pentru a vedea tabela ARP de asociere IP - MAC

Bibliografie

- [1] GIMP GNU Image Manipulation Program, User Manual, <http://docs.gimp.org>
- [2] John Russ, "The Image Processing Handbook" (Sixth Edition), CRC Press, 2011
- [3] Michael Freeman, "Școala de fotografie. Editarea digitală", Editura Litera, 2015
- [4] Carl Wenczek, "Digital images: GIMP – further techniques", University of Oxford, 2015
- [5] Andrew S. Tanenbaum, "Rețele de calculatoare", Ediția a patra, 2003 Editura Byblos;
- [6] Adrian Munteanu, Valerica Greavu Serban, "Rețele locale de calculatoare. Proiectare și administrare", editia a 2-a, Ed. Polirom, 2006
- [7] Dragos Acostachioaie, "Securitatea sistemelor Linux", Ed. Polirom, 2003
- [8] William Stallings, "Data and Computer Communications", 8-th Edition, Pearson Prentice Hall, ISBN: 0-13-243310-9, 2007
- [9] Rughiniș, R., Deaconescu, R., Ciorba, A., Doinea, B., "Rețele locale", București, Editura Printech ISBN: 978-606-521-092-9, 2009.
- [10] Tatiana Radulescu, Henri George Coanda, „QoS în rețelele IP multimedia”, Editura Albastra, Cluj-Napoca, ISBN: 973-650-219-4, 2007.
- [11] Lucian Ioan, Graziela Niculescu, "Calitatea servicii în rețelele cu comutație de pachete", ISBN: 978-973-755-319-5, Editura: MatrixRom
- [12] Lucian Ioan, Graziela Niculescu, "Modelare și evaluări de performanță în telecomunicații", ISBN: 978-973-755-396-6, Editura: MatrixRom

Resurse online:

- [1] Oracle VM VirtualBox® User Manual, Copyright © 2004-2015 Oracle Corporation, disponibil la: <https://www.virtualbox.org/manual/UserManual.html>
- [2] Hyper-V Getting Started Guide, Copyright © Microsoft, disponibil la: [https://technet.microsoft.com/en-us/library/cc732470\(v=ws.10\).aspx](https://technet.microsoft.com/en-us/library/cc732470(v=ws.10).aspx)
- [3] Steps to Remotely Manage Hyper-V Server 2012 Core, September 30, 2013 by Chris Davis, disponibil la: <http://pc-addicts.com/12-steps-to-remotely-manage-hyper-v-server-2012-core/>

- [4] iSCSI Initiator, Ubuntu 14.04 Server Guide, disponibil la: <https://help.ubuntu.com/lts/serverguide/iscsi-initiator.html>
- [5] Configure Ubuntu To Serve As An iSCSI Target, May 9, 2012, © luis ventura, disponibil la: <https://linhost.info/2012/05/configure-ubuntu-to-serve-as-an-iscsi-target/>
- [6] Linux Advanced Routing & Traffic Control HOWTO, © Bert Hubert, disponibil la: <http://lartc.org/howto/>
- [7] Port Forwarding Using iptables, Eric Zhiqiang Ma, Aug 29, 2013, disponibil la: <http://www.systutorials.com/816/port-forwarding-using-iptables/>
- [8] Hypertext Transfer Protocol -- HTTP/1.1, RFC 2616 Fielding, et al., disponibil la: <http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>
- [9] Introduction to Linux, A Hands on Guide, Machtelt Garrels, disponibil la: <http://tldp.org/LDP/intro-linux/html/index.html>