გ. ღვინეფაძე

Windows

ქსელური ოპერაციული სისტემები

I ნაწილი

“ტექნიკური უნივერსიტეტი”

უაკ 681.3.06

განხილულია ქსელური ოპერაციული სისტემა Windows 2000. მოცემულია აგრეთვე მისი წინამორბედი ოპერაციული სისტემის Windows NT 4.0-ის მოკლე აღწერა.

განკუთვნილია ინფორმატიკის 22.02, 22.01, 0719.08 სპეციალობათა შემსწავლელი სტუდენტებისა და ამ საკითხით დაინტერესებული პირებისთვის.

რეცენზენტები: პროფ. ო. ნატროშვილი,
ასოც. პროფ. თ. სუხიაშვილი

ISBN 99940-48-22-8

Windows 2000

2000 წელს კომპანია "მაიკროსოფტმა" კომპიუტერულ სამყაროს წარუდგინა ქსელური ოპერაციული სისტემა **Windows NT**-ის მნიშვნელოვნად გაუმჯობესებული ვერსია – **Windows 2000 Server**. საინტერესოა, რომ ამ სისტემის შექმნაზე მუშაობდა 2000-ზე მეტი პროგრამისტი, რომელთა მიერ დაწერილმა კოდმა დაახლოებით 65 მლნ-ზე მეტი სტრიქონი შეადგინა.

სპეციალისტები თვლიან, რომ **Windows 2000** წინამორბედთან შედარებით გამოირჩევა მეტი საიმედოობითა და სტაბილურობით. ახალი ინსტრუმენტების დამატებისა და არსებულთა საფუძვლიანი გადამუშავების შედეგად ოპერაციული სისტემა უფრო სრულყოფილი და ადვილად სამართავი გახდა.

სისტემის სტაბილურობაზე საუბრისას მხედველობაში აქვთ მისი უნარი, გაცილებით იშვიათად შეფერხდეს მუშაობის პროცესში *("ჩამოეკიდოს")* და ასევე ძალზე იშვიათად მოითხოვოს გადატვირთვა *(5 სხვადასხვა სიტუაცია, მაშინ, როცა NT-თვის ეს რიცხვი 75-ს აღწევდა).*

Windows 2000 აგრძელებს ქსელის ორგანიზაციისადმი კლიენტ-სერვერული მიდგომის ტრადიციას *(ეს მოდელი 90-იან წლებში შემუშავდა).* იგი ასევე იყენებს ინტერნეტში განაწილებული გამოთვლების მოდელს, მაგრამ ორივე ასპექტით ბევრი საკითხი ახლებურად, უფრო მაღალ დონეზე გადაწყდა, რამაც მოითხოვა **Windows NT 4.0**-საგან განსხვავებული კონტროლერის ტიპებისა და დომენური სტრუქტურის გამოყენება. ამასთან, გაუმჯობესდა უსაფრთხოების მონაცემთა ბაზების სტრუქტურაც ე.წ. *კატალოგების სამსახურის* - **Active Directory**-ის შემოღების გზით. აღნიშნული სამსახური ქსელის თითოეულ მომხმარებელს გამოუყოფს მხოლოდ საჭირო რესურსებს.

Windows 2000-ში **COM+** სახელით ჩამატებული კომპონენტ-ობიექტების განაწილებული მოდელის ტექნოლოგია უზრუნველყოფს კლიენტების მიერ ქსელში *გამოყენებებით* სარგებლობას, მათ გავრცელებას, ცენტრალიზებულად მართვას და განახლებას **(COM - Component Object Model).**

Windows 2000-ში ჩართულია **Web**-სერვერის ახალი ვერსიაც – **IIS 5.0 (Internet Information Server 5.0).** ახალი შესაძლებლობები *(მაგალითად, **Web**–კლიენტებისთვის რესურსებზე კვოტის დაწესება)* **Web**-სერვერს იცავს გადატვირთვისაგან, ხოლო პროვაიდერს

უადვილებს მომხმარებლებთან ურთიერთობას *(კერძოდ, კლიენტების მიერ რესურსებით სარგებლობის ფასის გამოთვლას).*

დაბოლოს, **Windows 2000**-ის ოჯახში, **Windows 2000 Server**-ის გარდა, შედის სერვერული ოპერაციული სისტემის უფრო მძლავრი ვერსიებიც:

Windows 2000 Advanced Server

და

Windows 2000 Datacenter Server.

ამ ოჯახის წევრია აგრეთვე **Windows 2000 Professional.**

სერვერული პროგრამები ერთმანეთისგან ძირითადად მწარმოებლურობის დონით განსხვავდება, რაც განპირობებულია იმ ფაქტორით, რომ ისინი გამოიყენება სხვადასხვა აპარატურული შესაძლებლობების მქონე კომპლექსებისთვის *(მართვადი ოპერატიული მეხსიერების მოცულობა, მიკროპროცესორების რიცხვი და სხვ).*

სისტემის არქიტექტურა და ჩატვირთვის პროცესი

Windows 2000 Server "მაიკროსოფტმა" **Windows NT**-ის ბაზაზე შექმნა. ოპერაციული სისტემა შეიცავს შედარებით მცირე ზომის ბირთვს, რომელიც ცენტრალური პროცესორის მუშაობას წარმართავს. თითოეული გამოყენების მიერ ინიციალიზებული პროცესები სრულდება მეხსიერების საკუთარ, დაცულ უბანში. ამრიგად, შესრულების პროცესში ერთ-ერთი მათგანის შეფერხება ხელს არ უშლის სხვა პროგრამებს მუშაობაში.

ოპერაციული სისტემა მოდულურ პრინციპზეა აგებული, ამიტომ თითოეული მოდულის შეცვლა-გაუმჯობესება ავტონომიურად ხორციელდება *(მთელი ოპერაციული სისტემის გადაწერის გარეშე).* მოდულებს შორის ურთიერთობა კი წარიმართება სისტემური გამოძახებების მეშვეობით.

შევნიშნოთ, რომ "მაიკროსოფტმა", მართალია, მთელი ოპერაციული სისტემის კოდი არ გახსნა, მაგრამ გამოაქვეყნა მის მოდულებს შორის ურთიერთობის საშუალებების აღწერა *გამოყენებების პროგრამულ ინტერფეისთა სტანდარტული **API** კრებულის* სახით.

Windows 2000 მუშაობს ერთმანეთისაგან იზოლირებულ 2 რეჟიმში:

- მომხმარებლის რეჟიმი;
- ბირთვის რეჟიმი.

მომხმარებლის რეჟიმი

მომხმარებლის რეჟიმი ოპერაციული სისტემის მუშაობის ის რეჟიმია, როცა მომხმარებელი ამყარებს ოპერაციულ სისტემასთან ურთიერთობას.

ეს პროცესი იზოლირებულია პერიფერიულ მოწყობილობებთან დაკავშირებული შეტანა-გამოტანისა და სხვა სისტემური პროცესებისაგან.

მომხმარებლის რეჟიმი მოიცავს ქვესისტემებს, რომელთა დანიშნულებაა სხვადასხვა ოპერაციული სისტემებისთვის გამიზნული პროგრამების შესრულება **Windows 2000 Server**-ის მიერ მართვის გზით *(ანუ ამ პროგრამების ხელახალი კომპილაციის გარეშე).*

ამ ქვესისტემების მაგალითებია: **POSIX, Win32, OS/2…** ამრიგად, პროგრამები უშუალოდ არ მიმართავენ პერიფერიულ მოწყობილობებს. მათ შორის იმყოფება ოპერაციული სისტემის ბირთვში არსებული ***აპარატურული აბსტრაქციების დონე*** **(HAL),** რომელიც შუამავლის როლს ასრულებს.

საკითხის ასეთი გადაწყვეტა **Windows 2000**-ს ანიჭებს კონკრეტული აპარატურული პლატფორმისაგან *(იგულისხმება მიკროპროცესორების ტიპები)* დამოუკიდებლობის თვისებას.

როცა გამოყენების მიერ მოითხოვება მისთვის დანიშნულ ოპერატიულ მეხსიერებაზე მეტი სივრცე, ადგილის გამოსათავისუფლებლად ხისტ დისკოზე განლაგებულ სპეციალურ "გადაქაჩვის" ფაილში ოპერაციული სისტემის მიერ გადაიწერება ის ინსტრუქციები, რომლებიც არ გამოიყენებოდა ბოლო პერიოდში. ამრიგად, ოპერატიული მეხსიერებისთვის იქმნება წარმოსახვითი ანუ ვირტუალური უბანი. ვირტუალური მეხსიერება გამოიყენება დამატებითი პროგრამების ჩამოტვირთვისთვისაც. ცხადია, ამ შემთხვევაში პროგრამები გაცილებით ნელა სრულდება, ვიდრე მაშინ, როცა ყველა პროგრამა "სუფთა" ოპერატიულ მეხსიერებაში განთავსდება.

რაც შეეხება პრიორიტეტებს, ბუნებრივია, რომ მომხმარებლის პროგრამების მიერ ინიცირებულ პროცესებს გაცილებით ნაკლები პრიორიტეტი გააჩნიათ ბირთვის რეჟიმში მიმდინარე პროცესებთან შედარებით.

ბირთვის რეჟიმი

ამ რეჟიმს პრიორიტეტულ რეჟიმსაც უწოდებენ. მასში ის მოდულები მუშაობენ, რომლებიც განაგებენ აპარატურულ საშუალებებთან შეღწევისა და მონაცემების გადაგზავნა-გადმოგზავნის პროცესებს. უშუალოდ თითოეული მოწყობილობის მართვისათვის კი განკუთვნილია საკუთარი პროგრამა-დრაივერი.

თვით ბირთვი წარმოგვიდგება როგორც ცენტრალური პროცესორის *(მიკროპროცესორის)* მიერ შესრულებადი ბრძანებების კრებული. რადგანაც ინფორმაციის შეტანა-გამოტანაზე ყოველი სიგნალი ცენტრალურ პროცესორს მიეწოდება და მის მიერ დამუშავდება, ანდა სხვა პროცესორში გადაიგზავნება, ბირთვი გვევლინება როგორც უშუალო შემსრულებლის, ასევე მთელი სისტემის მუშაობის კოორ-დინატორის როლში.

ბირთვის რეჟიმის ცალკეული ასპექტებისთვის კი განკუთვნილია სხვადასხვა სახის მენეჯერების *(დისპეტჩერების)* მოდულები. ამ მოდულების მეშვეობით წარიმართება ვირტუალურ მეხსიერებასა და სხვა ობიექტებთან მუშაობა, კონტროლდება პროცესები და მათი ურთიერთქმედება, ხდება COM+ მოდელის ფუნქციონირების უზრუნველყოფა.

ბირთვის რეჟიმის ნაწილია აგრეთვე ჩვენ მიერ ზემოთ ნახსენები აპარატურული აბსტრაქტული დონე - HAL, რომელიც აწესრიგებს პერიფერიულ მოწყობილობებს შორის ურთიერთობას.

ბირთვის რეჟიმის ზედა დონეა **Windows 2000 Executive** ***(Windows 2000-ის შემსრულებელი დონე).*** მასთან კავშირშია ზემოთ აღნიშნული პროცესები. იგი განაგებს უსაფრთხოების სისტემის მუშაობასაც.

ჩამოვთვალოთ ბირთვის რეჟიმში მომუშავე მენეჯერები *(სხვაგვარად, სამსახურები)* და მოკლედ აღვწეროთ მათი დანიშნულება:

- **I/O Manager** - შეტანა-გამოტანის მენეჯერი;
- **Security Reference Monitor** – უსაფრთხოების სისტემის მენეჯერი;
- **IPC Manager** – პროცესებსშორისი ურთიერთობების მენეჯერი;
- **Virtual Memory Manager (VMM)** – ვირტუალური მეხსიერების მენეჯერი;
- **Process Manager** – პროცესების მენეჯერი;

- **Plug&Play Manager – Plug&Play** მოწყობილობების მენეჯერი;
- **Power Manager** – ელექტროკვების მენეჯერი;
- **Windows Manager** – ფანჯრების მენეჯერი;
- **Object Manager** - ობიექტების მენეჯერი;
- მიკრობირთვის მოდული;
- **HAL** – აპარატურული აბსტრაქციების დონე.

Windows 2000-ის არქიტექტურის სხვა თავისებურებანი

მონაცემების მრავალნაკადიანობა

პროცესორის მიერ შესრულებადი პრაქტიკულად ყველა ბრძანება რამდენიმე ეტაპად სრულდება: თითოეული მათგანი იყოფა ამოცანებად, ხოლო თითოეული ამოცანისთვის კი იქმნება ნაკადი, რომელიც სრულდება აღნიშნული ბრძანების შემცველი პროგრამისთვის გამოყოფილი მეხსიერების უბანში. თითოეული ნაკადი იდენტიფიცირდება ნომრით.

პროცესორს შეუძლია ერთდროულად შეასრულოს მრავალი ნაკადი. თითოეულ ნაკადთან კავშირდება ორი სტეკი: ცალკე მომხმარებლისა და ცალკე ბირთვის რეჟიმებში მუშაობისთვის. მას ინფორმაციის შესანახად აქვს მეხსიერებაც, რომელიც საჭიროა სხვადასხვა ქვესისტემებსა და ბიბლიოთეკებთან ურთიერთობისთვის (**DLL – Dynamic Link Library** – დინამიკურად მიერთებადი ბიბლიოთეკა; **runtime library** - შესრულების ბიბლიოთეკა).

მრავალამოცანიანობა

ნაკადთან მუშაობა ჩერდება, როცა იგი ელოდება პერი-ფერიულ მოწყობილობასთან მონაცემების გაცვლა-გამოცვლას. ცენტრალური პროცესორი რომ არ მოცდეს, იგი გადაერთვება სხვა ამოცანის შესრულებაზე. ზოგჯერ ცენტრალური პროცესორი ნაკადთან მუშაობას თავისი ინიციატივითაც შეაჩერებს ხოლმე, როცა საჭირო ხდება უფრო მაღალი პრიორიტეტის მქონე ამოცანის მომსახურება. ამ დროს კომპიუტერის რესურსების გამოსათავისუფლებლად ნაკადის შესახებ ინფორმაცია შესაძლოა დროებით გარე მეხსიერებაში გადაიგზავნოს.

მიუხედავად იმისა, რომ დროის მოცემულ მომენტში პროცესორი მხოლოდ ერთ ამოცანასთან მუშაობს, ზემოთ აღწერილ ქმედებათა გამო იქმნება ილუზია, რომ ამოცანების შესრულება დროში პარალელურად მიმდინარეობს.

სწორედ, ეს გახლავთ მრავალამოცანიანობის არსი.

მრავალპროცესორული დამუშავება

Windows 2000-ს შეუძლია მართლაც პარალელურად დაამუშაოს რამდენიმე პროცესი. მაგრამ ეს მოხდება იმ შემთხვევაში, თუ სისტემაში გამოიყენება რამდენიმე პროცესორი, რომლებზეც მგეგმავის მიერ ხდება ნაკადების გადანაწილება.

მეხსიერება

Windows 2000 მუშაობს 32-თანრიგიან სამისამართო უბანთან, რომლის მეშვეობით ხდება 4-გიგაბაიტიანი ოპერატიული სივრცის დამისამართება *(მათ შორის ვირტუალურისაც).*

Windows-ის მომავალ ვერსიებში ამ მიზნით გათვალისწინებულია 64-თანრიგიანი სისტემის გამოყენება. ასეთი გადაწყვეტისას ოპერატიული სივრცე, რომელიც შეიძლება დამისამართდეს, ფაქტობრივად, ამოუწურავი იქნება.

ოპერაციული სისტემის ჩატვირთვის პროცესი Intel-ის პროცესორებზე

აღნიშნული პროცესი მოიცავს 5 სტადიას:

1. *წინასწარი ჩატვირთვა.* ამ სტადიაზე ხდება აპარატურული საშუალებების შემოწმება და საჭირო ბრძანებების ჩატვირთვა. სტადია მთავრდება მეხსიერებაში **NTLDR** ჩამტვირთავის გადმოწერით.
2. *ჩატვირთვის პროცესი.* განისაზღვრება აპარატურული საშუალებების კონფიგურაცია. მეხსიერებაში ჩაიტვირთება საჭირო დრაივერები.
3. *ბირთვის ჩატვირთვა.* **NTLDR** ჩამტვირთავი მეხსიერებაში გადმოწერს ოპერაციული სისტემის ბირთვს, ჩაიტვირთება აგრეთვე აპარატურული აბსტრაქციების დონე და ხდება აღჭურვილობის პროფილის არჩევა.

4. *ბირთვის ინიციალიზაცია*. ეს პროცესი წარიმართება **NTOSKRNL.EXE** ფაილის მეშვეობით. ბირთვს, თავის მხრივ, რეესტრში შეაქვს აპარატურული საშუალებების შესახებ ინფორმაცია. ამასთან, იგი მეხსიერებაში გადმოწერს ქვედა დონის დრაივერებს. ამ მომენტში მთავრდება ჩატვირთვის პროცესის "ტექსტური" ნაწილი და იწყება "გრაფიკული" ნაწილი.
5. *სისტემაში შესვლა*. ხდება მომხმარებლის აუთენტიფიცირება და სამსახურების ჩატვირთვა. მომხმარებელს საშუალება ეძლევა, ისარგებლოს მისთვის ნებადართული რესურსებით.

გამორიცხული არ გახლავთ, რომ ჩატვირთვის პროცესისათვის საჭირო ფაილები ხისტ დისკოზე დაზიანდეს. მაშინ უნდა ვისარგებლოთ წინასწარ მომზადებული დისკეტით, რომელზეც კოპირებული გვექნება ეს ფაილები **(NTLDR, NTDETECT.COM, BOOT.INI).**

უფრო დაწვრილებით განვიხილოთ ოპერაციული სისტემის ჩატვირთვის პროცესი.

წინასწარი ჩატვირთვის სტადიაზე აპარატურის შემოწმების გზით კომპიუტერი გადის თვითტესტირებას – **POST (Pover_on self test).** შემდეგ **BIOS** ახდენს ყველა მოწყობილობის კონფიგურირებას, ეძებს ჩატვირთვის დისკოს და მეხსიერებაში გადმოწერს *მთავარ ჩასატვირთ ჩანაწერს* **MBR (Master Boot Record),** რომლის კოდითაც მოინახება სისტემური ფაილების შემცველი აქტიური განყოფილება, მისგან კი გადმოიწერება ჩასატვირთი *(ნულოვანი)* სექტორი და შესრულდება მასში მოთავსებული ინსტრუქციები. წინასწარი ჩატვირთვის სტადია მთავრდება მეხსიერებაში **NTLDR** ჩამტვირთავის გადმოწერით.

ჩატვირთვის პროცესის დასაწყისში **NTLDR**-ი პროცესორს გადართავს 32-თანრიგიანი, ერთდონიანი მეხსიერების მოდელის რეჟიმში და ჩამოტვირთავს დრაივერებს სხვადასხვა ფაილურ სისტემებში **(FAT, FAT32, NTFS)** ჩაწერილ **Window 2000**-სთან ურთიერთობის დასამყარებლად.

შემდგომ, ჩვენ საშუალება გვეძლევა მენიუში ავირჩიოთ კომპიუტერზე დაყენებული რამდენიმე ოპერაციული სისტემიდან ერთ-ერთი *(თუ არჩევანს დავაყოვნებთ, მაშინ ხდება დუმილით გათვალისწინებული ოპერაციული სისტემის გაშვება).* ოპერაციული სისტემების მენიუ გადმოიწერება **BOOT.INI** ტექსტური ფაილიდან. თუ ეს ფაილი ხისტ დისკოზე არ ფიგურირებს, მაშინ **NTLDR** ჩამტვირთავი შეეცდება **Windows 2000**

ჩამოტვირთოს **WINNT** კატალოგიდან, რომელიც, როგორც წესი, განლაგებულია პირველი ხისტი დისკოს პირველ განყოფილებაში.

BOOT.INI ფაილს კი სისტემური განყოფილების ფესვურ საქაღალდეში ათავსებენ. იგი დამალული ფაილია, ამიტომ თუ მისი წაკითხვა გვსურს, საჭიროა ეს ფაილი გამოვაჩინოთ.

როცა კომპიუტერზე მხოლოდ **Windows 2000** ოპერაციული სისტემაა დაყენებული, **BOOT.INI** ფაილს შეიძლება ასეთი სახე ჰქონდეს:

[boot loader]

timeout=30

default = multi(0) disk(0) rdisk(0) partition(1) \WINNT

[operating systems]

multi (0)

disk(0) rdisk(0) partition(1) \WINNT="Microsoft Windows 2000 Server"

/fastdetect

საერთოდ კი, **[boot loader]** განყოფილებაში ჩაიწერება ინფორმაცია კომპიუტერზე დაყენებული ყველა ოპერაციული სისტემის შესახებ.

[operationg systems]-ში მიეთითება ამ ოპერაციული სისტემების ჩასატვირთ განყოფილებამდე გზები. ეს გზა ნაჩვენებია ე.წ. **ARC** სპეციალურ ფორმაში. მასში **multi** (ან **scsi**) ახდენს დისკოს კონტროლერის იდენტიფიცირებას, ხოლო **disk** პარამეტრი მიუთითებს დისკოს ნომერზე. **multi**-სთვის ეს მნიშვნელობა ნულია, ხოლო ეს **scsi**-სთვის იგი 0-7 დიაპაზონში იცვლება. **rdisk** პარამეტრი მხოლოდ **multi**-ის შემთხვევაში ფიგურირებს და ახდენს თვით დისკოს იდენტიფიცირებას.

partition პარამეტრის მნიშვნელობა განსაზღვრავს განყოფილების ნომერს. ამის შემდეგ შეიძლება მოდიოდეს რიგი მეტ-ნაკლებად მნიშვნელოვანი პარამეტრებისა, რომლებიც გავლენას ახდენენ ოპერაციული სისტემის ჩატვირთვის პროცესზე. ერთ-ერთი მათგანია **/fastdetect.**

შენიშვნა: *BOOT.INI-ში ცვლილებების შეტანა შეიძლება მაშინ განხორციელდეს, როცა ფაილის ატრიბუტებს მოეხსნებათ "სისტემური" და "მხოლოდ წაკითხვისათვის" მნიშვნელობები.*

როცა ეკრანზე ჩატვირთვის მენიუ ჩანს, **F8** ღილაკზე ხელის დაჭერით შეიძლება მივმართოთ ოპერაციული სისტემის

ჩატვირთვისათვის ისეთ დამატებით ვარიანტებს, როგორიცაა, მაგალითად: **Safe Mode** *(ოპერაციული სისტემის ჩატვირთვის უსაფრთხო რეჟიმი)* და სხვ. შევნიშნოთ, რომ **Safe Mode** რეჟიმში ჩაიტვირთება მხოლოდ ძირითადი დრაივერები.

ეკრანზე ჩატვირთვის მენიუს ასახვისას "ჰარზე" ხელის დაჭერით შეიძლება გამოვიყვანოთ სხვა მენიუც – **Hardware Profile/Configuration Recovery** *(აღჭურვილობის პროფილი/ კონფიგურაციის აღდგენა).* თუ კომპიუტერის ჩართვისას რაიმე პრობლემები შეგვექმნა *(მაგალითად, ახალი დრაივერის დაყენების გამო),* მათ თავიდან ავიცილებთ **<L>** კლავიშზე ხელის დაჭერით. ამ დროს ხდება მიმართვა ოპერაციული სისტემის ბოლო წარმატებული ჩატვირთვის ვარიანტისადმი **(Last Known Good Configuration)**, რომელიც უკვე დაფიქსირებულია რეესტრში.

თუ მენიუში არჩეულ იქნა **Windows 2000** ოპერაციული სისტემა, მეხსიერებაში ჩაიტვირთება **NTDETECT.COM** და **NTOSKRNL.EXE** ფაილები. ისინი განსაზღვრავენ დაყენებული აპარატურის პარამეტრებს. **NTLDR**-ს ეს ინფორმაცია რეესტრში შეაქვს.

ამის შემდეგ სისტემა გვთავაზობს **<L>** კლავიშით **Hardware Profile/ Configurations Recovery** მენიუს ეკრანზე გამოყვანას. აქაც გვეძლევა საშუალება ავირჩიოთ აღჭურვილობის ერთ-ერთი პროფილი ან **<L>** კლავიშზე ხელის დაჭერით მივმართოთ ოპერაციული სისტემის ბოლო წარმატებითი ჩატვირთვის დროს გამოყენებულ კონფიგურაციას **(Last Known Good Configuration).**

პროფილის არჩევას მოჰყვება **NTLDR**-ის მიერ **NTOSKRNL.EXE** ფაილის გადმოტვირთვა, რომელიც, თავის მხრივ, აკონტროლებს **Windows 2000**-ის ბირთვის გადმოწერას. თუ ოპერაციული სისტემის არჩევის შემდეგ პროფილს არ შევირჩევთ, მეხსიერებაში ჩაიტვირთება მისთვის დუმილით გათვალისწინებული ვარიანტი.

შემდეგ ხდება **HAL.DLL** ბიბლიოთეკის დინამიკურად მიერთება **HAL** აპარატურული აბსტრაქციების დონის გადმოსაწერად.

%SYSTEMROOT%\SYSTEM32\CONFIG\SYSTEM საქაღალდიდან გადმოიწერება ინფორმაცია აღჭურვილობის კონფიგურაციის შესახებ, რომელიც ჩაიწერება რეესტრის გასაღებში:

HKEY_LOCAL_MACHINE\SYSTEM

ამის შემდეგ მეხსიერებაში გადმოიწერება მოწყობილობების ქვედა დონის დრაივერები.

ჩატვირთვის გრაფიკული ნაწილი იწყება მაშინ, როცა ეკრანზე აისახება სურათი *(ქვედა ნაწილში ანიმაციური ზოლით).*

სისტემაში შესვლის სტადია შემდეგ ფაზებს მოიცავს: **WINLOGIN.EXE**-ის გაშვება, რომელიც, თავის მხრივ, უშვებს **Local Security Authority** სამსახურს (**LSASS.EXE** ფაილს). შედეგად, ეკრანზე აისახება შესვლის ლოკალური ფანჯარა **Logon.** მოდული **Service Controller** გაუშვებს სისტემასთან მუშაობისთვის განკუთვნილ სამსახურებს, მათ შორის **Workstation Service** და **Server Service**-ს.

აუთენტიფიკაციის წარმატებით გავლის შემდეგ ჩაიტვირთება მომხმარებლის პროფილი და მას უფლება ეძლევა, მიმართოს სერვერსა და ქსელურ სამსახურებს.

Windows 2000 Server-ის დაყენება

Windows 2000 Server მძლავრი ოპერაციული სისტემა გახლავთ და თავისი ფუნქციონირებისათვის იგი აპარატურას მაღალ მოთხოვნებს უყენებს. კერძოდ, სასურველია კომპიუტერი, რომელზეც ამ სისტემას ვაყენებთ, აგებული იყოს შემდეგ ბაზაზე:

პროცესორი – პენტიუმ-IV,

***RAM** – არანაკლებ 256 მეგაბაიტისა,*

ხისტი დისკის ტევადობა – არანაკლებ 4 გიგაბაიტისა,

*ვიდეოადაპტერი – **SVGA.***

თუ კომპიუტერზე არც ერთი ოპერაციული სისტემა არ ფუნქციონირებს, შეიძლება განვახორციელოთ მხოლოდ ე.წ. *სუფთა დაყენება*, სხვა შემთხვევაში საშუალება გვეძლევა, არჩევანი გავაკეთოთ სუფთა დაყენებასა და სისტემის განახლებას შორის *(უპირატესობას უკანასკნელ ვარიანტს ანიჭებენ).* დასაშვებია ძველი და ახალი ოპერაციული სისტემების თანაარსებობაც – ვგულისხმობთ ე.წ <u>ორმაგი ჩატვირთვის</u> რეჟიმით სარგებლობას, თუმცა ამ გზას შედარებით იშვიათად მიმართავენ. ბუნებრივია, ასეთ შემთხვევაშიც საქმე გვექნება სუფთა დაყენებასთან.

სუფთა დაყენებისას ოპერაციული სისტემა უნდა განთავსდეს ახალ საქაღალდეში ან ახალ განყოფილებაში. როცა სუფთა დაყენებას იმავე განყოფილებაში ვახდენთ, რომელშიც ძველი ოპერაციული სისტემა ფუნქციონირებს, სუფთავდება საქაღალდე **My Documents,** თავიდან გადაიწერება სისტემური ფაილები და იკარგება უსაფრთხოების ყველა პარამეტრი *(მათი განსაზღვრა ხელახლა მოგვიწევს).*

როცა კომპიუტერზე არანაირი ოპერაციული სისტემა არა გვაქვს *(როგორც ზემოთ აღვნიშნეთ, ასეთ შემთხვევაში მხოლოდ სუფთა დაყენება შეიძლება განვახორციელოთ),* პირველ ყოვლისა, მასზე ვაყენებთ ოპერაციული სისტემის მცირე ვერსიას. ამ მიზნით იყენებენ 4 დისკეტისაგან შემდგარ <u>ინსტალირების პაკეტს</u>. თუ ასეთი პაკეტი არ გაგვაჩნია, მას ვქმნით სხვა კომპიუტერზე, საბრძანებო სტრიქონიდან:

x:\makeboot.bat a:

აქ **x:** კომპაქტ-დისკოს წამკითხველი მოწყობილობის სახელია.

ოპერაციული სისტემის დაყენების პროცესი ორ ძირითად ეტაპად მიმდინარეობს:

- ტექსტური
- გრაფიკული

ძირითად ეტაპებს კი წინ უძღვის შემდეგი მოსამზადებელი სამუშაოები:

უპირველეს ყოვლისა, უნდა გადავწყვიტოთ, ოპერაციული სისტემის დაყენების რომელ ვარიანტს მივმართავთ: სუფთას თუ არსებულის განახლების. შემდეგ, თუ კომპიუტერზე ოპერაციული სისტემა უკვე ფუნქციონირებს *(მეტწილად ასეცაა),* შეგვიძლია არჩევანის გაკეთება. თუმცა, როგორც უკვე აღვნიშნეთ, რეკომენდაციას ამ შემთხვევაში განახლების ვარიანტს უწევენ.

ჯერ გავეცნოთ ოპერაციული სისტემის დაყენების პროცესს ისეთ კომპიუტერზე, რომელზეც სისტემის წინა ვერსია არა გვაქვს. ცხადია, ამ შემთხვევაში ოპერაციული სისტემის მხოლოდ სუფთა დაყენების ვარიანტს შეგვიძლია მივმართოთ.

დისკოწამყვანში ვათავსებთ ინსტალირების პაკეტიდან პირველ დისკეტას და კომპიუტერს გადავტვირთავთ. დაყენების პროგრამა თანამიმდევრულად მოგვთხოვს დანარჩენი სამი დისკეტის ჩადგმასაც, რასაც კვლავ მოჰყვება კომპიუტერის გადატვირთვა და ეკრანზე აისახება მოთხოვნა – წამკითხავ

მოწყობილობაში მოვათავსოთ ინსტალირებისათვის განკუთვნილი კომპაქტ-დისკო.

თუ არ გამოიყენება დისკოების ავტომატურად გამომცნობი **AutoPlay** ფუნქცია, მივმართავთ საბრძანებო სტრიქონს:

x:\setup

ეკრანზე აისახება დიალოგის ფანჯარა და იწყება ჩვენი დიალოგი ინსტალირების ოსტატ-პროგრამასთან.

იმ შემთხვევაში, როცა კომპიუტერზე ფუნქციონირებს წინა ვერსიის ოპერაციული სისტემა, ინსტალირების პროცესს ვიწყებთ წამკითხავ მოწყობილობაში კომპაქტ-დისკოს მოთავსებით და პირველივე ფანჯარაში ვიღებთ გადაწყვეტილებას - ოპერაციული სისტემის დაყენების რომელი ვარიანტია ჩვენთვის უფრო მისაღები. **Next.**

მომდევნო ფანჯარაში **Agree** *(თანხმობა)* ღილაკზე ხელის დაჭერით ვეთანხმებით **MicroSoft**-ის მიერ შემოთავაზებულ სალიცენზიო პირობებს, **Next** და შემდგომ ფანჯარაში შეგვაქვს ოპერაციული სისტემის ჩვენი ასლის სერიული ნომერი.

აღვნიშნოთ, რომ თუ ჩვენს კომპიუტერზე დაყენებულია **Windows NT** ან **Windows 2000,** შეგვიძლია მოვახდინოთ მათი განახლება **Windows 2000 Server**-მდე, **Windows 98**-ის განახლება კი არ ხდება – გადამრთველი ასეთ შემთხვევაში არააქტიური იქნება.

Next-ით გადავდივართ მომდევნო დიალოგურ ფანჯარაში. აქ შეგვიძლია განვსაზღვროთ სამუშაო ენები და მათი პარამეტრები.

Advanced Options *(დამატებითი პარამეტრები)* ღილაკზე დაწკაპუნებით იღება ახალი დიალოგური ფანჯარა. თუ კომპიუტერზე არსებული ოპერაციული სისტემის შენარჩუნება გვსურს, მოვნიშნავთ ალამს

I Want to Choose the Installation Partition During Setup *(მსურს დაყენების განყოფილების არჩევა).*

შესაბამისად, საშუალება გვეძლევა სხვა განყოფილებაში მოვახდინოთ ოპერაციული სისტემის სუფთა დაყენება.

თუ ხისტ დისკოზე თავისუფალი ადგილი გაგვაჩნია, შემდეგი ალმის მონიშვნით

Copy all Setup files from Setup CD to the Hard Drive

შევძლებთ კომპაქტ-დისკოდან საინსტალაციო ფაილების გადმოწერას, რაც მომავალში თავიდან აგვაცილებს კომპაქტ-დისკოს გამოყენების საჭიროებას. **OK. NEXT.**

დაიწყება ფაილების კოპირება.

საწყისი სტადიის ბოლოს დაყენების პროგრამა მოითხოვს კომპიუტერის გადატვირთვას. ამ მომენტისათვის ხისტ დისკოზე უკვე დაყენებული იქნება ოპერაციული სისტემის მინიმალური ჩამტვირთავი ვერსია.

ვაწკაპუნებთ **Finish**-ზე.

<u>დაყენების ტექსტური ეტაპი</u>

ეკრანზე აისახება ჩამტვირთავი მენიუ, რომელშიც ერთ-ერთ ვარიანტად ფიგურირებს

Microsoft Windows 2000 Server Setup

ვირჩევთ სწორედ ამ ვარიანტს.

ტექსტური ეტაპი **DOS** რეჟიმში მიმდინარეობს. თუ საქმე გვაქვს სუფთა დაყენებასთან, ხისტ დისკოზე უნდა შევქმნათ და დავაფორმატოთ ახალი განყოფილება *(იხ. ქვემოთ).* ოპერაციული სისტემის განახლებისას კი ეს ოპერაციები არ ტარდება. კოპირდება დრაივერების, ქსელური სამსახურების და ოპერაციული სისტემის კომპონენტების ფაილები. სისტემის განახლებისას მისი კონფიგურირება ხდება ყველა მოცემული პარამეტრის შენარჩუნებით.

მომდევნო გადატვირთვის შემდეგ გამოდის დიალოგური ფანჯარა

Configure Your Server *(სერვერის პარამეტრების გაწყობა).*

სანამ ოპერაციული სისტემის დაყენების გრაფიკულ ეტაპზე გადავიდოდეთ, განვიხილოთ დისკოზე ახალი განყოფილების შექმნის საკითხი *(დისკოს დაფორმატება).*

Welcome to Setup ფანჯარაში ვაწკაპუნებთ **Enter** ღილაკზე. ეკრანზე გამოდის დისკოების და განყოფილებების სია.

Windows 2000 Server-ის სუფთა დაყენების სამი ვარიანტიდან ვირჩევთ ერთ-ერთს:

- ხისტ დისკოზე ახალი განყოფილების შექმნა *(თუ რომელიმე განყოფილებაში ადგილი საკმარისია),*
- ხისტ დისკოზე ახალი განყოფილების შექმნა *(თუ არსებობს საკმარისი ადგილი, რომელიც არ ეკუთვნის არც ერთ არსებულ განყოფილებას),*
- არსებული განყოფილების გამოყენება *(თუ ადგილი საკმარისია).*

სივრცის გამოსათავისუფლებლად დასაშვებია არასაჭირო განყოფილების ამოგდებაც – მას მოვნიშნავთ და ვანადგურებთ ჯერ <D> და შემდეგ *(დასტურისათვის)* <L> კლავიშების მეშვეობით.

ოპერაციული სისტემისათვის სივრცის შერჩევის შემდეგ ახალ განყოფილებას ვქმნით <C> კლავიშზე ხელის დაჭერით. ახლახან შექმნილ განყოფილებას მოვნიშნავთ და <Enter>-ზე ხელის დაჭერით ვიწყებთ მასში ოპერაციული სისტემის ინსტალირებას.

Windows 2000 Server-ის დასაყენებლად უნდა ავირჩიოთ ერთ-ერთი ფაილური სისტემა:

FAT, FAT32 ან **NTFS.**

უპირატესობას აძლევენ უკანასკნელ ვარიანტს *(დაწვრილებით შემდგომ).* ამჯერად აღვნიშნოთ, რომ თუ კომპიუტერს დომენის კონტროლერის როლს ვუმზადებთ, მაშინ განყოფილებისთვის, რომელშიც ოპერაციული სისტემა დაყენდება, აუცილებელია **NTFS** ფაილური სისტემის არჩევა.

<u>დაყენების გრაფიკული ეტაპი</u>

კომპიუტერის მორიგი გადატვირთვის შემდეგ იწყება ოპერაციული სისტემის ინსტალირების გრაფიკული ეტაპი. მისი პირველი ნაწილი - ზოგადი პარამეტრების შეტანა – შეიძლება წარმოვიდგინოთ, როგორც **Control Panel**-ის აპლეტებისადმი თანმიმდევრული მიმართვები. ერთ-ერთი მათგანის - **Component**-ის მეშვეობით განვსაზღვრავთ იმ კომპონენტების სიას, რომლებიც უნდა დაყენდეს კომპიუტერზე *(ყველა კომპონენტისთვის საჭირო მეხსიერება 80,4 მეგაბაიტია).*

შევნიშნოთ, რომ კომპონენტების დაყენება-ამოგდება შესაძლებელია განხორციელდეს ოპერაციული სისტემის დაყენების შემდეგაც **Add/Remove Programs** აპლეტის მეშვეობით.

ზოგადი პარამეტრების განსაზღვრის შემდეგ დაყენების პროგრამა ცდილობს გამოიცნოს, რომელი ქსელური ადაპტერია ჩაყენებული კომპიუტერში *(მათი რაოდენობა შესაძლოა, ერთზე მეტიც იყოს).* უმეტეს შემთხვევაში ეს ხერხდება, მაგრამ თუ ასე არ მოხდა, დაყენების პროგრამა მოგვთხოვს საჭირო დრაივერის ჩვენებას.

ქსელური კომპონენტების დაყენების **Typical Settings** და **Custom Settings** ვარიანტებიდან (***Networking Settings*** *დიალოგურ*

ფანჯარაში) ჯერჯერობით უპირატესობას ვაძლევთ პირველ მათგანს.

პროგრამა გვეკითხება, კომპიუტერი სამუშაო ჯგუფს მიუერთოს თუ დომენს. რადგანაც ჩვენ მხოლოდ ვიწყებთ ქსელის შექმნას, ამ ეტაპზე მივუერთდებით წარმოსახვითს სამუშაო ჯგუფს – იყოს იგი **CORPGROUP** სახელის მქონე.

ქსელური კომპონენტების დაყენების პროცესის დასრულების შემდეგ კომპიუტერი უნდა გადაიტვირთოს. ეკრანზე კვლავ აისახება დიალოგური ფანჯარა **Configure Your Server** *(სერვერის პარამეტრების გაწყობა).* მასში ავირჩევთ ერთ-ერთ ვარიანტს ქვემოთ მოყვანილებიდან:

- **This is the onlay server in my network** *(ეს სერვერი ერთადერთია ქსელში)*
- **One or more server are already running on my network** *(ქსელში უკვე მუშაობს ერთი ან რამდენიმე სერვერი)*
- **I will configure this server later** *(სერვერის გაწყობა სხვა დროისათვის გადავდოთ)*

პირველ პუნქტს მივმართავთ მაშინ, როცა ქსელში ვქმნით დომენის პირველ კონტროლერს და მაშინაც, როცა მასში კიდევ ერთ კონტროლერს ვამატებთ.

ცალკე მდგომი სერვერის შესაქმნელად მივმართავთ მეორე ვარიანტს. ასეთი სერვერები, როგორც წესი, საერთო გამოყენებებით სარგებლობის საშუალებას იძლევა და მათ დრო არ ეხარჯება იმ ქსელურ ოპერაციებზე, რომლებიც სრულდება კონტროლერების მიერ.

კვლავ გადავტვირთავთ კომპიუტერს.

ოპერაციული სისტემის ინსტალირების ბოლოს უნდა შევირჩიოთ ლიცენზირების მოდელი.

არსებობს <u>ლიცენზია სამუშაო ადგილზე</u> და <u>ლიცენზია სერვერზე</u>. ჩვენი ქსელის მუშაობის სპეციფიკიდან გამომდინარე, შეიძლება მომგებიანი იყოს ან ერთი, ან მეორე ვარიანტი. განვსაზღვრავთ შესაძენი ლიცენზიების რაოდენობას ორივე მოდელისათვის და ვაკეთებთ არჩევანს.

დავუშვათ, ქსელში ჩართულია 100 კომპიუტერი-კლიენტი, რომელთაგან სერვერთან ერთდროულად მიერთებული შეიძლება იყოს მხოლოდ 25. კლიენტ-სერვერული შეერთებების რიცხვი *(ქსელში მხოლოდ ერთი სერვერის არსებობისას)* იქნება:

სერვერზე ორიენტირებული მოდელისათვის – 1 x 25 = 25
კლიენტზე ორიენტირებული მოდელისათვის – 1 x 100 = 100

აქ სავსებით ცხადია, რომელ მოდელს უნდა მიენიჭოს უპირატესობა, მაგრამ სიტუაცია შეიძლება შეიცვალოს იმ შემთხვევისათვის, როცა ქსელში სერვერების რიცხვი მეტია.

დავუშვათ, ამჯერად ქსელში ჩართულია 5 სერვერი. ამასთან, ერთ-ერთ მათგანთან ერთდროულად კავშირს ამყარებს 50 კლიენტი, ხოლო 20 კლიენტს კი ასევე ერთდროულად დანარჩენ 4 სერვერთან აქვს კავშირი. ამ შემთხვევაში გვექნება:

სერვერზე ორიენტირებული მოდელისათვის – 1 x 50 + + 4 x 20=130

კლიენტზე ორიენტირებული მოდელისათვის კი, რომელიც მხოლოდ კლიენტების მაქსიმალურ რიცხვს ითვალისწინებს, არაფერი შეიცვლება – 1x100= 100

ამჯერად, არჩევანი მეორე მოდელის სასარგებლოდ უნდა გავაკეთოთ. საერთოდ კი, ქსელის შექმნის დასაწყისში უმჯობესია პირველი მოდელის არჩევა. საქმე ისაა, რომ მაიკროსოფტი ნებას გვრთავს, ერთხელ ეს მოდელი უფასოდ შევცვალოთ მეორეთი *(პირიქითი ქმედება კი არ დაიშვება).*

როდესაც არჩეული გვაქვს სერვერზე ორიენტირებული მოდელი, **Systems Mamagement Server** პროგრამა ამოწმებს, ერთდროულად რამდენი კლიენტი უერთდება მოცემულ სერვერს. თუ შეერთებების რიცხვმა ლიცენზიების რიცხვს გადააჭარბა, ხდომილობების რეგისტრაციის ჟურნალში კეთდება შესაბამისი ჩანაწერი, რომლის ნახვაც შესაძლებელია **Event Viewer** *(ხდომილებების ჩათვალიერება)* პროგრამის მეშვეობით.

დომენები

Active Directory –კატალოგების სამსახური

დომენებს ჯერ კიდევ **Windows NT** ქსელური ოპერაციული სისტემის შესწავლის დროს გავეცანით.

გავიხსენოთ, დომენი არის კომპიუტერების ჯგუფი, რომელიც იყენებს მონაცემების საერთო ბაზასა და უსაფრთხოების საერთო პოლიტიკას.

დომენებს შორის შესაძლებელია ნდობითი დამოკიდებულებების დამყარება.

Windows NT-ში ნდობით დამოკიდებულებებს ტრანზიტულობა არ ახასიათებდა, რაც ართულებდა სისტემის მართვას.

ამასთან, შეზღუდული იყო დომენების ურთიერთდაკავშირების ვარიანტების რიცხვი.

Windows 2000-ში არსებითად შეიცვალა დომენების სტრუქტურა და მართვის იდეოლოგია. თუკი **NT**-ში გვქონდა 3 ტიპის სერვერი, მათ შორის 2 ტიპის კონტროლერი *(მთავარი და სარეზერვო),* **Windows 2000**-ის დომენში შეიძლება იმყოფებოდეს რამდენიმე ტოლფასი კონტროლერი, ე.ი. სარეზერვო კონტროლერის ტიპი აღარ გამოიყენება.

როგორც კი **Active Directory** კატალოგების სამსახურში *(იხ. ქვემოთ)* რაიმე ცვლილებები მოხდება, ისინი გავრცელდება ყველა სხვა კონტროლერზეც. ამ პროცესს ეწოდება <u>*რეპლიკაცია*</u>. რეპლიკაცია დიდად ამაღლებს მტყუნებებისადმი სისტემის მდგრადობის უნარს. ასეთ მიდგომას სხვა უპირატესობაც გააჩნია: კლიენტ-კომპიუტერს **Active Directory**-ში არსებული ინფორმაცია შეუძლია უფრო სწრაფად *(უახლოესი კონტროლერიდან)* მიიღოს.

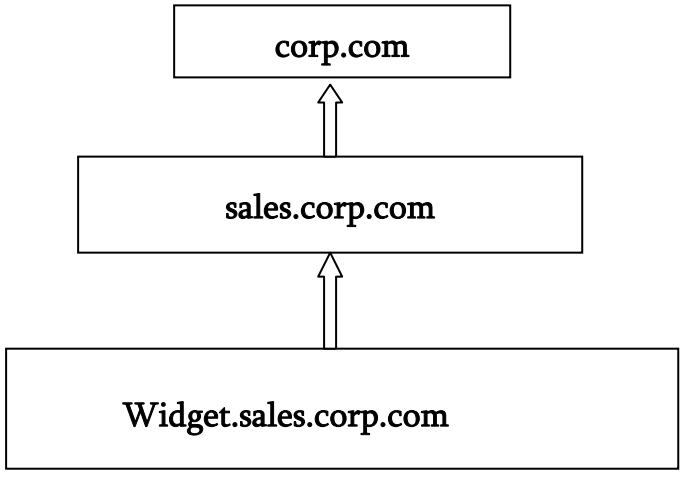
თუ **Active Directory**-ში განთავსებული ინფორმაცია ინტენსიურად იცვლება, ქსელში ტრაფიკის შესამცირებლად მიმართავენ სპეციალურ ღონისძიებებს - ქმნიან <u>კვანძებს</u> და იყენებენ <u>გლობალური კატალოგების სერვერებს</u> *(იხ. ქვემოთ).*

<u>კატალოგების სამსახურის - Active Directory-ის არსი</u>

Windows NT-ში სისტემის მომხმარებლების შესახებ ინფორმაციის შენახვა-მართვას განაგებდა **SAM (Security Account Manager)** კომპონენტი. იგი თავის ფუქციებს თავს კარგადართმევდა, მაგრამ ჰქონდა ნაკლიც, კერძოდ, ადმინისტრატორს არ შეეძლო ბაზაში თავის სურვილისამებრ რაიმე ინფორმაციის შეტანა *(მაგალითად, მომხმარებლის ფოტოგრაფიის განთავსება).* **Windows 2000**-ში კი გაჩნდა როგორც ეს შესაძლებლობა, ასევე – სხვებიც, მაგალითად, საჭიროების შემთხვევაში მომხმარებლისთვის ადმინისტრატორის ზოგი ფუნქციის გადაცემის, ინტერნეტის პროტოკოლების საყოველთაო გამოყენების და ა.შ.

კატალოგების სამსახური ქსელის ყველა ობიექტის შესახებ ინახავს ინფორმაციას. მაგალითად, პრაქტიკულად მყისიერად შეიძლება მივიღოთ ინფორმაცია ყველა იმ პროგრამისტის *(თუ იურისტის)* შესახებ, რომლებიც მუშაობენ კორპორაციის ადმინისტრაციული შენობის მესამე სართულზე და სამუშაო მაგიდაზე უდგათ ფერადი ლაზერული პრინტერი.

შემდეგ, **Windows 2000**-ის დომენური სტრუქტურა ძალზე დაუახლოვდა ინტერნეტში მიღებულს, რაც აისახა კიდეც **Active Directory**-ის სტრუქტურაში. მოვიყვანოთ კორპორაციისა და მისი შემადგენელი ნაწილებისათვის დომენური სტრუქტურის მაგალითი.



ისარი მიგვითითებს ნდობითი დამოკიდებულების მიმართულებაზე.

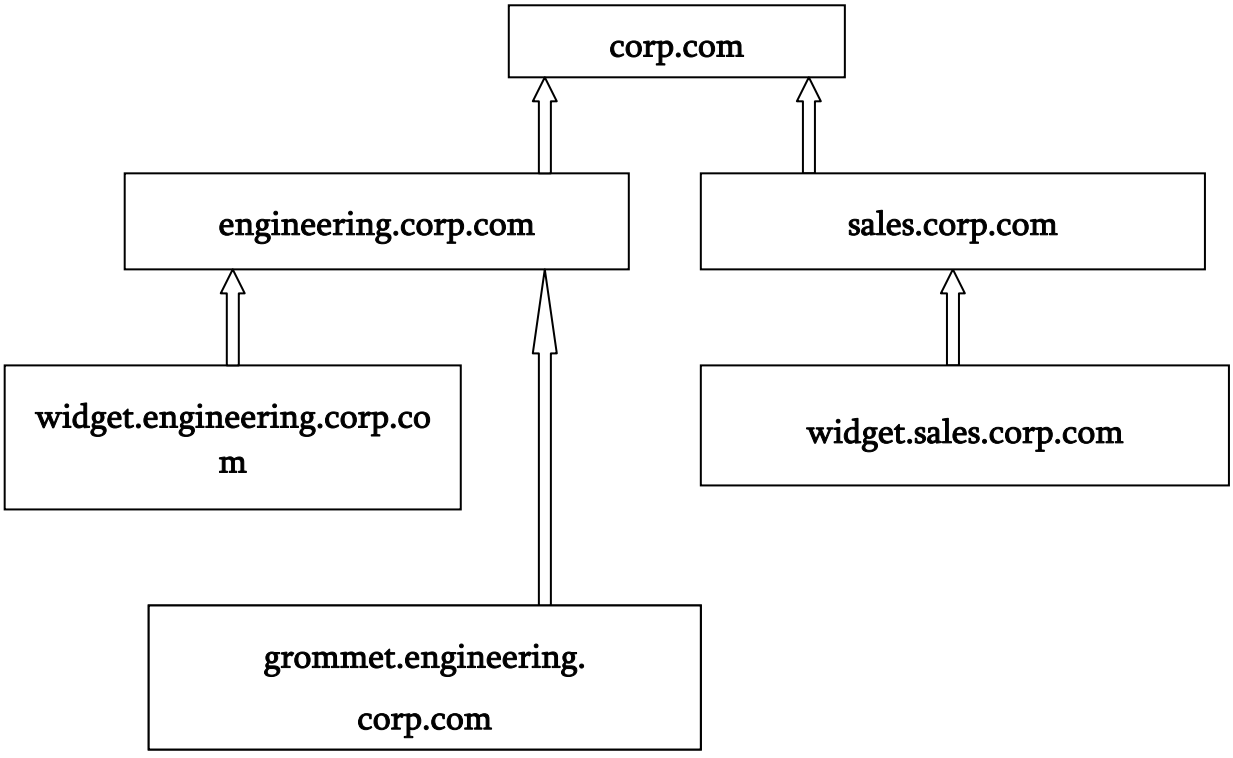
ცხადია, აქაც გვჭირდება **DNS (Domain Name Service)** სამსახურით სარგებლობა.

გავიხსენოთ, რომ **DNS** სამსახური ინტერნეტში **Web**-სერვერის ადვილად დასამახსოვრებელ სიტყვიერ სახელწოდებას გარდაქმნის ოთხი ბაიტისაგან შემდგარ ციფრულ მისამართად, მაგალითად: www.boutell.com-ს შეეთანადება 207.55.56.4 ინფორმაცია.

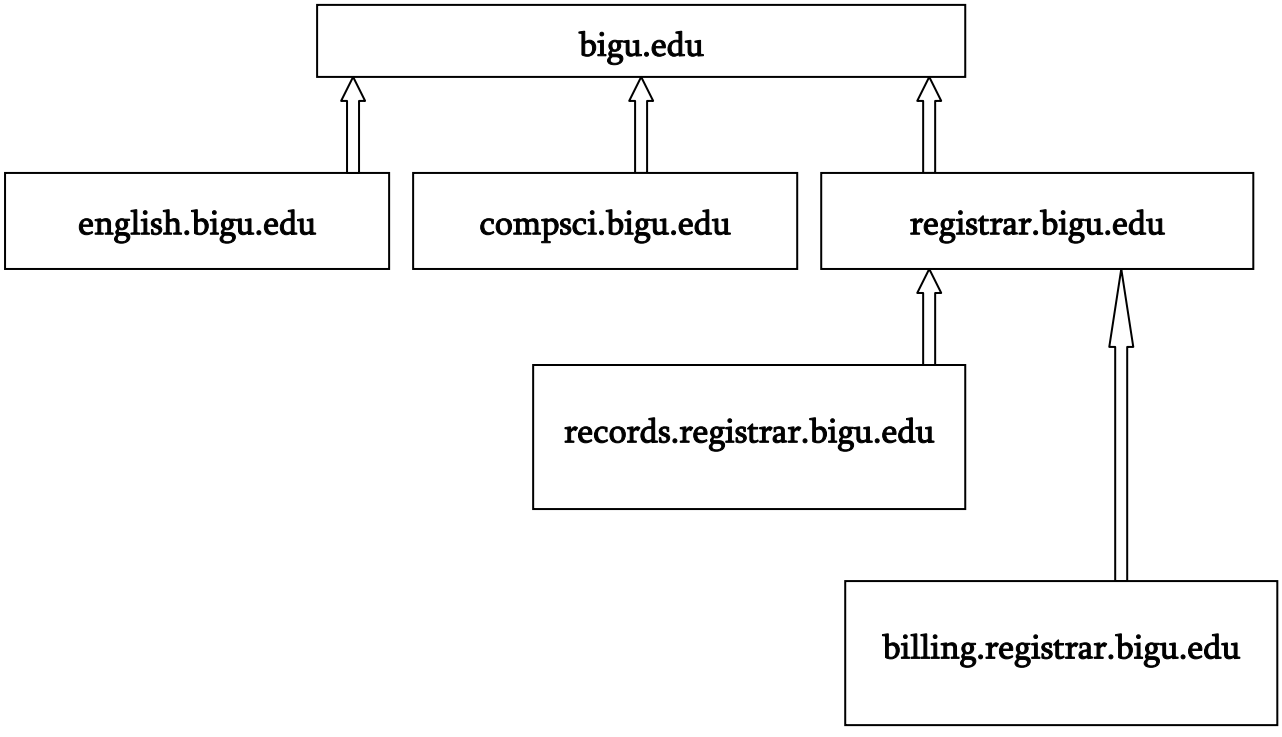
ამ მოთხოვნის გამო, **Windows 2000**-ის სერვერები თავის თავზე იღებენ **DNS**-სერვერების ფუნქციასაც.

დომენების ხეები და ტყეები

ვიცით, რომ **Windows NT**-ში დომენები ერთმანეთთან კავშირდება ნდობითი დამოკიდებულებებით. **Windows 2000**-ში გაჩნდა დომენების წარმოდგენის საშუალებები ხეებისა და ტყეების სტრუქტურებად. მათ შეეთანადა **Active Directory**-ის შესაბამისი სტრუქტურებიც. მოვიყვანოთ ხის სტრუქტურის მაგალითები:

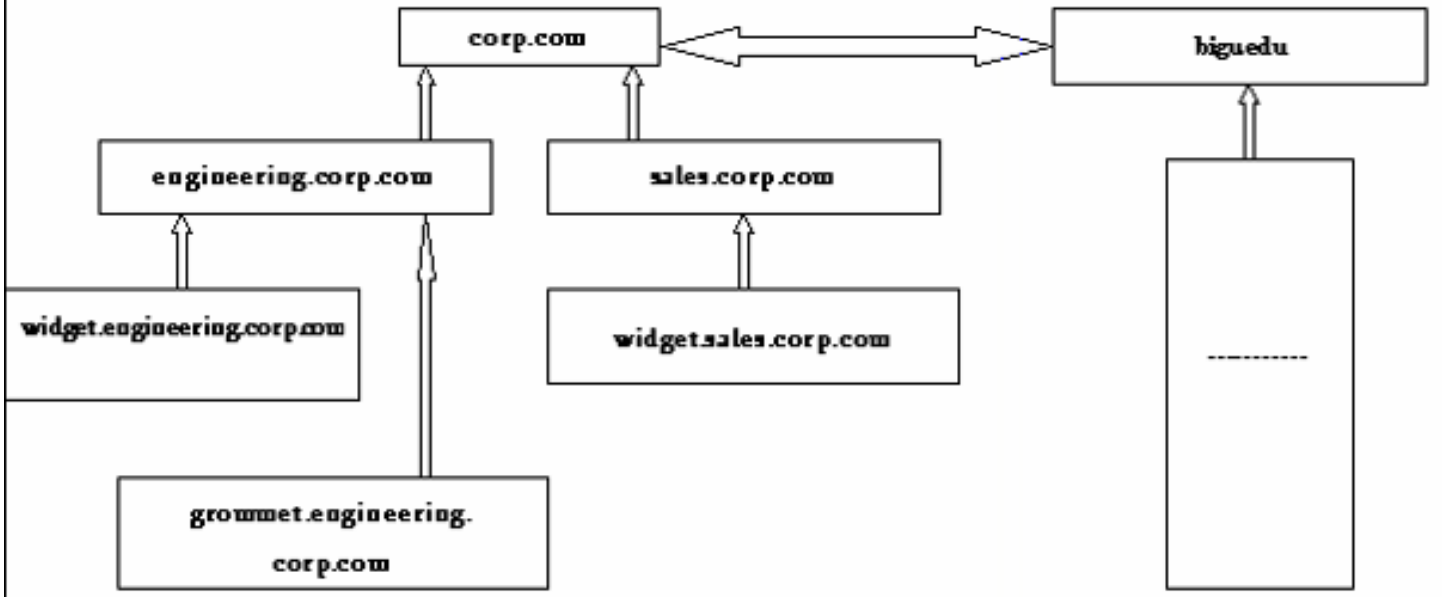


ნახ. 1. ხის სტრუქტურის მაგალითი



ნახ. 2. ხის სტრუქტურის მაგალითი

დომენური ტყის სტრუქტურის მაგალითია:



ნახ. 3

ვხედავთ, რომ ორი ხის ფესვური დომენი ერთმანეთთან კავშირდება ორმხრივ მიმართულ ნდობით დამოკიდებულებით.

კომპიუტერის გარდაქმნა დომენის კონტროლერად

მას შემდეგ, რაც კომპიუტერის ერთ-ერთ, **NTFS5** ფაილური სისტემით დაფორმატებულ განყოფილებაში დავაყენებთ **Windows 2000 Server**-ს, შესაძლებელია კომპიუტერი დომენის კონტროლერად ვაქციოთ.

შევნიშნოთ, რომ ამ ეტაპის წინ აუცილებელია სხვა ფაილური სისტემით დაფორმატებული განყოფილება გადაყვანილ იქნეს **NTFS5** ფორმატში, რისთვისაც გათვალისწინებულია **Convert.exe** უტილიტა.

გარდაქმნა ასე ხორციელდება:

1. **Start⇨Run**-ით გამოყვანილ სტრიქონში შეგვყავს **dcpromo** ბრძანება, **<Enter>**.
2. შემდგომ დიალოგურ ფანჯარაში ვაწკაპუნებთ **Next** ღილაკზე.
3. მომდევნო სამ ფანჯარაში კვლავ **Next**-ს ვირჩევთ.
4. შემდგომ ლოკალურ ფანჯარაში დომენის სახელად ავირჩიოთ **corp.com.** კვლავ **Next.**
5. **NetBIOS** დომენის სახელად ვტოვებთ დუმილით შემოთავაზებულ **CORP** სახელწოდებას.
6. **Database and Log Locations** *(მონაცემთა ბაზის და ჟურნალის მდებარეობა)* ფანჯარაშიც დუმილით გათვალისწინებულ პარამეტრებს ვინარჩუნებთ. მაშასადამე, ვირჩევთ იმავე საქაღალდეს, რომელშიც მანამდე ოპერაციული სისტემა იქნა დაყენებული.
7. საერთო სისტემური ტომის (**Shared System Volume**) არჩევის დროსაც შემოთავაზებულ ვარიანტს ვეთანხმებით.
8. თუ ოპერაციული სისტემის ინსტალირების დროს **DNS** სერვერი არ დაგვიყენებია, ეკრანზე ვღებულობთ შესაბამის შეტყობინებას. **OK.**
9. **Configure DNS** დიალოგურ ფანჯარაში ვაწკაპუნებთ **Next**-ზე *(პირველი გადამრთველის დაყენების ვარიანტისთვის).*
10. **Permissions** ფანჯარაში ვირჩევთ ვარიანტს **Windows 2000 Compatible Only** (*მხოლოდ **Windows 2000**-თან შეთავსებადი*).
11. ეკრანზე **Summary** ფანჯარაში აისახება ჩვენ მიერ შერჩეული პარამეტრების სია. **Next.**

12. იწყება ინსტალირების პროცესი *(იგი გარკვეულ დროს მოითხოვს).*
13. კატალოგების სამსახურის დაყენების შესახებ ინფორმაციას გვაწვდის იგივე **Summary** ფანჯარა. **Finish**.
14. დაბოლოს, ვაწკაპუნებთ **Restart Now** ღილაკზე.

შენიშვნა: შემდგომ, საჭიროების შემთხვევაში იმავე დიალოგურ ფანჯრებში შევძლებთ არსებული ტყისა და ხისათვის ახალი დომენების და ხეების დამატებას.

ორგანიზაციული ერთეულები

ყოველი დომენის შექმნა მოითხოვს მასში მინიმუმ ერთი კონტროლერის არსებობას. კონტროლერის ფუნქციების შესრულება კი ძვირადღირებულ კომპიუტერს შეუძლია. ამ და სხვა მოსაზრებების გამოც ზოგჯერ ქსელურ რესურსებს დომენის მაგიერ განათავსებენ ე.წ. **ორგანიზაციულ ერთეულებში**. ორგანიზაციული ერთეულები, თავისი არსით, ობიექტების ლოგიკურ კონტეინერებს წარმოადგენს. ამასთან, ობიექტის როლში, ფაქტობრივად, ნებისმიერი რესურსი შეიძლება გამოვიდეს: მომხმარებლის სააღრიცხვო ბარათი, პრინტერი, კომპიუტერი, თვით სხვა ორგანიზაციული ერთეულიც კი.

საინტერესოა, რომ ორგანიზაციული ერთეულის ადმინისტრირება შეიძლება მიენდოს რიგით მომხმარებელსაც, მაგალითად, ნება დაერთოს მას, შექმნას ორგანიზაციულ ერთეულში ახალი სააღრიცხვო ჩანაწერები.

ალტერნატივებიდან - შეიქმნას დომენი თუ ორგანიზაციული ერთეული - პირველს უპირატესობას აძლევენ შემდეგ შემთხვევებში:

- სხვადასხვა ჯგუფებისათვის უსაფრთხოების პოლიტიკის პარამეტრებისადმი სხვადასხვა მოთხოვნების დროს;
- ჯგუფები იყენებენ სხვადასხვა ენებს;
- საჭიროა ჯგუფის მმართველი ფლობდეს ადმინისტრირების ყველა ინსტრუმენტს.
- ობიექტების რიცხვი ძალიან ბევრია (*ქსელი მოიცავს ათეულ ათასობით ობიექტს).*

სხვა შემთხვევებში არჩევენ ორგანიზაციულ ერთეულის შექმნას *(ეს პროცესი შემდგომში უფრო დაწვრილებით იქნება განხილული.)*

კვანძების დაპროექტება და მხარდაჭერა

Windows 2000 Server-ისთვის **დომენისა** და **კვანძის** ცნებები იდენტური არ გახლავთ.

გავიხსენოთ, რომ დომენი არის კომპიუტერების და საადრიცხვო ჩანაწერების ლოგიკური ჯგუფი.

კვანძი კი არის ***ფიზიკური ადგილი*** *(კომპიუტერი, კომპიუტერების ჯგუფი, ჯგუფები),* რომელთანაც დაკავშირება ხორციელდება **TCP/IP** ოქმების მეშვეობით.

ერთი დომენი შეიძლება შეიცავდეს რამდენიმე კვანძს, ასევე, ერთი კვანძიც შეიძლება შეიცავდეს რამდენიმე დომენს.

უფრო კარგად რომ გავერკვეთ კვანძის არსში, აღვნიშნოთ ზოგიერთი გარემოება:

Active Directory-ის მონაცემების ბაზის რეპლიკაცია ავტომატურად ხორციელდება მოცემული დომენის ყველა კონტროლერზე. იგივე ხდება მოცემულ კვანძში შემავალ ყველა კონტროლერზეც. მაგრამ თუკი დომენში შემავალი კონტროლერები სხვადასხვა ქალაქშია, მათ შორის ძვირადღირებული გლობალური ქსელით ინფორმაციის ხშირი გაცვლა *(ბაზაში ყოველი ცვლილებებისას)* დიდ ხარჯებს გამოიწვევს.

გამოსავალი ასეთია – დომენში ითვალისწინებენ რამდენიმე კვანძს, რომელთა შორისაც ინფორმაციის გადაგზავნისთვის ირჩევენ მისაღებ დროით ინტერვალებს.

კვანძების შესაქმნელად და მათთან მუშაობისთვის განკუთვნილია პროგრამა

Active Directory Sites and Service Manager

აღვნიშნოთ, რომ ჩვენ მიერ კონტროლერის შექმნისას მასთან დუმილით დაკავშირდა **Default-First-Site** კვანძი. გადავარქვათ კვანძს სახელი და ვუწოდოთ **Delaware**, ხოლო მომსახურე სერვერად დავუნიშნოთ **W2kserver1.**

ამის შემდეგ, შევქმნათ ახალი კვანძი **Arizona** და სერვერად დავუნიშნოთ **AZDC1.**

შემდგომი ნაბიჯებია თითოეული კვანძისთვის ქვექსელის განსაზღვრა, კვანძებს შორის კავშირის დამყარება და რეპლიკაციის ინტერვალის განსაზღვრა *(დუმილით გაით-ვალისწინება 180 წუთი).*

DNS სახელების მინიჭება

პირველ ყოვლისა, ორგანიზაციისთვის ირჩევენ ზედა დონის დომენის ისეთ სახელს, რომელიც ყველაზე მეტად შეესაბამება მას. ინტერნეტის შესწავლისას ჩვენ უკვე გავეცანით ამ სახელებს. ესენია:

- სახელმწიფოს შემოკლებული დასახელება, მაგალითად, **uk, ge, ru;**
- ორგანიზაციის საქმიანობის აღმნიშვნელი შემოკლებული ტერმინები, მაგალითად, **com, edu, org, mil, gov;**
- ზოგი სხვა დასახელებაც.

მომდევნო დონის დომენური სახელის ასარჩევად უნდა შევამოწმოთ, ხომ არ გამოიყენება ჩვენ მიერ შერჩეული სახელწოდება უკვე ინტერნეტში? ამ ინფორმაციის მოსაპოვებლად მიმართავენ **www.internic.net** კვანძს *(საიტს).*

შევნიშნოთ, რომ საკმარისია მხოლოდ "ფესვური" სახელწოდების დარეგისტრირება *(მაგალითად, corp.com-ის),* ანუ საჭირო არ გახლავთ ხეზე არსებული მომდევნო დონის დომენებისათვის სახელის დარეგისტრირება.

(დაწვრილებით ეს საკითხი განიხილება ლაბორატორიულ მეცადინეობაზე).

Active Directory კატალოგების სამსახურის მონაცემთა ბაზის რეპლიკაცია და გლობალური კატალოგი

დომენების ტყეში გლობალური კატალოგის განთავსება, როგორც წესი, ხდება პირველად შექმნილ დომენის კონტროლერზე. უპირველეს ყოვლისა, სწორედ ასეთი კონტროლერი ცდილობს მომხმარებელთა მოთხოვნების დაკმაყოფილებას და უმეტეს შემთხვევაში პოულობს კიდეც საჭირო ინფორმაციას ე.წ. **ნაწილობრივ რეპლიკაში** (**Active Directory**-ის ბაზის ნაწილის ასლში). მაგრამ ზოგჯერ შედარებით მეორეხარისხოვანი ინფორმაციის მოსაძებნად გლობალური კატალოგი მოთხოვნას გადაუგზავნის შესაბამისი დომენის კონტროლერს.

ამრიგად, დომენის რიგითი კონტროლერი თავის **Active Directory**-ში ინფორმაციას ინახავს საკუთარ დომენში შემავალი ყოველი ობიექტის თითოეული ატრიბუტის შესახებ, ხოლო დომენის ის კონტროლერი, რომელზეც გლობალური კატალოგია

განლაგებული *(ანუ გლობალური კატალოგის სერვერი),* ტყეში შემავალი დომენების კატალოგებიდან გადმოიწერს ინფორმაციას ყოველი ობიექტის მხოლოდ ზოგიერთი ატრიბუტის შესახებ.

Microsoft კორპორაციის მიერ შემოთავაზებული სქემა ასეთია:

გლობალური კატალოგის შემცველ სერვერზე *(იგი აუცილებლად კონტროლერია)* ხდება **Last Name, First Name** და **Username** ატრიბუტების რეპლიცირება. ამასთან, ადმინისტრატორს შეუძლია გლობალურ კატალოგში განათავსოს მომხმარებლების მიერ ხშირად მოთხოვნილი ზოგიერთი ატრიბუტიც.

აუცილებელი არ გახლავთ, მაგრამ ქსელის ეფექტიანი მუშაობის უზრუნველსაყოფად ფრიად სასურველია, მოხდეს გლობალური კატალოგის დუბლირება დომენის სხვა კონტროლერებზეც, განსაკუთრებით მაშინ, როცა ქსელი რამდენიმე კვანძს მოიცავს. *(გავიხსენოთ, რომ კვანძი არის ქსელში კომპიუტერის განლაგების ისეთი ადგილი, რომელთანაც ფართოზოლიანი შეღწევა TCP/IP ოქმის მეშვეობით ხორციელდება).*

ქსელის მწარმოებლურობის მაღალი დონის მისაღწევად აუცილებელია მის თითოეულ კვანძზე დომენის სულ ცოტა ერთი კონტროლერის არსებობა, რომელიც შეითავსებს გლობალური კატალოგის სერვერის ფუნქციებსაც.

მოთხოვნების მომსახურების ასეთი სქემა უზრუნველყოფს ქსელის რესურსების რაციონალურად გამოყენებას.

ახლა კი, **AZDC1** სერვერი ვაქციოთ გლობალური კატალოგის სერვერად *(შევნიშნოთ, რომ **Delaware** კვანძში განლაგებული **W2kserver1** დუმილით უკვე არის გლობალური კატალოგის სერვერი),* რათა გავაადვილოთ **Active Directory** კატალოგების სამსახურის მონაცემთა ბაზაში ინფორმაციის ძებნა *(ამ ბაზაში შესვლა უკვე შესაძლებელი იქნება **Arizona** კვანძიდანაც).*

ამ მიზნით:

- გამოვიძახებთ **Start → Programs → Administrative Tools → Active Directory Sites and Services** ინსტრუმენტს;
- **Sites** საქაღალდეში ვირჩევთ **Arizona**-ს, **Server** საქაღალდეში – **AZDC1**-ს, აქ კი **NTDS Settings**-ისთვის კონტექსტურ მენიუში – **Properties**-ს;
- **General** ჩანართში ვაყენებთ ალამს **Global Catalog, OK.**

სერვერების ტიპები

Windows 2000 Server შეიძლება ფუნქციონირებდეს განცალკევებულ *(ავტონომიურ)* სერვერზე, დომენის წევრ რიგით სერვერზე და დომენის კონტროლერზე.

ავტონომიური სერვერი ინახავს საკუთარ *(ლოკალურ)* მონაცემთა ბაზას და განკარგავს საკუთარი ობიექტების უსაფრთხოებას.

საკუთარ მონაცემთა ბაზას ინახავს დომენის წევრი რიგითი სერვერიც. მაგრამ მას, დამატებით, შეუძლია ნებართვების შემოწმება დომენის რესურსებთან შეღწევაზეც, რისთვისაც იგი მიმართავს დომენის მონაცემთა ბაზას. დომენის რიგითი წევრი სერვერი განკარგავს როგორც საკუთარი, ასევე დომენის ობიექტების უსაფრთხოებასაც.

დომენის კონტროლერი შეიცავს ინფორმაციას მთელი დომენის ობიექტების შესახებ და, ცხადია, მათ უსაფრთხოებასაც განკარგავს.

მომხმარებელთა სააღრიცხვო ჩანაწერები ლოკალური მომხმარებლებისთვის *(ავტონომიურ სერვერზე)*

ლოკალურ მონაცემთა ბაზაში ინახება მომხმარებელთა ლოკალური სააღრიცხვო ჩანაწერები *(ისინი შეიცავენ ინფორმაციას ლოკალური მომხმარებლების შესახებ, მათ პაროლებს).*

ავტონომიურ სერვერზე ოპერაციული სისტემის დაყენებისას თავდაპირველად დუმილით იქმნება ორი ჩაშენებული სააღრიცხვო ჩანაწერი: **Administrator** და **Guest**•. შემდგომ შესაძლებელია ადმინისტრატორის ტიპის სხვა ჩანაწერების შექმნაც. რაც შეეხება **Guest** სააღრიცხვო ჩანაწერს, დუმილით იგი გამორთულია.

მომხმარებლის ახალი ლოკალური ჩანაწერის შესაქმნელად:

1. გავცემთ ბრძანებას

   Start → Programs → Administrative Tools → Computer Management;

• *ოპერაციული სისტემის დაყენებისას სისტემის პარამეტრების შერჩევით შესაძლებელია სხვა სააღრიცხვო ჩანაწერების შექმნაც.*

2. **System Tools → Local Users and Groups** ბრძანების გაცემის შემდეგ მენიუდან ვირჩევთ **Action → Create Users;**
3. **Create User** დიალოგურ ფანჯარაში შეგვაქვს შესაბამისი ინფორმაცია, მაგალითად, **Joe User-**ის შესახებ; **Create, Close.**

სააღრიცხვო ჩანაწერის პარამეტრებს ჩვენ უკვე გავეცანით **Windows NT Server 4.0-**ის შესწავლისას. შემდგომში შესაძლებელია ამ პარამეტრების ჩათვალიერება და საჭიროების შემთხვევაში მათში კორექტივების შეტანაც.

ლოკალური ჯგუფები (ავტონომიურ სერვერებზე)

ავტონომიურ სერვერზე მხოლოდ ლოკალური ტიპის ჯგუფები გამოიყენება. ამ ჯგუფებს *(მაშასადამე, მათ წევრებსაც)* შეიძლება მიენიჭოთ განსაკუთრებული უფლებები.

ავტონომიურ სერვერზე ლოკალური ჯგუფის წევრი შეიძლება იყოს მხოლოდ ლოკალური ტიპის ჩანაწერი. ლოკალური სააღრიცხვო ჩანაწერები და ლოკალური ჯგუფები მხოლოდ იმ კომპიუტერებზე გამოიყენება, რომლებზეც ისინი შექმნეს.

ლოკალური ჯგუფები ორი სახისაა:

- მომხმარებლის ჯგუფები. მათ ქმნის ქსელის ადმინისტრატორი;
- ჩაშენებული ჯგუფები. ასეთი ჯგუფები ავტომატურად იქმნება და მათ წევრებს დუმილით ენიჭება პრივილეგიები. ეს პრივილეგიები, ცხადია, ვრცელდება ჯგუფში შემდგომ დამატებულ მომხმარებლებზეც, ოღონდ ასეთი მომხმარებელი სისტემაში ხელახლა უნდა შევიდეს.

ჩაშენებული ჯგუფები (ავტონომიურ სერვერზე)

ავტონომიურ სერვერზე **Computer Management** ფანჯარაში ორჯერ დავაწკაპუნოთ **Local Users and Groups → Groups** საქაღალდეზე. ეკრანის მარჯვენა მხარეს აისახება ჩაშენებული ჯგუფები:

- **Administrators**. ამ ჯგუფის წევრები ყოველგვარი უფლებებით სარგებლობენ.
- **Backup Operators** *(არქივის ოპერატორები).* ჯგუფის წევრების მოვალეობაა სისტემური და მომხმარებელთა ფაილების სარეზერვო კოპირება. ხშირად ამ ჯგუფის წევრებს ეძლევათ სერვერის გამორთვის უფლებაც.
- **Guests** *(სტუმრები).* ამ ჯგუფით, როგორც წესი, არ სარგებლობენ.
- **Power Users** *(გამოცდილი მომხმარებლები).* მის წევრებს უფლება აქვთ შექმნან მომხმარებელთა სააღრიცხვო ბარათები და შეცვალონ სამუშაო მაგიდის პარამეტრები. მათვე ძალუძთ გაანადგურონ ის სააღრიცხვო ბარათები, რომლებიც თვითონვე შექმნეს.
- **Replicator.** რეპლიკაციის ჯგუფის წევრები განაგებენ სერვერებს შორის ფაილების რეპლიკაციის საქმეს. საჭირო სააღრიცხვო ჩანაწერები ამ ჯგუფში ავტომატურად ხვდება.
- **Users** *(მომხმარებლები).* ყოველი ახალი სააღრიცხვო ჩანაწერი ავტომატურად განთავსდება ამ ჯგუფში. ეს რიგითი მომხმარებლები არავითარი განსაკუთრებული უფლებებით არ სარგებლობენ.

ოპერაციული სისტემის ინსტალირებისას პარამეტრების შერჩევით შეიძლება ზოგიერთი სხვა სპეციალური ჩაშენებული ჯგუფის შექმნაც. ესენია:

DHCP Administrators,

DHCP Users,

WINS Users.

ადმინისტრატორის მიერ შექმნილი ლოკალური ჯგუფები

ვნახოთ, როგორ უნდა მოიქცეს ადმინისტრატორი ლოკალური ჯგუფების შესაქმნელად და მასში, ვთქვათ, **Joe User** მომხმარებლის სააღრიცხვო ჩანაწერის მოსათავსებლად:

1. **Computer Management** ფანჯარაში **System Tools** ხის მომდევნო დონეზე შევდივართ **Local Users and Groups → Groups** საქაღალდეში;
2. **Action** მენიუში გავცემთ ბრძანებას **New Group;**

3. ჯგუფის სახელად ავირჩიოთ **Paymanagers,** შევიტანოთ ინფორმაცია **Description** *(აღწერა)* ველშიც;
4. ჯგუფში მომხმარებლების შესაყვანად დავაწკაპუნოთ **Add** ღილაკზე;
5. სიაში მოვძებნოთ **User** მომხმარებლის სააღრიცხვო ბარათი, მოვნიშნოთ და დავაწკაპუნოთ **Add** ღილაკზე. სააღრიცხვო ბარათი აისახება **New Group** ფანჯრის **Member** არეში. **OK, Create.**

შემდგომ უკვე შეგვეძლება ჩვენ მიერ შექმნილი ჯგუფის განადგურება და მასში მყოფი ცალკეული ჩანაწერების ამოგდებაც.

დომენის კონტროლერებზე მომხმარებლების, ჯგუფების და კომპიუტერების სააღრიცხვო ჩანაწერები

1. მომხმარებლები

დომენის კონტროლერი მთელი დომენისათვის შეიცავს მონაცემების ბაზას, რომლის რეპლიცირება ხდება მოცემული დომენის სხვა კონტროლერებსა და ნდობის გამომცხადებული დომენების კონტროლერებზეც. მონაცემების ბაზაში ცვლილებების შესახებ დომენის სხვა კონტროლერების ინფორმირებისათვის 5 წუთის ვადაა განსაზღვრული. ამ ბაზაში განთავსდება მხოლოდ გლობალური სააღრიცხვო ჩანაწერები. მათი შექმნა, პრინციპში, დომენის ნებისმიერ კონტროლერზე შეიძლება განხორციელდეს.

შევქმნათ მომხმარებლის პირველი სააღრიცხვო ბარათი **corp.com** დომენში:

1. **Start → Programs → Administrative Tools → Active Directory Management;**
2. ვირჩევთ საქაღალდეს **corp.com → Users;**
3. მენიუდან გავცემთ ბრძანებას

   Action → New → User.

ეკრანზე აისახება დიალოგური ფანჯარა **Create New Object (User).**

ლოკალური მომხმარებლის სააღრიცხვო ბარათისაგან განსხვავებით, აქ გამოიყენება ე.წ. <u>მეგობრული სახელი</u>, რომელსაც აქვს შემდეგი სახე:

მომხმარებლის_სახელი@დომენის_სახელი

ვირჩევთ შემდეგ სახელს:

mwier@corp.com

მოცემული სააღრიცხვო ბარათისათვის შეგვაქვს სხვა ინფორმაციაც *(დაწვრილებით შემდგომ).*

ასეთივე წესით ვქმნით სააღრიცხვო ბარათებს შემდეგი მომხმარებლებისთვისაც:

მარტინ ვიერი, ჯონ ჯონსი, ბეტ მარტინსი.

მომხმარებლების შესახებ ინფორმაციის შეტანის შემდეგ შეგვიძლია ამ ინფორმაციის ჩათვალიერება *(მომხმარებლის სააღრიცხვო ბარათზე ორჯერ დაწკაპუნებით).*

აქვე შეგვიძლია მომხმარებლისთვის სახელის გადარქმევაც.

2. ჯგუფები

ავტონომიური სერვერის ანალოგიურად, კონტროლერზეც საქმე გვაქვს ჩაშენებულ და ადმინისტრატორის მიერ შექმნილ მომხმარებელთა ჯგუფებთან.

ამასთან, ჯგუფები შეიძლება იყოს ლოკალური და გლობალური. ოღონდ უნდა დავაზუსტოთ, რომ ესენია დომენის ჯგუფები *(ავტონომიური სერვერისაგან განსხვავებით).*

ამრიგად, საბოლოდ შეიძლება ითქვას, რომ კონტროლერზე გვაქვს დომენის ლოკალური და გლობალური ჯგუფები, რომლებიც, თავის მხრივ, შეიძლება იყოს ჩაშენებული ან ადმინისტრატორის მიერ შექმნილი *(სიმოკლისათვის დასაშვებია, არ გამოვიყენოთ სიტყვა „დომენის“).*

ახლა, გავიხსენოთ, რომ დომენის ლოკალური ჯგუფების სამოქმედო არეალი მშობლიური დომენის ფარგლებითაა შემოზღუდული, გლობალური ჯგუფები კი ჩანს ტყეში შემავალი ყოველი დომენიდან.

გლობალურ ჯგუფში შედის მომხმარებელთა მხოლოდ ისეთი სააღრიცხვო ბარათები, რომლებიც მშობლიურ დომენში შეიქმნა.

რაც შეეხება ლოკალურ ჯგუფებს, მათში შეიძლება გაერთიანდეს არა მარტო ამგვარი ბარათები, არამედ გლობალური ჯგუფებიც ტყეში შემავალი ნებისმიერი დომენიდან.

აღნიშნულ შესაძლებლობას *(ასე ვთქვათ, შემოვლითი გზით სარგებლობას)* ხშირად მიმართავენ ქსელის ადმინისტრატორები, როცა სურთ ერთ დომენში დარეგისტრირებულ მომხმარებლებს მისცენ სხვა დომენის რესურსებით სარგებლობის საშუალება.

(ამ საკითხს უფრო დაწვრილებით ლაბორატორიული სამუშაოების ჩატარებისას გავეცნობით).

დავაზუსტოთ ზოგიერთი მნიშვნელოვანი დეტალი.

ჩვენ უკვე გავეცანით ჩაშენებულ ლოკალურ ჯგუფებს. მათ დუმილით გარკვეული უფლებები აქვთ მინიჭებული.

ალბათ, უკვე შეამჩნიეთ, რომ არსებობს ჩაშენებული გლობალური ჯგუფებიც. ოღონდ ეს არის, რომ ამგვარი ჯგუფები შექმნილია მხოლოდ მომხმარებელთა გასაერთიანებლად. მათთვის არ შეიძლება რაიმე უფლებების დანიშვნა. მაგრამ თუ ეს საჭიროა, გლობალურ ჯგუფს ჩართავენ შესაბამისი უფლების მქონე ლოკალურ ჯგუფში.

ჩამოვთვალოთ ჩაშენებული გლობალური ჯგუფები:

- **Domain Admins** - დომენის ადმინისტრატორების ჩაშენებული გლობალური ჯგუფი. იგი არის **Administrators** ჩაშენებული ლოკალური ჯგუფის წევრი.
- **Domain Users** – დომენის მომხმარებლების ჩაშენებული გლობალური ჯგუფი – **Users** ჩაშენებული ლოკალური ჯგუფის წევრია.
- **Guests** - სტუმრების ჩაშენებული გლობალური ჯგუფი – **Guest** ჩაშენებული ლოკალური ჯგუფის წევრია.

უნივერსალური ჯგუფები

დომენის ლოკალური და გლობალური ჯგუფების გარდა, არსებობს ჯგუფების კიდევ ერთი ტიპი: უნივერსალური.

უნივერსალური ჯგუფი გახლავთ ლოკალური ჯგუფი ზედომენის სახით წარმოდგენილი დომენების მთელი ტყისათვის.

უნივერსალური ჯგუფები დიდძალ ქსელურ რესურსებს საჭიროებს, რადგანაც მათ შემადგენლობაში ცვლილებების შესახებ ინფორმაცია რეპლიცირდება გლობალური კატალოგის შემცველ ყველა სერვერზე.

საუბარია ისეთი სახის ცვლილებებზე, როგორიცაა უნივერსალური ჯგუფების შემადგენლობაში მომხმარებლის ან გლობალური ჯგუფის დამატება-ამოგდება. ოღონდ აქ არ

იგულისხმება ცვლილებები თვით გლობალური ჯგუფის შემადგენლობაში და რადგანაც ასეთი სახის ცვლილებების რეპლიცირება არ ხდება, ადმინისტრატორები წარმატებით იყენებენ შემდეგ ხერხს:

<u>როცა საჭიროა მომხმარებელთა უფლებების შეცვლა, ისინი გადაჰყავთ ერთი გლობალური ჯგუფიდან მეორეში.</u>

უნივერსალური ჯგუფები მხოლოდ მაშინ გამოიყენება, როცა ქსელი მუშაობს არა ე.წ. <u>შერეულ</u> რეჟიმში, არამედ – <u>საკუთარში</u>. *(შერეულია რეჟიმი, როცა ზოგიერთ სერვერზე ფუნქციონირებს ძველი ქსელური ოპერაციული სისტემა Windows NT 4.0).*

ასეთ შემთხვევაში ფუნქციონირებს ორი უნივერსალური ჯგუფი:

Enterprise Admins - საწარმოს ადმინისტრატორები;

Schema Admins - სქემების ადმინისტრატორები.

პირველთაგან განსხვავებით, სქემების ადმინისტრატორების უფლებები რამდენადმე შეზღუდულია *(მათ შეუძლიათ ცვლილებების შეტანა მხოლოდ კატალოგების სამსახურის სქემებში).*

როცა ქსელი საკუთარ რეჟიმში მუშაობს, შესაძლებელია ლოკალური და გლობალური ჯგუფების უნივერსალურ ჯგუფად გარდაქმნა. აღვნიშნოთ, რომ უკუპროცესი ვერ განხორციელდება.

ლაბორატორიული სამუშაოების შესრულებისას გავეცნობით, თუ როგორ ხდება სხვადასხვა ობიექტებისთვის *(ჩვენ მიერ დარეგისტრირებული მომხმარებლებისათვის)* ნებართვების გაცემა, მაგალითად, ძვირადღირებული ფერადი პრინტერის გამოყენებაზე. *(იგულისხმება, რომ ასეთი პრინტერი ქსელში უკვე დაყენებული გვაქვს).* გამოვიყენებთ ორ ხერხს:

1. მომხმარებლებს გავაწევრიანებთ **Server Operators** ჩაშენებულ ჯგუფში;
2. ვისარგებლებთ ე.წ. „1-2-3“ მეთოდით. ჯერ შევქმნით გლობალურ ჯგუფს, რომლის წევრებად გავხდით იმავე მომხმარებლებს. შემდეგ შევქმნით ლოკალურ ჯგუფს, რომლის წევრებსაც ნებართვას მივცემთ საჭირო რესურსთან შეღწევაზე. დაბოლოს, აღნიშნული ლოკალური ჯგუფის წევრად ვაქცევთ ჩვენ მიერ შექმნილ გლობალურ ჯგუფს.

კომპიუტერების სააღრიცხვო ბარათები

დომენის წევრ სერვერ-კომპიუტერებისა თუ სამუშაო სადგურებისათვის აუცილებელია საკუთარ დომენში დარეგისტრირება. კომპიუტერის დარეგისტრირება, მომხმარებლების დარეგისტრირების ანალოგიურად, აქაც გულისხმობს სააღრიცხვო ბარათის შექმნას.

მოცემული კომპიუტერის შემცველი დომენისა თუ სამუშაო ჯგუფის შესახებ ცნობების მოძიება შესაძლებელია **Control Panel**-ში **System** აპლეტის **Network Identification** ჩანართზე.

თუ კომპიუტერი ჯერ კიდევ არ გახლავთ დომენის წევრი, ჩანართზე აისახება **Change** *(შეცვლა)* და **Advanced** *(დამატებით)* ღილაკები.

დომენის წევრი რიგითი სერვერისათვის ეკრანზე ამ ღილაკების თავზე გამოვა წარწერა - **Member of Domain** *(დომენის წევრი)*, ხოლო კონტროლერისთვის კი ეს ღილაკები აღარ აისახება.

Change ღილაკზე დაწკაპუნებით ხდება ოსტატის გაშვება. ვისარგებლოთ ამ ოსტატის მომსახურებით და **AZWORKGRP** სამუშაო ჯგუფის წევრი **AZSERV1** ავტონომიური სერვერი მივუერთოთ **corp.com** დომენს:

1. **Change** ღილაკის მეშვეობით გამოყვანილ ფანჯარაში ვაწკაპუნებთ **Next**-ზე;
2. **Connecting to the Network** ფანჯარაში ვეთანხმებით მუშაობის შემოთავაზებულ ვარიანტს, **Next**;
3. ასევე, არ ვცვლით დომენთან მიერთების შემოთავაზებულ ვარიანტს სხვაზე, **Next**;
4. შემდეგ ვირჩევთ დომენისა და მომხმარებლის სახელებს *(მოცემულ შემთხვევაში CORP და Administrator-ს).* ცხადია, შეგვაქვს ადმინისტრატორის პაროლიც;
5. მომდევნო ფანჯარაში კომპიუტერისა და დომენის სახელებზე დასტურის მიცემის შედეგად იქმნება კომპიუტერის სააღრიცხვო ბარათი, **Next**;
6. ეკრანზე აისახება დიალოგური ფანჯარა **Domain User and Password**. აქ საჭიროა განვსაზღვროთ იმ მომხმარებლის სახელი და პაროლი, რომელსაც უფლება ექნება **CORP** დომენში დაარეგისტრიროს სხვა კომპიუტერებიც. შეგვყავს

ადმინისტრატორის სახელი (Administrator) და ჩვენ მიერ უკვე განსაზღვრული მისი პაროლი, **OK.**

7. მომდევნო **Access Level** *(შეღწევის დონე)* დიალოგურ ფანჯარაში ვეთანხმებით შემოთავაზებულ ვარიანტს – **CORP** სააღრიცხვო ბარათი თავსდება **CORP** დომენის **Administrators** ჯგუფში. შედეგად, ამ დომენის ადმინისტრატორებს შეეძლებათ **AZSERVER1** სერვერის მართვა.
8. ბოლო დიალოგურ ფანჯარაში საჭიროების შემთხვევაში შევძლებთ ჩვენ მიერ არჩეული პარამეტრების მნიშვნელობის კორექტირებას. **Finish** და ვახდენთ კომპიუტერის გადატვირთვას.

ამრიგად, ჩვენ შევქმენით კომპიუტერის სააღრიცხვო ბარათი **AZSERVER1** სერვერისთვის. მას უკვე შეუძლია დომენის კონტროლერთან ურთიერთობის დამყარება.

შევამოწმოთ ჩატარებული სამუშაოს შედეგი:

1. **W2kserver** კონტროლერზე მოვძებნოთ **AZSERVER1** სერვერის სააღრიცხვო ბარათი, რისთვისაც კონტროლერის დისპლეიზე ავსახოთ **Aitive Directory Users and Computers** ფანჯარა;
2. შევდივართ ჯერ **corp.com**, შემდეგ კი **Computers** საქაღალდეში.
3. მარჯვენა პანელზე ასახულია **AZSERVER1** – დომენის წევრი სერვერის სახელი.

კომპიუტერების სააღრიცხვო ბარათების შექმნის ალტერნატიული ხერხი

კომპიუტერის დომენში გაწევრიანება უშუალოდ კონტროლერზეც შეიძლება მოხდეს. ამასთან, საჭირო აღარაა ადმინისტრატორის შესახებ მონაცემების ჩვენება რიგით სერვერსა თუ სამუშაო სადგურზე.

აი, როგორ ჩატარდება ეს ოპერაცია, მაგალითად, **martinspc** სამუშაო სადგურისთვის:

1. **W2Kserver** კონტროლერის ეკრანზე გამოგვყავს **Active Directory Users and Computers** დიალოგური ფანჯარა;
2. ვაღებთ ჯერ **corp.com** და შემდეგ მასში მყოფ **Computers** საქაღალდეებს;
3. **Action** მენიუდან ვირჩევთ ბრძანებას **New → Computer;**

გამოდის დიალოგური ფანჯარა **Create New Object-Computer**, რომლის **Computer Name** ტექსტურ ველში შეგვყავს სახელი **martinspc;**

4. დუმილით, მხოლოდ ადმინისტრატორის დონის მომხმარებლებს შეუძლიათ დომენისათვის კომპიუტერის დამატება. გამონაკლისის წესით, ეს უფლება მივანიჭოთ მისტერ მარტინსაც დომენისათვის საკუთარი კომპიუტერის მისაერთებლად;
5. ვაწკაპუნებთ **Change** ღილაკზე. ვირჩევთ მომხმარებელ მარტინის სააღრიცხვო ბარათს. **Add, Ok, Next;**
6. მომდევნო დიალოგურ ფანჯარაში ჯერჯერობით არაფერს ვცვლით. **Next;**
7. ბოლო ფანჯარაში შეგვიძლია მოვახდინოთ პარამეტრების მნიშვნელობების კორექტირება. **Ok.**

ამის შემდეგ, მარტინს შეეძლება, ოსტატის მეშვეობით დომენს მიუერთოს საკუთარი კომპიუტერი.

შენიშვნა: **Change** ღილაკის გვერდით მყოფ **Advanced** ღილაკით გამოყვანილ ფანჯარაში შესაძლებელია სახელის შეცვლა როგორც სამუშაო ჯგუფის წევრ კომპიუტერისათვის, ასევე რიგითი სერვერისთვისაც *(უკანასკნელ შემთხვევაში საჭიროა ადმინისტრატორის უფლებამოსილება).* დომენის კონტროლერისთვის სახელის შეცვლა არ დაიშვება.

Active Directory კატალოგების სამსახურის ობიექტები

უფრო დაწვრილებით განვიხილოთ კატალოგების სამსახურით სარგებლობასთან დაკავშირებული ზოგიერთი საკითხი. გავეცნოთ მის სხვა, აქამდე ჩვენთვის უცნობ შესაძლებლობებსაც.

დავიწყოთ **Active Directory**-ში ობიექტების პუბლიკაციის საკითხის შესწავლით.

Active Directory-ში ობიექტის პუბლიკაცია ანუ გამოქვეყნება აღნიშნავს ისეთ ქმედებას, რომლის შედეგადაც ობიექტი ხელმისაწვდომი ხდება ყოველი მომხმარებლისათვის.

როდესაც სხვადასხვა ობიექტებს ვქმნიდით (მომხმარებლები, ჯგუფები, კომპიუტერები, დომენები და სხვ.), ფაქტობრივად, ჩვენ ვახდენდით მათ ავტომატურ პუბლიკაციასაც **Active Directory**-ში.

მაგრამ არსებობს ისეთი ობიექტებიც, რომელთა ავტომატური პუბლიკაცია არ ხდება. მაგალითად შეიძლება მოვიყვანოთ იმ კომპიუტერებზე შექმნილი ობიექტები, რომლებიც იმართება **Windows 95, 98,** ან **NT** ოპერაციული სისტემის მეშვეობით, ასევე ობიექტები, რომლებიც **Windows 2000 Server**-ით მართულ კომპიუტერზე შეიქმნა მის დომენში გაერთიანებამდე.

საჭიროების შემთხვევაში ამგვარ ობიექტებს **Active Directory**-ში მომხმარებელიევე გამოაქვეყნებს. მაგალითად, ჩავატაროთ ეს სამუშაო "ხელით" პრინტერისათვის, რომლის მისამართია **\\buster\hp3d:**

*(აქ **buster** არის კომპიუტერის სახელი, **hp3d** კი – პრინტერის).*

1. **Active Directory Users and Computers** ფანჯარაში მივმართავთ საქაღალდეს **corp.com;**
2. მასში ვირჩევთ **Users** საქაღალდეს და გავცემთ ბრძანებას:

   Action → New → Printer
3. პრინტერის სახელად შეგვყავს **\\buster\hp3d, OK.**
4. ახალი პრინტერის სახელი აისახება **Users** საქაღალდეში.
5. ობიექტზე ორჯერ დაწკაპუნებით ეკრანზე გამოვა ინფორმაცია მისი თვისებების შესახებ. რადგანაც ეს პრინტერი-ობიექტი შეიქმნა არა **Windows 2000**-ით მართულ კომპიუტერზე, საჭიროა მისი თვისებების შესახებ ინფორმაცია **Active Directory** მონაცემთა ბაზაში ჩვენ თვითონ შევიტანოთ. კერძოდ, აუცილებელია მოვნიშნოთ ალამი **Double Sided Directory** *(მიუთითებს, რომ პრინტერი მუშაობს ორმხრივ რეჟიმში).*
6. შეგვაქვს ინფორმაცია პრინტერის ფაქტობრივი ადგილსამყოფლის შესახებ და ზოგიერთი სხვა მონაცემიც. **OK.**

დაბოლოს, აღვნიშნოთ, რომ საერთო საქაღალდის გამოქვეყნებაც ამგვარადვე ხდება. მხოლოდ, ობიექტის ასარჩევად გავცემთ ბრძანებას:

New → Shared Folder

Active Directory-ში ობიექტების ძებნა

Windows 2000 იძლევა ქსელში რაიმე, განსაზღვრული თვისებების მქონე ობიექტების მოძებნის შესაძლებლობას. ასეთი ობიექტის როლში შეიძლება მოგვევლინოს: ფერადი ან ორმხრივი ბეჭდვის უნარის მქონე პრინტერები, თანამშრომელი, რომლის ტელეფონის ნომრის გაგება გვსურს და სხვ.

ძებნის პროცესის ჩასატარებლად გავცემთ ბრძანებას **Start → Search** და დიალოგის ფანჯარაში ვახდენთ შესაბამის არჩევანს.

ორგანიზაციული ერთეულების დაპროექტება

ორგანიზაციული ერთეულები საშუალებას იძლევა, დომენის სტრუქტურამ მაქსიმალური სიზუსტით ასახოს დაწესებულების *(ორგანიზაციის)* სტრუქტურა.

ორგანიზაციული ერთეულების ყველაზე ფართოდ გამოყენებული სახეებია *(მოდელებია):*

- *ბიზნეს-განყოფილებების მოდელი*
- *გეოგრაფიული მოდელი*
- *ადმინისტრაციული მოდელი*

პირველი მოდელი "ზემოდან-ქვემოთ" არქიტექტურის შესატყვისია, რომელიც მიღებულია კორპორაციების უმეტესობაში.

გეოგრაფიულ მოდელში აქცენტი გადატანილია ობიექტის ფიზიკურ ადგილმდებარეობაზე.

ადმინისტრაციული მოდელის ორგანიზაციული ერთეულები იქმნება კონკრეტული ფუნქციების მიხედვით. მაგალითად, კორპორაციის განყოფილებებს ემსახურება დამლაგებელი თუ პროგრამისტთა ჯგუფი, რომლებიც "საკუთარ" უფროსს ემორჩილებიან *(პირს, ზემდგომ სტრუქტურულ ერთეულს).*

დავეყრდნოთ პირველ მოდელს და დომენში შევქმნათ ორგანიზაციული ერთეული:

1. კვლავ შევიდეთ **corp.com** საქაღალდეში;
2. გავცეთ ბრძანება:

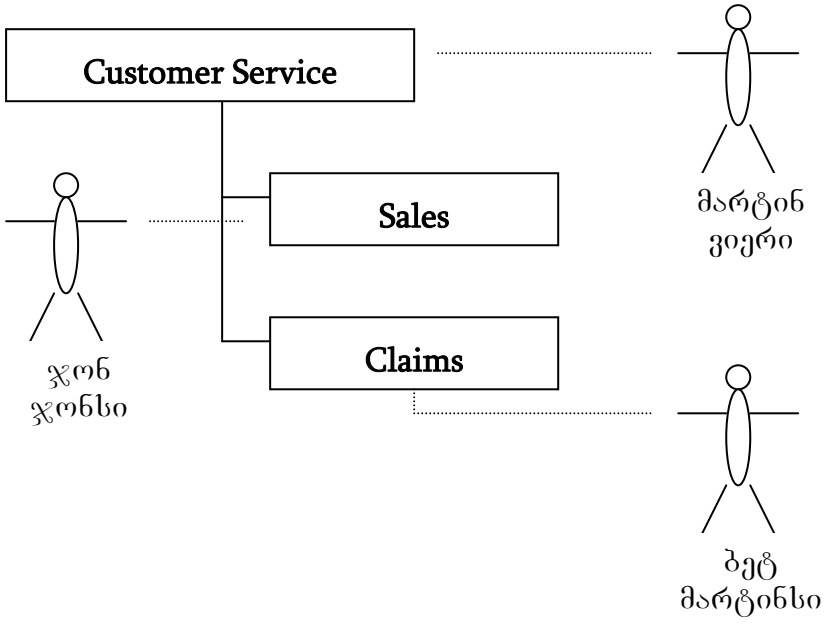
   Action → New → Organization Unit
3. ორგანიზაციული ერთეულის სახელად ავირჩიოთ: **Customer Service, OK;**

4. შევდივართ ახლახან შექმნილ საქაღალდეში და ზემოთ მოყვანილი წესებით ამჯერად მასში ვქმნით **Sales** და **Claims** ორგანიზაციულ ერთეულებს.

ამის შემდეგ ჩვენ მიერ ადრე შექმნილ, მომხმარებლის სამ სააღრიცხვო ბარათს გადავიტანთ შესაბამის ორგანიზაციულ ერთეულებში.

ჩავატაროთ ეს ოპერაციები მენიუს **Action** პუნქტის შესაბამისი ქვეპუნქტების გამოყენებით.

საბოლოოდ, მივიღებთ შემდეგი სახის ორგანიზაციულ სტრუქტურას:



ორგანიზაციული ერთეულის შექმნას აზრი მაშინ აქვს, თუკი მას რაიმე უფლებები დაენიშნება *(განისაზღვრება მისთვის უსაფრთხოების პარამეტრები).* მაგრამ უფლებების მინიჭება მხოლოდ გლობალური ჯგუფებისათვის შეიძლება. მაშასადამე, უნდა მოხდეს ორგანიზაციული ერთეულისათვის გლობალური ჯგუფის შექმნა *(რაც ჩვენ უკვე შევისწავლეთ)* და ამ ჯგუფში ორგანიზაციული ერთეულის ყველა წევრის *(ან ნაწილის მაინც)* გაერთიანება *(მათი სააღრიცხვო ბარათების კოპირების გზით).*

საინტერესოა, რომ კონკრეტული ორგანიზაციული ერთეულის მართვის შესაძლებლობები ადმინისტრატორის მიერ შეიძლება მთლიანად ან ნაწილობრივ დელეგირებული იქნეს რომელიმე მომხმარებლისა თუ ჯგუფისადმი. ეს ხდება მონიშნული ორგანიზაციული ერთეულისთვის **Action → Delegate Control**

ბრძანების გაცემით და დიალოგის მომდევნო ფანჯრებში პირისა და მისთვის მინიჭებული უფლებების ჩვენებით.

დისკოები და ტომები

კომპიუტერის გარე მეხსიერების მართვისათვის გავცემთ ბრძანებას:

Start → Programs → Administrative Tools → Computer Management → Strogare *(მონაცემთა საცავი)* და ორჯერ ვაწკაპუნებთ ქვეხეზე მყოფ **Disk Management** პიქტოგრამაზე.

მარჯვნივ, ქვეფანჯარაში დისკოების შესახებ ინფორმაცია აისახება როგორც გრაფიკულად, ასევე სიის სახითაც.

თავიდანვე აღვნიშნოთ, რომ **Windows 2000** იყენებს ორი ტიპის დისკოებს:

ბაზურ დისკოს, რომელიც შეიძლება შეიცავდეს ერთიდან ოთხამდე განყოფილებას. მათგან მხოლოდ ერთი შეიძლება იყოს ე.წ. გაფართოებული განყოფილება, დანარჩენები კი *(მინიმუმ ერთი)* იქნება პირველადი განყოფილებები.

დინამიკურ დისკოს, რომელიც წარმოადგენს ერთ დისკოს ან რამდენიმე მათგანის ერთობლიობას. ამ დისკოების ბაზაზე იქმნება სხვადასხვა ტიპის ტომები:

- *ერთი ტომი,*
- *შედგენილი,*
- *სარკული,*
- *მონაცვლეობითი,*
- ***RAID-5*** *ტიპის და სხვ.*

ბაზურ დისკოებს წინა ვერსიების ოპერაციული სისტემებიც იყენებდნენ, დინამიკური დისკოები კი მხოლოდ **Windows 2000** ოპერაციული სისტემისათვის არის დამახასიათებელი. დასახელებიდანვე ჩანს, რომ დინამიკური ტომების ზომა შეიძლება იცვლებოდეს. ამასთან, მათ ახასიათებთ რიგი თავისებურებებისა *(რომელთა შესახებაც ქვემოთ გვექნება საუბარი).*

აღვნიშნოთ, რომ ბაზური დისკო შეიძლება გარდაიქმნეს დინამიკურ დისკოდ *(და პირიქით).* უნდა გავითვალისწინოთ, რომ ასეთი სახის გარდაქმნა მოითხოვს 1 მგბაიტი ზომის ისეთ თავისუფალ მეხსიერებას, რომელიც არც ერთ განყოფილებას არ ეკუთვნის.

თუ ბაზური განყოფილება რომელიმე სხვა ოპერაციული სისტემის ჩატვირთვისთვის არის განკუთვნილი, ტიპის გარდაქმნის შემდეგ ის ამ თვისებას დაჰკარგავს.

ბაზური დისკოს გარდაქმნა დინამიკურად ამგვარად ხორციელდება:

1. არჩეული დისკოსათვის გამოგვყავს კონტექსტური მენიუ. გავცემთ ბრძანებას:

   Upgrade to Dinamic Disk *(განვაახლოთ დინამიკურ დისკოდ);*

2. თუ დისკო შეიცავს ჩატვირთვად განყოფილებას, ვიღებთ შესაბამის გაფრთხილებას;
3. ვაწკაპუნებთ **Upgrade** *(განახლება)* ღილაკზე.

შედეგად, ბაზური დისკოს პირველადი და გაფართოებული განყოფილებები *(მათ შორის სისტემური და ჩატვირთვადი განყოფილებები, ლოგიკური დისკოები)* გარდაიქმნება დინამიკური დისკოს მარტივ ტომებად.

შევნიშნოთ, რომ არსებობს სხვა, უფრო იშვიათი ვარიანტებიც, მაგალითად, **Windows NT** კრებულების გარდაქმნის სახით.

სისტემაში დისკოების დამატება-ამოგდება

კომპიუტერში დისკოს დამატებისას იგი დაყენდება როგორც ბაზური. დისკოების დამატება-ამოგდებისას გამოიყენება ბრძანება **Action → Rescan Disks.** თუ **Windows 2000 Server**-მა დისკოების ახალი კონფიგურაცია ვერ გამოიცნო, კომპიუტერი უნდა გადაიტვირთოს.

დამატებული დისკო **Disk Management**-ში აისახება, როგორც "გარეშე" დისკო. მისთვის გამოძახებულ კონტექსტურ მენიუში გავცემთ ბრძანებას: **Import Foreign Disk.** პროგრამა-ოსტატი დაგვეხმარება დისკოს იმპორტირებაში.

სხვა კომპიუტერიდან ტომების სხვადასხვა ტიპის ერთობლიობის გადმოსატანად:

1. მათი ფიზიკური გადმოტანის შემდეგ ნებისმიერი დისკოსთვის გამოძახებულ კონტექსტურ მენიუში ავირჩევთ ბრძანებას:

   Add Disk

   ეკრანზე აისახება დისკოების ჯგუფი.

2. **Select Disk**-ით კრებულიდან ვირჩევთ საჭირო დისკოს.

განყოფილებები

განყოფილება წარმოადგენს ფიზიკური დისკოს დაფორმატებულ უბანს. დაფორმატების პროცესში განყოფილებაში ფორმირდება ფაილური სისტემა (**FAT**, **FAT32** ან **NTFS**) და ფაილების განლაგების ცხრილი.

განყოფილება შეიძლება შეიცავდეს როგორც სისტემურ, ასევე მომხმარებელთა ფაილებს.

სისტემური ეწოდება ისეთი სახის პირველად განყოფილებას, რომელიც სისტემურ ფაილებს შეიცავს.

თუ პირველადი განყოფილება ოპერაციული სისტემის ჩატვირთვისათვის განკუთვნილ ფაილებს ინახავს, მაშინ მას ეწოდება ჩამტვირთავი განყოფილება. ამ განყოფილებაში ხდება განყოფილებების დარეგისტრირებაც.

შევქმნათ დისკოზე განყოფილება:

1. კვლავ ორჯერ დავაწკაპუნოთ **Strogare** და შემდეგ ერთხელ **Disk Management** პიქტოგრამებზე, რათა ეკრანზე გამოვიყვანოთ **Disk Management** ფანჯარა.
2. დისკოს თავისუფალ ადგილზე გამოგვყავს კონტექსტური მენიუ და გავცემთ ბრძანებას **Create Volume** *(შექმენი ტომი).* გაიშვება ტომის შემქმნელი პროგრამა-ოსტატი.
3. გადამრთველს ვაყენებთ **Simple Volume** *(მარტივი ტომი)* მდგომარეობაში, **Next.**
4. ვირჩევთ დისკოს და განვსაზღვრავთ ტომის მოცულობას, **Next.**
5. შემდგომ დიალოგის ფანჯარაში ან არაფერს ვცვლით, ანდა ვირჩევთ დისკოსათვის ასო-დასახელებას, **Next.**
6. ვირჩევთ ფაილურ სისტემას და ტომისათვის დამატებით დასახელებას – ჭდეს. *(შესაძლებელია კლასტერის ზომის განსაზღვრაც და თუ ფაილური სისტემა საშუალებას იძლევა, დისკოზე ინფორმაციის შეკუმშულად ჩაწერის პარამეტრების მითითებაც),* **Next.**
7. გამოდის რეზიუმე, **Finish.**

თუ საჭიროა, განყოფილება მოინიშნოს, როგორც აქტიური *(რომლიდანაც სისტემის ჩამოტვირთვა იწყება),* მისთვის გამოძახებულ კონტექსტურ მენიუში ავირჩევთ ბრძანებას:

Mark Partition Active

ნებისმიერი ტიპის განყოფილების ამოგდება ხორციელდება კონტექსტურ მენიუში ბრძანების გაცემით, ოღონდ:

- გაფართოებული განყოფილებისთვის მანამდე აუცილებელია მასში მყოფი ტომებისა და ლოგიკური დისკოების განადგურება.
- შეუძლებელია ჩამტვირთავი განყოფილების და ნებისმიერი სხვა განყოფილების ამოგდება, რომელშიც განლაგებულია აქტიური გადმოქაჩვის ფაილი.

გაფართოებული და დამატებითი განყოფილებები

გაფართოებული განყოფილება ლოგიკური დისკოების და ტომების საცავის როლში გვევლინება. მისი მეშვეობით ხერხდება გვერდი ავუაროთ შეზღუდვას – დისკოზე არა უმეტეს ოთხი განყოფილების არსებობის ლიმიტს.

აღვნიშნოთ, რომ გაფართოებული განყოფილების შექმნისას რეკომენდირებულია, გამოვიყენოთ მთელი თავისუფალი მეხსიერება – წინააღმდეგ შემთხვევაში დარჩენილი მეხსიერების უბნით უკვე ვეღარ ვისარგებლებთ.

პირველადი განყოფილებებისაგან განსხვავებით, გაფართოებული განყოფილებების დაფორმატება ხდება არა შექმნისთანავე, არამედ მისი ლოგიკურ დისკოებად დაყოფისას. ამ დროს განისაზღვრება დისკოებისათვის ასო-დასახელებები და ჭდეები.

როგორც წესი, აქტიური *(ჩამტვირთავი)* განყოფილება განთავსდება **C:** ლოგიკურ დისკოზე. ფიზიკურ დისკოზე თუ სამი პირველადი და ერთი გაფართოებული განყოფილებაა *(ან ოთხივე პირველადია),* მათ ტომის სახელად ენიჭება **C:,D:, E:, F:,** თუმცა შესაძლებელია სხვა არჩევნის გაკეთებაც.

რადგანაც კომპიუტერი განყოფილებებს იყენებს, როგორც დამოუკიდებელ ფიზიკურ დისკოებს, ცხადია, მათთვის მინიჭებული სახელები უნიკალური უნდა იყოს.

კონტექსტური მენიუდან შესაძლებელია დისკოს *(ან ტომის)* მაიდენტიფიცირებელი ასოს შეცვლა *(გამონაკლისია ჩამტვირთავი დისკო).*

ტომები

ტომის არსს რომ კარგად ჩავწვდეთ, გავიხსენოთ, რომ ლოგიკური დისკოები შემოზღუდულია ფიზიკური დისკოს ფარგლებით, გარდა ამისა, მათ შორის "საზღვრების გადაწევა" პრობლემებთან არის დაკავშირებული. ტომი კი ამ შეზღუდვებისაგან თავისუფალია. ერთი ტომი შეიძლება რამდენიმე ფიზიკურ დისკოს *(უფრო ხშირად კი მის ცალკეულ ნაწილებს)* მოიცავდეს. ამასთან, ტომები უზრუნველყოფს მუშაობის დინამიკურ რეჟიმს, რაც შემდეგ უპირატესობებს იძლევა:

რადგანაც შეტანა-გამოტანის ოპერაციების დროს ერთდროულად დისკოს რამდენიმე თავაკი მუშაობს, მკვეთრად იზრდება ინფორმაციის გადაცემის სიჩქარე;

იზრდება მეხსიერების მოცულობის კორექტირების შესაძლებლობები – ტომს ადვილად შეგვიძლია დავუმატოთ დისკოებზე არსებული *(გამოთავისუფლებული)* ე.წ. მარტივი ტომები *(მარტივი ტომი იდენტურია განყოფილების, ოღონდ იმ განსხვავებით, რომ განყოფილების მიერთება სხვა "კონგლომერატის" მიერ გაუშვებელია);*

მტყუნებებისადმი მდგრადობა, რასაც განაპირობებს მონაცემთა სიჭარბე *(სარკული კოპირება, მონაცემთა შემოწმება, იხ. ქვემოთ);*

მუშაობის პროცესშივე მწყობრიდან გამოსული *(ან გამორთული)* დისკოების შეცვლა.

მარტივი ტომები

მარტივი ტომების საზღვრები შეიძლება მთელ დისკოს მოიცავდეს. ასეთი ტომები მაქსიმალურად იყენებს მეხსიერებას მოცემული ფაილური სისტემისათვის, მაგრამ, სამაგიეროდ, ვერ უზრუნველყოფს მტყუნებებისადმი მდგრადობას, რაც დამატებით მეხსიერებას მოითხოვს *(იხ. ქვემოთ).*

მარტივი ტომები შეიძლება გამოყენებულ იქნეს, როგორც ერთეულები, მონაცემთა შენახვის უფრო რთული მიდგომების

განსახორციელებლად *(მაგალითად, სარკულ და მონაცვლე ტომებში).*

მარტივი ტომის შექმნა ასე ხდება *(შედგენილისაც):*

- დისკოს თავისუფალ ადგილზე გამოყვანილ კონტექსტურ მენიუში ვირჩევთ ბრძანებას:

  Create Volume

- დიალოგში შევდივართ ოსტატ-პროგრამასთან. იგი წარიმართება ზემოთ განხილული, განყოფილების შესაქმნელი დიალოგის მსგავსად.

შედგენილი ტომები

თუ ორ ან რამდენიმე დისკოს ერთ ლოგიკურ დისკოდ გავაერთიანებთ, მივიღებთ შედგენილ ტომს *(**Windows NT**-ში მსგავსი სტრუქტურა იწოდებოდა ტომების კრებულად).* შედეგად, შეიძლება შევქმნათ მეტად დიდი მეხსიერების, პირობითად უწყვეტი უბანი. აღვნიშნოთ, რომ **Windows 2000 Server** უშვებს შედგენილ ტომში 32-მდე დისკოს გაერთიანების შესაძლებლობას.

შედგენილი ტომი, ვთქვათ, სარკული ტომებისაგან განსხვავებით, არ მოითხოვს მისი შემადგენელი მარტივი ტომების ტოლობას *(მეხსიერების მოცულობის მხრივ),* მაგრამ კრძალავს ტომების ფაილურ სისტემებს შორის განსხვავებულობას – ასეთ შემთხვევაში ზოგი მათგანი თავიდან უნდა დაფორმატდეს.

შედგენილი ტომის შექმნის დიალოგი იმით განსხვავდება მარტივი ტომის შექმნის დიალოგისაგან, რომ აქ გადამრთველი **Simple Volume** მდგომარეობიდან გადაგვყავს **Spanned Volume** მდგომარეობაში. ამასთან, ცხადია, ვირჩევთ მეხსიერების რამდენიმე უბანს.

როგორც მარტივი, ისე შედგენილი ტომების გასაფართოებლად კონტექსტური მენიუდან ვირჩევთ ბრძანებას: **Extend Volume.** მაგრამ შედგენილი ტომისათვის ეს დასაშვებია მხოლოდ **NTFS** ფაილური სისტემით მისი დაფორმატების შემთხვევაში.

შედგენილი ტომის ნებისმიერი ნაწილის ამოგდება მთელი ტომის განადგურებას იწვევს.

Windows 2000 კრძალავს შედგენილი ტომების გამოყენებას ჩასატვირთი და სისტემური ფაილების შესანახად.

RAID ტექნოლოგიები

მარტივი და შედგენილი ტომები, ინფორმაციის უსაფრთხოდ შენახვის თვალსაზრისით, ნაკლებად საიმედოა.

ამ მიზნით, შემუშავებულ იქნა **RAID**-ის სახელით ცნობილი აპარატურული და პროგრამული საშუალებები. შევნიშნოთ, რომ ასეთი აპარატურა გაცილებით ძვირი ჯდება ხისტი დისკოების მასივთან შედარებით, რომლისთვისაც **Windows 2000**–ში გათვალისწინებულია **RAID** ტექნოლოგიის პროგრამული რეალიზაციის შემდეგი სახეები:

- **RAID – 0** – მონაცვლე ტომები *(ტომების კრებული)*
- **RAID – 1** – სარკული ტომები *(ტომების კრებული)*
- **RAID – 5** – მონაცვლე ტომები ლუწობაზე შემოწმებით
- **RAID – 1+0** *(იგივე **RAID – 10**)* – სარკული მონაცვლე ტომები

RAID – 1 – სარკული ტომები

ეს ტექნოლოგია ითვალისწინებს ერთი ტომის ინფორმაციის მეორეზე სრულ – სარკულ კოპირებას. ერთი მათგანის დაზიანების შემთხვევაში შესაძლებელი ხდება მეორეთი სარგებლობა. მაგრამ, ცხადია, მდგომარეობა, რაც შეიძლება მალე უნდა გამოვასწოროთ – ხომ შესაძლებელია, რომ მეორე ტომიც გამოვიდეს მწყობრიდან! ამასთან, ასეთი ტექნოლოგია გვიმტყუნებს მაშინაც, როცა მწყობრიდან გამოდის დისკური მოწყობილობების მომსახურე ერთადერთი ადაპტერი.

სარკული ტომები ასე იქმნება:

1. **Disk Management** ფანჯარაში თაგვის მარჯვენა ღილაკით დავაწკაპუნოთ ტომზე, რომლისთვისაც გვსურს შევქმნათ ასლი და კონტექსტურ მენიუში გავცეთ ბრძანება: **Create Volume**
2. **Create Volume Wizard** ფანჯარაში ვაწკაპუნებთ **Next**-ზე და შემდეგ გადამრთველს ვაყენებთ **Mirrored Volume** *(სარკული ტომი)* მდგომარეობაში.
3. დისკოზე შევირჩევთ არანაკლები ზომის მეხსიერების უბანს, ვიდრე საწყისი ტომის განკარგულებაშია.

4. **Finish**-ზე დაწკაპუნებით იქმნება ტომის სარკული ასლი. შევნიშნოთ, რომ მასზე დარჩენილი გამოუყენებელი მეხსიერება შეუღწევადი გახდება.

არსებული ტომისათვის სარკულის დამატება შესაძლებელია იმავე მენიუდან, **Add Mirror** ბრძანების გაცემით.

თუ სარკული ტომი დაზიანდა, მისი მდგომარეობა იდენტიფიცირდება, როგორც **Failed Redudancy** *(არასაკმარისი სიჭარბის მქონე),* ხოლო ერთ-ერთი დისკოს მდგომარეობა – როგორც **Offline** *(გამორთული).*

სარკული ტომის აღდგენა შესაძლებელია საღი ტომის კონტექსტური მენიუდან **Reactivate Disk** (*დისკოს აღდგენა)* ბრძანების გაცემით.

თუ ერთ-ერთი ტომი გამორთული გვქონდა, კონტექსტური მენიუდან **Resynchronize Mirror** ბრძანების მეშვეობით მოვახდენთ ტომების სინქრონიზებას.

Break Mirror ბრძანებას სარკული ტომები ჩვეულებრივ რეჟიმში გადაჰყავს *(ამ ხერხით ხდება მეხსიერების უბნის გამოთავისუფლება),* ხოლო თუ გვსურს ტომის მეხსიერება ამავე დროს დაუკავებელ სივრცეში გადავიყვანოთ, გავცემთ ბრძანებას:

Remove Mirror

შედეგად, სარკული ტომი განადგურდება.

<u>მონაცვლე ტომები - RAID – 0</u>

მონაცვლე ტომებში *(გავიხსენოთ, **Windows NT**-ში მათ ტომების მონაცვლე კრებულები ეწოდება)* ერთ ლოგიკურ ტომად ერთიანდება მეხსიერების უბნები რამდენიმე ლოგიკური დისკოდან.

ინფორმაციის ჩაწერა ერთდროულად ხდება ყველა დისკოზე *(64 კილობაიტი ზომის სეგმენტებად),* რაც, ცხადია, მნიშვნელოვნად აჩქარებს ამ პროცესს. მაქსიმალური ეფექტი მაშინ მიიღწევა, როცა დისკოებზე გამოყოფილი ადგილები ერთნაირი ზომისაა. **Windows 2000** მოითხოვს, ეს პირობა დაახლოებით მაინც შესრულდეს.

მონაცვლე ტომების შესაქმნელად ასე ვმოქმედებთ:

1. დინამიკური დისკოს თავისუფალ ადგილზე გამოძახებულ კონტექსტურ მენიუში გავცემთ ბრძანებას:

Create Volume

2. **Create Volume Wizard** ფანჯარაში ვაწკაპუნებთ **Next**-ზე და შემდეგ გადამრთველს ვაყენებთ **Stripped Volume** (*მონაცვლე ტომი)* მდგომარეობაში.
3. ვირჩევთ მონაცვლე ტომის შემადგენელ დისკოებს. სისტემა მოითხოვს ყველა დისკოზე დაახლოებით ერთნაირი ზომის უბნების მონიშვნას.
4. შევირჩევთ ფაილურ სისტემას. *(განისაზღვრება ზოგი სხვა პარამეტრიც).* **Finish.**

აღვნიშნოთ, რომ მონაცვლე ტომებს *(ისევე, როგორც შედგენილებს)* ერთი მნიშვნელოვანი ნაკლი აქვს:

<u>ერთი დისკოს დაზიანების შემთხვევაშიც კი იკარგება მთელი ტომის მონაცემები.</u>

<u>მონაცვლე ტომები ლუწობაზე შემოწმებით - RAID–5</u>

სწორედ, ზემოთ აღნიშნული ნაკლის გამოსწორებას ემსახურება **RAID–5** ტექნოლოგია. მისი მეშვეობით გამოითვლება, ფიქსირდება და განუწყვეტლად მოწმდება <u>მონაცემთა საკონტროლო ჯამი</u>. ეს ჭარბი ინფორმაცია უმრავლეს შემთხვევაში შესაძლებლობას იძლევა, აღდგენილ იქნეს მონაცემები და ეს ხდება მაშინაც კი, როდესაც მთელი დისკო გამოდის მწყობრიდან.

RAID–5 ტექნოლოგიის განხორციელება ხდება 3 – 32 დაახლოებით ერთნაირი ზომის დინამიკური ტომისათვის, რისთვისაც:

1. დინამიკური დისკოს თავისუფალ ადგილზე გამოძახებულ კონტექსტურ მენიუში გავცემთ ბრძანებას:

Create Volume

2. **Create Volume Wizard** ფანჯარაში ვაწკაპუნებთ **Next**-ზე და ამჯერად გადამრთველს ვაყენებთ **RAID–5** მდგომარეობაში.
3. აქაც ვირჩევთ დისკოებს, ფაილურ სისტემას და განვსაზღვრავთ სხვა პარამეტრებსაც. **Finish.**

დისკოს დაზიანებისას ტომის ჭდეზე, ჩვეულებრივ, ძახილის ნიშანი აისახება. სიტუაციის გამოსასწორებლად:

1. ტომისთვის გამოძახებულ კონტექსტურ მენიუში გავცემთ ბრძანებას:

   Reactivate Disk *(დისკოს აღდგენა).*

   ფანჯარაში ვაწკაპუნებთ **Next**-ზე და გადამრთველს **RAID–5** მდგომარეობაში ვაყენებთ.
2. თუ დისკოს აღდგენა ვერ მოხერხდა, მას ახლით შევცვლით და ამჯერად მისი კონტექსტური მენიუდან ამასვე შევეცდებით შემდეგი ბრძანებით:

   Repair Disk
3. ოსტატს მივუთითებთ, რომელმა დისკომ უნდა შეცვალოს დაზიანებული მოწყობილობა.
4. ვირჩევთ დისკოებს, ფაილურ სისტემას და განვსაზღვრავთ სხვა პარამეტრებსაც. **Finish.**

"ცხელ" მდგომარეობაში დისკოს შეცვლის ნებართვას ოპერაციული სისტემა მხოლოდ ძვირადღირებული აპარატურისათვის იძლევა. ამის გამო, უმეტეს შემთხვევაში იძულებული ვართ, დისკოს შეცვლის წინ კომპიუტერი გამოვრთოთ.

საინტერესოა, რომ მონაცემების მთლიანობის დარღვევა დისკოს ფიზიკური დაზიანებით არ არის განპირობებული. მაგალითად, თუ ეს მოხდა ელექტროენერგიის მოულოდნელად გამორთვის მიზეზით, ძაბვის ჩართვისას **Windows 2000** ავტომატურად აღადგენს **RAID–5** ტიპის ტომს.

აღვნიშნოთ **RAID–5** ტექნოლოგიის ნაკლიც:

<u>ასეთი ტომის გაფართოება ან სარკული ტიპის სტრუქტურაში ჩართვა შეუძლებელია.</u>

დაბოლოს, შევნიშნოთ, რომ **Windows 2000** ფლობს ინფორმაციის სარეზერვო კოპირების, დისკოებზე ფრაგმენტაციის აღმოფხვრისა და მონაცემთა ე.წ. შორეული შენახვის უზრუნველმყოფ მარტივ და ეფექტიან საშუალებებს.

NTFS ფაილური სისტემა. ნებართვები.

NTFS ფაილური სისტემის გამოყენება მნიშვნელოვნად ამაღლებს სერვერის მუშაობის უსაფრთხოებას. მაგალითად, ჩვენ შევძლებთ, ამა თუ იმ ფაილთან შეღწევის უფლება მივცეთ კონკრეტულ მომხმარებელს ან მომხმარებელთა ჯგუფს – საკმარისია, ისინი შევიყვანოთ ფაილთან *(საქაღალდესთან)* **ACL** (**Access Control List**) დაკავშირებულ სიაში.

ფაილებსა და საქაღალდეებთან მუშაობისათვის გათვალისწინებულია ორი ტიპის ნებართვა:

ძირითადი და **დამატებითი**.

საქაღალდეებისათვის გამოიყენება შემდეგი ნებართვები:

- **Full Control** *(სრული მართვა).* მომხმარებელს შეუძლია ყველა ქვემოთ ჩამოთვლილი და ორი დამატებითი ოპერაციის შესრულება.
- **Modify** *(ცვლილებები).* მომხმარებელს შეუძლია ფაილებისა და საქაღალდეების შექმნა-განადგურება, აგრეთვე სხვა მომხმარებლებისათვის ამ საქაღალდესთან მუშაობაზე დაწესებული ნებართვების გაცნობაც.
- **Read & Execute** *(კითხვა და შესრულება).* მომხმარებელს უფლება აქვს წაიკითხოს და შესრულებაზე გაუშვას ფაილები.
- **List Folder Contents** *(საქაღალდის შემცველობის ჩათვალიერება).*
- **Read** *(კითხვა).* მომხმარებელს შეუძლია წაიკითხოს ფაილები, ამასთან, შეიტყოს, კიდევ ვის აქვს ასეთივე ნებართვა.
- **Write** *(ჩაწერა).* მომხმარებელს შეუძლია შეინახოს ფაილები, ამასთან, შეიტყოს, კიდევ ვის აქვს ასეთივე ნებართვა.

ფაილებისათვის დასაშვებია შემდეგი ნებართვების გამოყენება:

- **Full Control** *(სრული მართვა).* ზედა შემთხვევის ანალოგიურია.
- **Modify** *(ცვლილებები).* მომხმარებელს შეუძლია ცვლილებები შეიტანოს ფაილებში და გაანადგუროს ისინი. დასაშვებია ამ ნებართვის მქონე სხვა მომხმარებლების სიასთან გაცნობაც.

- **Read & Execute** *(კითხვა და შესრულება).* მომხმარებელს შეუძლია წაიკითხოს და შესრულებაზე გაუშვას ფაილები. დასაშვებია ანალოგიური ნებართვის მქონე სხვა მომხმარებლების სიასთან გაცნობაც.
- **Read** *(კითხვა).* წინა პუნქტის იდენტურია, შესრულებაზე ნებართვის გამოკლებით.
- **Write** *(ჩაწერა).* მომხმარებელს შეუძლია შეინახოს ფაილები, ამასთან, შეიტყოს, კიდევ ვის აქვს ასეთივე ნებართვა.

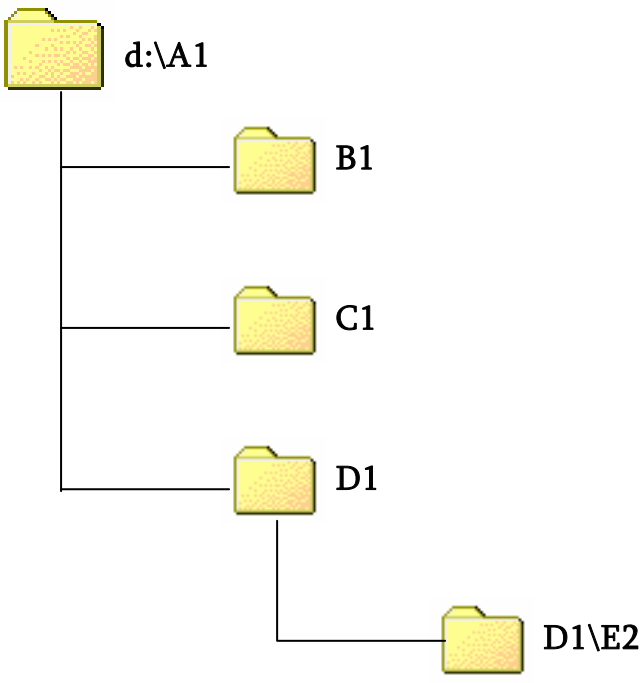
ნებართვების მემკვიდრეობითობა, ჯგუფური ნებართვები.

დუმილით, მშობელ საქაღალდესთან მუშაობაზე გაცემული ნებართვები ვრცელდება შვილობილ საქაღალდეებსა და ფაილებზეც. თუმცა დასაშვებია ამ სიტუაციის შეცვლაც როგორც შვილობილი საქაღალდეებისათვის, ისე ფაილებისთვისაც. მაშინ საქაღალდე უკვე თვითონ განსაზღვრავს შვილობილი ელემენტებისათვის ნებართვების ხასიათს.

თუ მომხმარებელი რამდენიმე ჯგუფის წევრია, ამ ჯგუფებისათვის გაცემული ყველა ნებართვა ძალაში რჩება, ანუ ისინი ჯამდება. გამონაკლისს ადგილი აქვს მხოლოდ ისეთი ფაილისათვის, რომელთანაც შეღწევა აკრძალულია **Deny**-ს მეშვეობით.

NTFS ნებართვების გაცემა

ჯერ, როგორც ადმინისტრატორმა, შევქმნათ საქაღალდეების ასეთი სტრუქტურა **NTFS** ტომზე:



თავდაპირველად გავცეთ ნებართვები **A** საქაღალდეზე **Customer Service** და **Domain Admins** ჯგუფებისათვის:

1. **A** საქაღალდისთვის კონტექსტური მენიუდან გამოვიძახოთ **Properties** ფანჯარა და გადავიდეთ *(უსაფრთხოება)* ჩანართზე.
2. რადგანაც, დუმილით, **d** დისკოს ფესვურ საქაღალდესთან მუშაობისას ყოველგვარი ქმედების განხორციელება შეუძლია **Everyone** *(ყველა მომხმარებელი)* ჯგუფს, ეს უფლებები გადადის ყველა შვილობილ ელემენტზეც.
3. ჩვენი მიზანია ახალი მშობელი საქაღალდის შექმნა. ამიტომ მოვხსნით ალამს:

   Allow Inheritable Permissions from Parent to Propagate to This object

   შედეგად, ეკრანზე აისახება **Security** დიალოგური ფანჯარა.
4. ვაწკაპუნებთ **Remove** ღილაკზე, რათა უარი ვთქვათ მშობელი საქაღალდიდან გადმოსულ ნებართვებზე **Everyone** ჯგუფისთვის.
5. ვაწკაპუნებთ **Add** ღილაკზე. აისახება დიალოგური ფანჯარა **Select Users, Computers or Groups.** ვირჩევთ **Customer Service** და **Domain Admins** ჯგუფებს. **OK.**

6. ვბრუნდებით წინა ფანჯარაში და პირველი გადმოსული ჯგუფისათვის მოვნიშნავთ **Full Control**, ხოლო მეორესათვის – **Read Execute** ალმებს. **OK.**
7. ამის შემდეგ სისტემაში შევდივართ **Customer Service** ჯგუფის წევრ მარტინ ვიერის სახელით *(ვკრეფთ მომხმარებლის სახელს **mwier-ს, პაროლია fastcar6).*** ვაღებთ **D:\A** საქაღალდეს და ვქმნით მასში შემდეგ საქაღალდეებს: **Projects, Appts, Orders** და **Misc.**

ნებართვების გაცემა საქაღალდეებთან მუშაობაზე.

დავუშვათ, გვსურს, **EXECS** ჯგუფის წევრებს ნება დავრთოთ **Projects** საქაღალდის ფაილებში ცვლილებების შეტანაზე. ამისათვის:

1. **Projects** საქაღალდისთვის კონტექსტური მენიუდან გავცემთ **Properties** ბრძანებას და გადავდივართ **Security** ჩანართზე.
2. ვაწკაპუნებთ **Add** ღილაკზე. ეკრანზე გამოსულ **Select Users, Computers or Groups** ფანჯარაში ვირჩევთ **EXECS** ჯგუფს. **OK**-თი ვბრუნდებით **Project Properties** ფანჯარაში.
3. ვირჩევთ **EXECS** ჯგუფს და ვაყენებთ მისთვის **Modify** ალამს. **OK.**

აღვნიშნოთ, რომ **Project** საქაღალდესთან სრული შეღწევის *(ანუ ნებისმიერი ქმედების)* უფლება ავტომატურად მიენიჭა **Customer Service** და **Domain Admins** ჯგუფის წევრებს – **Projects** საქაღალდე ხომ, როგორც შვილობილი, მემკვიდრეობას ღებულობს მშობელი საქაღალდიდან!

ამგვარივე გზით მივანიჭოთ **Misc** საქაღალდის შემცველობის წაკითხვის უფლება **(Read)** ყველა მომხმარებელს ანუ **Everyone** ჯგუფის წევრებს.

ნებართვების გაცემა ფაილებთან მუშაობაზე

Project საქაღალდეში შევქმნათ სამი ფაილი: **Proj1.txt; Proj2.txt; Proj3.txt** და მათთან მუშაობაზე სხვადასხვა სახის ნებართვები გავცეთ ცალკეულ მომხმარებლებსა და მათ ჯგუფებზე, შემდეგი სიის მიხედვით:

ფაილი ***Proj1.txt***

Customer Service ჯგუფი – **Full Control,**
Execs – **Modify**

ფაილი ***Proj2.txt***

მარტინ ვიერი – **Full Control**

ფაილი ***Proj3.txt***

მარტინ ვიერი – **Full Control,**
Customer Service – **Read**

ფაილების შექმნის შემდეგ შევამჩნევთ, რომ ზემოთ წაყენებული მოთხოვნები ავტომატურად სრულდება **Proj1.txt** ფაილისათვის მშობელი საქაღალდიდან მიღებული მემკვიდრეობის გამო. ოღონდ, ცხადია, ფაილისათვის თვისებების ფანჯრის **Security** ჩანართში მონიშნული უნდა იყოს შესაბამისი ალამი.

რაც შეეხება მეორე ფაილს, აღნიშნული ალმის დაყენების შემდეგ **Remove** ღილაკით უნდა ამოვაგდოთ მისთვის მემკვიდრეობით გადმოცემული ყველა ნებართვა და ამის მერე ვირჩევთ *მარტინ ვიერის* სააღრიცხვო ბარათს (**mwier@corp.com**) და უკვე ნაცნობი ხერხით გავცემთ მასზე **Full Control** ნებართვას.

ანალოგიური წესებით მოხდება ნებართვების განსაზღვრა მესამე ფაილისთვისაც.

შემდეგი ნაბიჯი იქნება ჩვენ მიერ გაცემული ნებართვის გამოყენება *მარტინ ვიერისა* და ზემოთ ჩამოთვლილი ჯგუფების წევრ მომხმარებლებისთვის. შევამოწმოთ, აგრეთვე, *მარტინ ვიერის* მიერ შექმნილი ფაილებიდან რომელ მათგანთან აქვს შეღწევის უფლება, ვთქვათ, *ბეტ მარტინს.* (მომხმარებლის სახელია **bmartins,** პაროლი – **fastcar1**).

ნებართვების შეცვლა და მფლობელობის გადაცემა

მომხმარებელი, რომელმაც ფაილი შექმნა, მისი მფლობელიც ხდება. ფაილის მფლობელს ავტომატურად ენიჭება უფლება, განსაზღვროს ამ ფაილთან შეღწევის სახეები სხვა მომხმარებლებისათვის. მას შეუძლია გადააბაროს კიდეც ფაილი ნებისმიერ მომხმარებელს. იშვიათად, მაგრამ ზოგჯერ მაინც საჭირო ხდება, ფაილთან მუშაობაზე ნებართვების შეცვლა მისი მფლობელის უკითხავადაც მოხდეს. მაგალითად, როცა ფაილის მფლობელი შვებულებაში გავიდა, ანდა, სულაც, თავი დაანება

კომპანიაში მუშაობას, ხოლო ფაილში კი კომპანიისთვის რაიმე მნიშვნელოვანი ინფორმაცია ინახება. ასეთ შემთხვევაში პრობლემის გადასაწყვეტად ადმინისტრატორებისათვის დაშვებულია გამონაკლისი – მათ შეუძლიათ, თვითონ გადმოიბარონ ფაილი *(გახდნენ მისი მფლობელი),* ანდა ნებისმიერ მომხმარებელს მიანიჭონ უფლება, სურვილის შემთხვევაში ჩაიბაროს ფაილი. შევნიშნოთ, რომ ადმინისტრატორს უშუალოდ არ შეუძლია, მომხმარებელი აქციოს ფაილის მფლობელად.

ამჯერად, გადმოვიბაროთ **Proj2.txt** ფაილი, როგორც ადმინისტრატორმა, და შემდეგ ნება დავრთოთ **Customer Service** ჯგუფის წევრებს, შეეძლოთ მისი წაკითხვა:

1. სისტემაში შევიდეთ ადმინისტრატორის პაროლით და **Proj2.txt** ფაილის თვისებების ფანჯარაში გადავიდეთ **Security** ჩანართზე.
2. დავაწკაპუნოთ **Advanced** ღილაკზე.
3. გადავდივართ **Owner** *(მფლობელი)* ჩანართზე.
4. ვაწკაპუნებთ **Administrators** ჯგუფზე. **OK.**
5. ვბრუნდებით წინა დიალოგურ ფანჯარაში. **OK.**

ახლა, ჩვენ, როგორც ფაილის მფლობელს, შეგვიძლია მივაღწიოთ დასახულ მიზანს – უკვე არაერთხელ გამოყენებული გზით გავცეთ შესაბამისი ნებართვა *(მოცემულ შემთხვევაში* **Customer Service** *ჯგუფის წევრებს მიეცეთ* **Proj2.txt** *ფაილის წაკითხვის უფლება),* რის შედეგადაც უკვე ბეტ მარტინსიც შეძლებს ფაილის წაკითხვას. შევამოწმოთ.

NTFS ტომებზე ფაილების და საქაღალდეების შეკუმშვა

NTFS სისტემა ფაილების *(საქაღალდეების)* შესაკუმშად მეტად მოხერხებულ, დამატებით საშუალებებს გვთავაზობს. საინტერესოა, რომ შეკუმშულია თუ არა ფაილი, ეკრანზე არც კი ჩანს, თუ სპეციალური ზომები არ მივიღეთ *(მისი სხვაგვარ ფერად მონიშვნის გზით).*

ფაილი გამოძახებისას მომხმარებელს შეუკუმშავი სახით მიეწოდება, შეტანილი ცვლილებების დამახსოვრებისას კი ავტომატურად ხდება მისი ხელახლა შეკუმშვა.

შეკუმშვის შედეგად მოგებას ვნახულობთ არა მარტო მეხსიერების გამოთავისუფლების მხრივ. ნაკლები ზომის ფაილის წაკითხვას დროც ნაკლები სჭირდება. მაგრამ თუ ფაილებში

ცვლილებების შეტანა ხშირად ხდება, მაშინ განმეორებით შეკუმშვებზე, ბუნებრივია, პროცესორს დროც მეტი დაეხარჯება.

ასე რომ, ამჯობინებენ, შეკუმშონ ისეთი ფაილები, რომლებიც იშვიათად იცვლება.

ფაილის *(საქაღალდის)* შესაკუმშად მივმართავთ თვისებების ფანჯარაში **Advanced** ჩანართს და ვაყენებთ მასში ალამს:

Compress Contents to Save Disk Space.

დაბოლოს, როგორც აღვნიშნეთ, შეკუმშული ფაილების *(საქაღალდეების)* სახელები ფერითაც *(ლურჯით)* შეიძლება გამოვარჩიოთ:

1. **Control Panel** საქაღალდეში ორჯერ ვაწკაპუნებთ **Folder Options** აპლეტზე.
2. გადავდივართ **View** ჩანართზე.
3. ვაყენებთ ალამს:

Display Compressed Files and Folders with Alternate Color.

შეკუმშულ ფაილს *(საქაღალდეს)* ენიშნება **Compressed** ატრიბუტი. შეკუმშული ელემენტის პიქტოგრამაზე დაწკაპუნებისას ამაში დავრწმუნდებით.

ფაილების და საქაღალდეების კოპირება და გადაადგილება

NTFS ფაილურ სისტემაში შექმნილი ფაილების *(საქაღალდეების)* კოპირებისას *(გადატანისას)* **FAT** განყოფილებებში ყველა ნებართვა იკარგება. იკარგება შეკუმშულობაც.

რაც შეეხება სიტუაციას **NTFS → NTFS,** აქ ცალ-ცალკე უნდა განვიხილოთ კოპირების და გადატანის შემთხვევები.

NTFS ფაილების *(საქაღალდეების)* კოპირებისას ნებისმიერ ადგილას *(თუნდაც სხვა კომპიუტერზე)* მყოფ **NTFS** საქაღალდეში, ნებართვების სახე განისაზღვრება ადგილზე მემკვიდრეობით. ასევე, ახალ ადგილზე მშობელი საქაღალდის ატრიბუტები განსაზღვრავს კოპირებული საქაღალდის ატრიბუტებს, შეკუმშვა-შეუკუმშველობას.

NTFS ფაილების *(საქაღალდეების)* გადატანისას **NTFS** საქაღალდეში:

- მემკვიდრეობითობა შენარჩუნდება, თუ გადატანა ხდება მოცემული ტომის ფარგლებში.
- სხვა ტომის სივრცეში ფაილების *(საქაღალდეების)* გადატანა განიხილება, როგორც კოპირება + საწყისი ობიექტის ამოგდების პროცესი, რის გამოც ფაილები *(საქაღალდეები)* კარგავენ მათზე გაცემულ ყველა ნებართვას, ატრიბუტებს და მათ ადგილზე მემკვიდრეობით განესაზღვრება *(ახალი)* მშობლისაგან შესაბამისი მახასიათებლები.

მომხმარებლებისათვის დისკოზე გამოყოფილი მეხსიერების კვოტირება

Windows 2000-ში გათვალისწინებულია მომხმარებლებისათვის დისკოზე გამოყოფილი მეხსიერების შეზღუდვის *(კვოტირების)* მექანიზმიც. აღვნიშნოთ, რომ იგი მუშაობს მხოლოდ ტომის დონეზე *(არსებობს სხვა კომპანიების მიერ დამუშავებული პროგრამული საშუალებები, რომლებიც კვოტებს განსაზღვრავენ საქაღალდის, სერვერის, დომენის და სხვა დონეებზეც).*

ამასთან, როცა ხდება კვოტის საზღვართან წინასწარ დადგენილი "მანძილით" მიახლოვება, სისტემა გვაფრთხილებს მომხდარის შესახებ.

არსებობს კვოტების ორი სახე: ძირითადი და დამატებითი. დამატებითი კვოტა შეიძლება ძირითადზე ნაკლები ან მეტიც იყოს. მექანიზმი ასეთია: ადმინისტრატორს შეუძლია სტანდარტული *(ძირითადი)* კვოტა განუსაზღვროს ყველა მომხმარებელს, შემდეგ კი ზოგიერთ მათგანს იგი გაუზარდოს ან, პირიქით, შეუმციროს.

მოვიყვანოთ მაგალითი:

1. ადმინისტრატორის მიერ **D** დისკოსათვის გამოძახებულ კონტექსტურ მენიუში გადავიდეთ **Properties → Quota** ჩანართზე.
2. ვაყენებთ ალმებს:

   Fnable quota management და

   Deny disk space to users exceeding quota limit.

3. ძირითადი კვოტის ზომის განსაზღვრისათვის ვაყენებთ გადამრთველს **Limit disk space to** მდგომარეობაში და ვირჩევთ საჭირო ზომას *(მაგალითად, 2 მეგაბაიტს).*

4. გაფრთხილების დონისათვის, ცხადია, ნაკლები ზომა უნდა ავირჩიოთ. **Set Warning Level to** პარამეტრისთვის დავაყენოთ 1 მეგაბაიტი მნიშვნელობა.
5. ახლა უკვე შეგვიძლია სხვადასხვა კვოტები დავუნიშნოთ ცალკეულ მომხმარებლებს. ამ მიზნით, დავაწკაპუნებთ **Quota Entries** ღილაკზე და გამოსულ ფანჯარაში გავცემთ ბრძანებას:

   New Quota Entry *(ახალი კვოტის განსაზღვრა).*
6. ეკრანზე აისახება **Select Users** დიალოგური ფანჯარა. ავირჩიოთ მასში *მარტინ ვიერის* სააღრიცხვო ჩანაწერი. **OK.**
7. შევცვალოთ, ჩვენი შეხედულებისამებრ, მომხმარებლისათვის გათვალისწინებული კვოტირების პარამეტრები. **OK.**

საერთო რესურსები და განაწილებული ფაილური სისტემა Dfs (Distributed File System)

ჯერ განვიხილოთ საერთო რესურსების საკითხი. ჩვენ მასთან უკვე გვქონდა შეხება, როცა პრინტერი გადავეცით საერთო მფლობელობაში.

აღსანიშნავია, რომ **NTFS** ფაილური სისტემის შექმნამდე საერთო რესურსების მექანიზმში გამოიყენებოდა ქსელში უსაფრთხოების უზრუნველსაყოფად.

საერთო რესურსებთან შეღწევის უფლება მომხმარებელს *(ჯგუფს)* ან გადაეცემა, ან არა. თუ რესურსის შემადგენლობაში ცალკეული ფაილებია, მათ დონეზე საკითხის გადაწყვეტა არ ხდება. ცალკეული ფაილების დაცვა შეიძლება განხორციელდეს მხოლოდ საერთო რესურსებისა და **NTFS** ნებართვების კომბინირების შედეგად.

საერთო რესურსებისათვის გაცილებით ნაკლები რიცხვის ნებართვებია გათვალისწინებული:

- **Full Control.** მომხმარებლებს შეუძლიათ ყველა სახის ოპერაციის შესრულება, მათ შორის მფლობელობის უფლების გადაცემაც.
- **Change** *(ცვლილებები).* დასაშვებია ფაილების *(საქაღალდეების)* შექმნა, ამოგდება, შეცვლა.

- **Read** *(კითხვა).* მომხმარებლებს შეუძლიათ საერთო რესურსების შემცველობის ჩათვალიერება და პროგრამების გაშვება.

ისევე, როგორც **NTFS**-ის ნებართვების შემთხვევაში, კონკრეტული მომხმარებლებისათვის აქაც ადგილი აქვს ცალკეულ ჯგუფებში მათი წევრობის შედეგად მიღებული ნებართვების შეჯამებას.

შესაძლებელია ცალკეულ მომხმარებლებს **Deny** *(აკრძალვა)*-ის მეშვეობით საერთო რესურსებთან შეღწევა აეკრძალოს. ამ შემთხვევაში მნიშვნელობა აღარ აქვს, შედის თუ არა იგი ისეთი ჯგუფის წევრების რიცხვში, რომელთაც უფლება აქვთ, აღნიშნული რესურსებით ისარგებლონ.

ამრიგად, **Deny**-ის აქვს "ვეტოს უფლება".

ვნახოთ, როგორაა საქმე დანარჩენ შემთხვევებში, როდესაც მომხმარებლები ცდილობენ **NTFS** ტომზე გამოყოფილ საერთო რესურსებთან შეღწევას. აქ ნებადართულია ის, რის უფლებასაც ორივე "მხარე" იძლევა, ანუ გამოიყენება ნებართვების გადაკვეთის საერთო უბნის ელემენტები.

საერთო რესურსების შექმნა

დომენის კონტროლერზე საერთო რესურსების შექმნის უფლება აქვს ადმინისტრატორებს და სერვერის ოპერატორებს. სხვა კომპიუტერებზე ეს უფლება ეძლევა გამოცდილ მომხმარებლებსაც *(ანუ* **Power Users** *ჯგუფის წევრებს).* აქვე შევნიშნოთ, რომ თუ საერთო რესურსი **NTFS** ტომზე იმყოფება, მომხმარებელს *(ოპერატორს)* უნდა ჰქონდეს **NTFS**-ის დონეზე მისი წაკითხვის უფლება.

დუმილით, როცა საერთო რესურსი იქმნება *(მაგალითად, როცა საქაღალდეს ვაქცევთ საერთო რესურსად)* ნებისმიერ მომხმარებელს ეძლევა მასთან შეღწევის უფლება.

ვაქციოთ **D** დისკოზე **A1** საქაღალდე საერთო რესურსად:

1. შევიდეთ სისტემაში როგორც ადმინისტრატორი.
2. **D** დისკოზე მყოფ **A**1 საქაღალდისთვის გამოვიძახოთ კონტექსტური მენიუ და გავცეთ **Sharing** ბრძანება.
3. დავაწკაპუნოთ ღილაკზე **Share This Folder** *(ვაქციოთ საერთო რესურსად).*

4. ნუ შევცვლით საერთო რესურსისთვის სისტემის მიერ შემოთავაზებულ სახელს და რესურსთან შეღწევის უფლების მქონე მომხმარებელთა რიცხვს.
5. დავაწკაპუნოთ **Permission** ღილაკზე და ნაცნობი გზით ავუკრძალოთ რესურსთან შეღწევა **Everyone** ჯგუფის წევრებს და, პირიქით, მივცეთ ამის უფლება **Customer Service** და **EXECS** ჯგუფებს *(**Full Control**-ზე შესაბამისი ალმის დაყენებით).*

საერთო რესურსებთან შეღწევა შეიძლება განხორციელდეს მათდამი მიმართვისას **UNC** (**Universal Naming Convention**) სახელების გამოყენებით.

UNC სახელის სტრუქტურა შემდეგი სახისაა:

\\სერვერის-სახელი\საერთო-რესურსის-სახელი\კატალოგის-სახელი\ქვეკატალოგის-სახელი

განაწილებული ფაილური სისტემა Dfs

ეს ფაილური სისტემა პირველად **Windows 2000** ოპერაციულ სისტემაში იქნა გამოყენებული. იგი იძლევა საერთო რესურსების ერთ დიდ სისტემაში მოქცევის საშუალებას. ფაილური სისტემის უმაღლესი წერტილის ანუ ფესვის სახელწოდებაა **Dfs.** თითოეულ სერვერზე მხოლოდ ერთი ფესვი შეიძლება არსებობდეს.

Dfs ფაილური სისტემის რეალიზება ორი სახით არის შესაძლებელი:

ცალკე მდგომი და მტყუნებებისადმი მდგრადი კონფიგურაციებით.

ცალკე მდგომი კონფიგურაცია

ამ მარტივი რეალიზაციისას საერთო რესურსებს განაგებს მხოლოდ ერთი სერვერი. ფაილური სისტემის შესახებ ინფორმაცია ქსელის ყოველ სერვერზე გადაიცემა ფონურ რეჟიმში, ჩვენი ჩარევის გარეშე.

Dfs განცალკევებული ფესვის შესაქმნელად:

1. სისტემაში შევდივართ როგორც ადმინისტრატორი.
2. გავცემთ ბრძანებას:

Start → Programs → Administrative Tools

3. ვირჩევთ ბრძანებას: **Distributed File System.**

   მივაქციოთ ყურადღება გაფრთხილებას იმის შესახებ, რომ **Dfs** ფესვი არ არსებობს. **OK.**

4. გაიშვება **Distributed File System** აღჭურვილობა. ახალი ფესვის შესაქმნელად გავცემთ ბრძანებას:

   Action → New Dfs Root

5. ეკრანზე აისახება ფანჯარა **Select Dfs Root Type** *(ფესვის ტიპის არჩევა).* არ ვცვლით დუმილით შემოთავაზებულ პარამეტრებს. **Next.**

6. მომდევნო ფანჯარაში (**Specify Server to Host Dfs**) ფესვისათვის სისტემა გვთავაზობს იმ სერვერს, რომელზეც ვმუშაობთ. აქაც ვაცხადებთ თანხმობას. **Next.**

7. შემდგომ ფანჯარაში (**Select Share for Dfs Root**) ვირჩევთ ლოგიკურ დისკოს – ლოკალურ რესურსს, რომლის ბაზაზეც შეიქმნება ფესვი. ტექსტურ ველში **Path to Share** *(გზა საერთო რესურსამდე)* ავკრიბოთ: **d:**, ხოლო საერთო რესურსის სახელად *(**Share Name** ტექსტურ ველში)* შევიტანოთ **Saroot.**

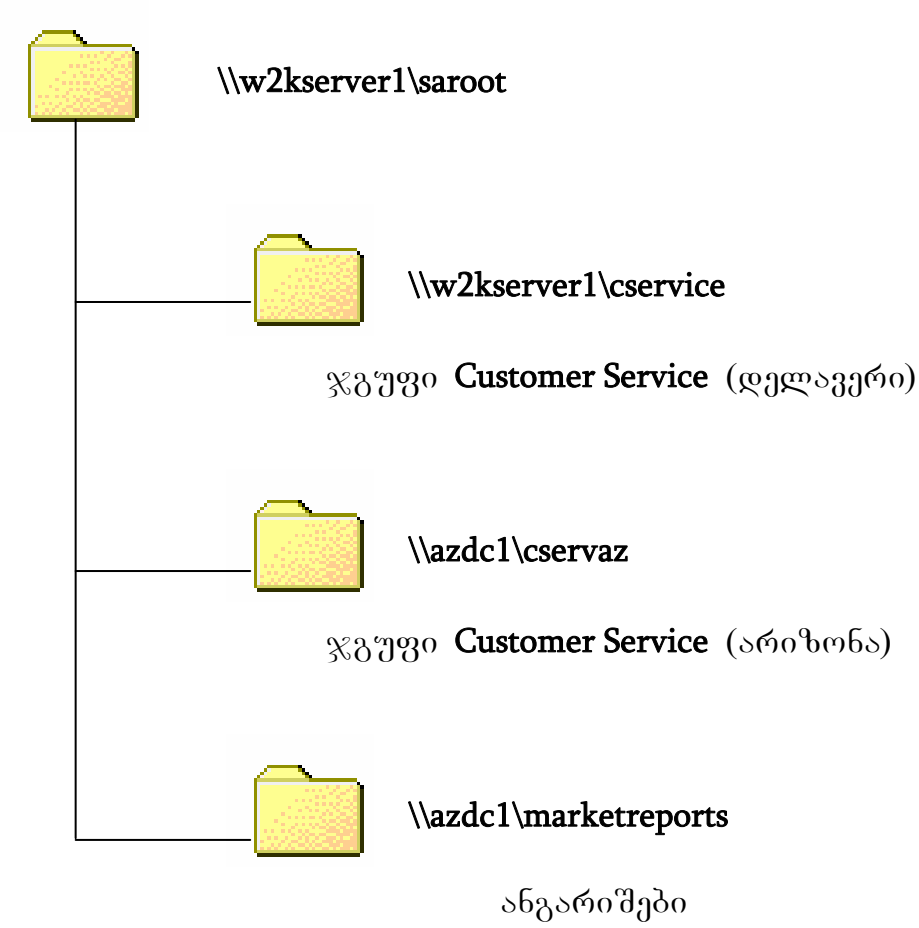
   ამრიგად, რესურსის სრული მისამართი იქნება:

   \\ W2kserver1.corp.com\saroot

8. **Next**-ზე დაწკაპუნების შემდეგ ეკრანზე გამოვა დიალოგური ფანჯარა: **Provide the Dfs Root Name** *(**Dfs** ფესვის სახელის ჩვენება).* მასში მხოლოდ ის მიეთითება, ვაპირებთ თუ არა ფესვთან დაკავშირებას ადმინისტრატორის კონსოლიდან. აქაც არაფერს ვცვლით. **Next.**

9. ბოლო ფანჯარაში აისახება ჩვენ მიერ არჩეული პარამეტრები. ვხედავთ, რომ დომენის და ფესვის სახელები ჯერჯერობით მიჩნეულია, როგორც გამოუყენებელი (**Not Applicable**). *(სახელების გამოყენების შესახებ ქვემოთ).* **Finish,OK.**

შექმნის შემდეგ ფესვს შეიძლება დაემატოს შვილობილი კვანძები. ვთქვათ, გვსურს განაწილებულ ფაილურ სისტემას შემდეგი სახე ჰქონდეს:



*ნახ. **Dfs** განცალკევებული ფესვის გრაფიკული წარმოდგენა*

1. სისტემაში შევდივართ, როგორც ადმინისტრატორი.
2. გავცემთ ბრძანებას:
   Start → Programs → Administrative Tools →
   Distributed File System.
3. ავირჩიოთ ფესვი **\\W2kserver1.corp.com\saroot**
4. **Dfs** შვილობილი კვანძის შესაქმნელად გავცემთ ბრძანებას:
   Action → New Dfs Child Node
5. ეკრანზე გამოვა დიალოგური ფანჯარა **Add to Dfs.**
   Child Node Type *(შვილობილი კვანძი)* ტექსტურ ველში შევიტანოთ **Customer Service – Delavere,** ხოლო **Send the user to this shared folder** *(გადავიდეთ ამ საერთო რესურსზე)* ტექსტურ ველში კი **\\W2kserver1\cservice**

კეშირების პარამეტრის მნიშვნელობას (1800 წმ) არ ვცვლით.

7. ზემოთ მოყვანილ პუნქტებს გავიმეორებთ სხვა შვილობილი კვანძებისათვისაც. შედეგად, როდესაც კლიენტები მიუერთდებიან **\\W2kserver1\saroot** საერთო რესურსს, მათ მთელს განაწილებულ სისტემასთან შეეძლებათ მუშაობა.

მტყუნებებისადმი მდგრადი Dfs ფესვის შექმნა

წინა შემთხვევაში საერთო რესურსებთან შეღწევა ხდებოდა ერთადერთი სერვერიდან *(თუმცა ეს რესურსები შეიძლება სხვა სერვერებზეც ყოფილიყვნენ განლაგებული),* ქსელში ჩართულ ნებისმიერ კომპიუტერზე კი აისახებოდა განაწილებული **Dfs** ფაილური სისტემის სტრუქტურა.

მაგრამ არსებობს სხვა გზაც. იგი გულისხმობს **Dfs** ფაილური სისტემის სრულ კოპირებას რამდენიმე კონტროლერზე *(თითოეულ ასლს ეწოდება <u>რეპლიკა</u>).* მისი გამოყენების შემთხვევაში ერთ-ერთი სერვერის მწყობრიდან გამოსვლა არ იწვევს მთელი ქსელის მუშაობის შეფერხებას. ამასთან, შესაძლებელია ქსელის მოცემული წერტილიდან სასურველ რესურსამდე უფრო მოკლე გზით სარგებლობაც.

დადებით მხარეებთან ერთად ზემოთ აღნიშნულ მიდგომას აქვს ორი მნიშვნელოვანი ნაკლიც:

- რესურს-ფაილში თუნდაც ერთი ბაიტის შეცვლისას საჭირო ხდება მთელი ფაილის გადაწერა *(ამ პროცესს რეპლიკაცია ეწოდება).* ცხადია, ამის გამო სისტემის მუშაობის სისწრაფე მცირდება.
- თუ ორი მომხმარებელი სხვადასხვა რეპლიკებში ერთდროულად ამუშავებს ორ ფაილს, "იმარჯვებს" ის მომხმარებელი, რომელიც უკანასკნელი იმახსოვრებს ცვლილებებს.

მტყუნებებისადმი მდგრადი **Dfs** ფესვის შექმნის პროცესი თითქმის ანალოგიურია ცალკეული **Dfs** ფესვის შექმნის. განსხვავება თავს იჩენს, ცხადია, მხოლოდ **Dfs** ფაილური სისტემის ვარიანტის არჩევაში.

მას შემდეგ, რაც მტყუნებებისადმი მდგრად **Dfs** ფესვს შევქმნით, რეპლიკაციების განსახორციელებლად სქემას შეიძლება დავუმატოთ სხვა სერვერებიც:

1. გამოვყოფთ **Dfs** ფესვს და გავცემთ ბრძანებას:

   Action → New Root Replica Member

   (შეიქმნას რეპლიკაციის ახალი წევრი)

2. ეკრანზე გამოსულ დიალოგურ ფანჯარაში **Specify Server to Host Dfs** (*სერვერის ჩვენება **Dfs**-ის განლაგებისათვის)* დუმილით შემოთავაზებულია ის სერვერი, რომელზეც ვმუშაობთ. ამჯერად, ავირჩევთ დომენის წევრ სხვა სერვერს. **Next.**

3. ვიმეორებთ **Dfs** ფესვის შექმნისთვის საჭირო ზემოთ მოყვანილ პუნქტებს.

<u>მთელი ეს პროცესი განმეორდება რეპლიკაციის სქემაში ყოველი ახალი სერვერის ჩართვისას.</u>

დასასრულ, მივუთითებთ, რომელი სერვერიდან უნდა დაიწყოს რეპლიკაციის პროცესი:

1. გამოვყოფთ **Dfs** ფესვს. გავცემთ ბრძანებას:

   Action → Replication Topology

2. მომდევნო დიალოგურ ფანჯარაში ვირჩევთ შემდეგ **Dfs** ფესვს: <u>\\W2kserver1.corp.com\ftroot</u>. ვაწკაპუნებთ ღილაკზე **Set Primary** *(განისაზღვროს, როგორც ძირითადი).* შევნიშნოთ, რომ **ftroot** კვანძის სახელია, რომელიც უკვე შექმნილი უნდა გვქონდეს.
3. რეპლიკაციის პროცესის აქტივიზებისათვის კვლავ ამოვირჩევთ <u>\\W2kserver1.corp.com\ftroot</u> ფესვს და ვაწკაპუნებთ **Enable** *(გააქტიურება)* ღილაკზე.
4. ვხურავთ ფანჯარას **OK**-ზე დაწკაპუნებით.

<u>კლიენტების მიერთება Dfs საერთო რესურსებთან და კავშირის წერტილებთან</u>

არსებობს მიერთების 4 ხერხი:

- ვაწკაპუნებთ სამუშაო მაგიდაზე განლაგებულ **My Network Places** *(ჩემი ქსელური რესურსები)* პიქტოგრამაზე. **Map Network** ბრძანებით ხდება რესურსისათვის ლოგიკური დისკოს შერჩევა და მისთვის **UNC** სახელის მინიჭება.

- გავცემთ **Run** ბრძანებას. საბრძანებო სტრიქონში ვკრებთ, მაგალითად, შემდეგი სახის ინფორმაციას:

  net use s:\\W2kserver1\cservice

- **Windows Explorer**-ის **Address** ველში შეგვაქვს **UNC** სახელი.
- **My Network Places** საქაღალდეში ჩავათვალიერებთ კომპიუტერების სიას, ვირჩევთ კომპიუტერს და ვხსნით სასურველ საქაღალდეს, რომელიც წარმოადგენს საერთო რესურსს ან **Dfs** ფესვს.

  რა თქმა უნდა, საქაღალდეში შესვლას მხოლოდ იმ შემთხვევაში შევძლებთ, თუ გვაქვს ამ საერთო რესურსთან შეღწევისა და **NTFS** ნებართვები.

ოქმები. ქსელის ფუნქციონირების საფუძვლები

ცნობილია, რომ ქსელის მუშაობისა და მისი მართვის პროცესები 7-დონიანი სქემის ჩარჩოებში განიხილება. თითოეულ დონეზე ორი, კავშირში მყოფი კომპიუტერის ურთიერთობას წარმართავენ ამ დონის შესაბამისი **პროტოკოლები** (ოქმები) – ქცევის წესები. ისინი უზრუნველყოფენ კომპიუტერებს შორის "საერთო ენის" გამონახვას, თუ რომელი მათგანი როდის როგორ უნდა მოიქცეს.

ამჯერად, ჩვენ გვაინტერესებს ქსელური და სატრანსპორტო დონეების შესატყვისი ოქმები. ზოგჯერ ორივეს ერთად ინტერნეტის დონესაც უწოდებენ. დღეისათვის ამ უბანზე გაბატონებული მდგომარეობა მოიპოვა **TCP/IP** პროტოკოლთა წყვილმა. მათ დაწვრილებით შევისწავლით. მოკლედ განვიხილავთ ამ მიზნით გამოყენებულ ზოგიერთ სხვა ოქმსაც. კერძოდ, ესენი გახლავთ **NetBEUI, IPX/SPX,NWLink** და **DLC** პროტოკოლები.

NetBEUI პროტოკოლი

ეს ოქმი მცირე ზომის ქსელებში გამოიყენება. იგი ერთმანეთთან აკავშირებს სამუშაო ჯგუფში შემავალ კომპიუტერებს. **NetBEUI** პროტოკოლი წარმოადგენს **NetBIOS** პროტოკოლის გაფართოებას. **NetBIOS** კი ახდენს კომპიუტერის იდენტიფიკაციას ქსელში ერთნაწილიანი დასახელებით. ცხადია, მხოლოდ **NetBIOS** სახელით სარგებლობა ინტერნეტსა და სხვა დიდ ქსელებში წარმოუდგენელია. ამის გამო, **NetBIOS** ერთ-ერთ კომპონენტად შედის სხვა, უფრო სრულყოფილ პროტოკოლებში, რომლებიც კომპიუტერის სახელდებას ახდენენ გლობალურ ქსელებში.

არსებობს სპეციალური სამსახური *(რომელიმე სერვერზე),* რომელიც ზრუნავს გლობალურ ქსელებში კომპიუტერების ცალსახა იდენტიფიცირებაზე. ეს გახლავთ **DNS (Domain Name System)** - დომენური სახელების ფორმირების სამსახური.

თუ სამუშაო ჯგუფში კომპიუტერის იდენტიფიცირებისათვის სავსებით საკმარისია მისთვის ერთნაწილიანი სახელის, ვთქვათ, **rci1**-ის მინიჭება, **DNS** სამსახური ამ სახელს დომენური სტრუქტურის სახელების ჯაჭვს, მაგალითად, **scribes.com**-ს მიუერთებს და **rci1.scribes.com** სრული სახელით მოახდენს კომპიუტერის ცალსახა იდენტიფიცირებას უფრო დიდი მასშტაბის ქსელში *(რეგიონალურში, გლობალურსა თუ ინტერნეტში).*

სახელების ამგვარი სტრუქტურა გვიადვილებს ლოკალურ ქსელში შემავალი კომპიუტერებისადმი მიმართვას – სავსებით საკმარისია გამოვიყენოთ სრული სახელის მხოლოდ პირველი, **NetBIOS** ნაწილი, მოცემულ შემთხვევაში - **rci1.**

რადგანაც **NetBIOS** და **NetBEUI** პროტოკოლების მარშრუტიზაცია არ ხდება, ანუ ინფორმაცია ქსელის ერთი სეგმენტიდან მეორეში ვერ გადაიგზავნება, საჭირო გახდა ახალი ოქმების შემუშავება. სწორედ, მათ ეყრდნობა ზემოთ განხილული **DNS** სამსახური. ერთ-ერთი მათგანია **IPX/SPX** პროტოკოლი.

IPX/SPX და NWLink პროტოკოლები

მიუხედავად იმისა, რომ **IPX/SPX** პროტოკოლი გაცილებით რთულია, ვიდრე **NetBIOS,** კომპიუტერზე მისი დაყენებაც ადვილად ხორციელდება, რადგანაც პარამეტრების უმეტესობისთვის ვეთანხმებით დუმილით გათვალისწინებულ მნიშვნელობებს.

IPX/SPX პროტოკოლი დუმილით გამოიყენება **Novel NetWare** ქსელებში, **Windows 2000 Server** კი იყენებს მასთან შეთავსებად **NWLink** პროტოკოლს. იგი შესანიშნავად მუშაობს მცირე და საშუალო ზომის ქსელებში. მაგრამ, რადგანაც იშვიათად გვხვდება ისეთი ქსელი, რომელსაც კავშირი არა აქვს ინტერნეტთან, *(აქ კი უდავოდ **TCP/IP** ლიდერობს),* უპირატესობას მაინც ამ უკანასკნელს ანიჭებენ, მიუხედავად მისი დაყენების ერთგვარი სირთულისა *(იხ. ქვემოთ).*

DLC პროტოკოლი

Windows 2000 Server-ის ბაზაზე აგებულ ქსელებში ეს ოქმი ძირითადად გამოიყენება სერვერების დასაკავშირებლად **Hewlett Packard** ფირმის ქსელურ პრინტერებთან.

TCP/IP პროტოკოლებისაგან განსხვავებული სახის ოქმების დაყენება და პარამეტრების განსაზღვრა

დავუშვათ, გვსურს სისტემაში გამოვიყენოთ **NWLink** პროტოკოლი. ამ მიზნით:

1. **Control Panel** ფანჯარაში მივმართავთ **Local Area Connection** აპლეტს და შემდეგ ვაწკაპუნებთ **Properties** ღილაკზე.
2. მომდევნო დიალოგურ ფანჯარაში ვაწკაპუნებთ **Install** ღილაკზე, ვირჩევთ **Protocol** პუნქტს და ვაწკაპუნებთ **Add** ღილაკზე. ამოვირჩევთ სასურველ პროტოკოლს. **OK.**

3. დამატებული ოქმისთვის სისტემა განსაზღვრავს თვისებებს, საჭიროების შემთხვევაში შესაძლებელია კორექტივების შეტანაც.

   შევნიშნოთ, რომ თუ კომპიუტერზე ორი ქსელური ფირფიტაა დაყენებული, დაგვჭირდება უნიკალური შინაგანი ქსელური ნომრის გამოყენება. **OK, CLOSE.**

   სერვერის გადატვირთვა აუცილებელი არ არის.

TCP/IP დამისამართება, ქვექსელის ნიღბები

როგორც ვიცით, **IP** ოქმი ზრუნავს ქსელში დანომრილი პაკეტის გადაცემაზე, ხოლო **TCP** – გადასაცემი შეტყობინების პაკეტებად დაყოფასა და დანიშნულების ადგილზე მათი მიღების შემდეგ იმავე შეტყობინების აწყობაზე.

კომპიუტერებს შორის ურთიერთობის დასამყარებლად საჭირო ხდება მანქანისათვის გასაგებ ენაზე მათი იდენტიფიცირება – დანომრვა. ამ მიზნით **TCP/IP** ოქმი იყენებს 4 ბაიტს. მაგალითად, **IP** მისამართს შეიძლება ასეთი სახე ჰქონდეს:

24. 0. 64. 18

თითოეული ბაიტი მნიშვნელობებს ღებულობს 0 – 255 დიაპაზონში, ანუ მისთვის შესაძლო მნიშვნელობათა რიცხვია 256.

კომპიუტერის 4-ბაიტიან მისამართი შეიძლება შემდეგი ორნაწილიანი სტრუქტურის სახით წარმოვადგინოთ:

საკუთრივ კომპიუტერის ნომერი *(ბოლო ნაწილი)* და მისი გარემომცველი ქსელის ნომერი *(პირველი ნაწილი).* დასაშვებია, კონკრეტულ მისამართებში ამ ნაწილებს თანრიგების განსხვავებული რაოდენობები დაეთმოს.

დაახლოებით 4 მილიარდი მისამართის მომცველი სიმრავლე **TCP/IP** ოქმის მიხედვით 3 კლასად წარმოდგება:

A კლასში ქსელების რიცხვია 126 *(შესაბამისად, მათი ნომრები 1 – 126 დიაპაზონში ვარირებს),* 127-ე ნომერი სპეციალური მიზნებისათვის გამოიყენება.

B კლასს ეთმობა დიაპაზონი 128 – 191.

C კლასისათვის დიაპაზონია 192 – 223.

(აქვე შევნიშნოთ, რომ არსებობს ზოგი სხვა კლასიც).

A კლასი განკუთვნილია უმსხვილესი ორგანიზაციებისათვის. ამ ტიპის 126 ქსელიდან თითოეულში შეიძლება განთავსდეს 256*256*256–2 = 16 777 214 კომპიუტერი. აღვნიშნავთ, რომ მისა-

მართების განაპირა მნიშვნელობები სპეციალურ მიზნებს ემსახურება.

B კლასი ეთმობა საშუალო მასშტაბის ორგანიზაციებს, კორპორაციებსა და უნივერსიტეტებს. რადგანაც მათი რიცხვი გაცილებით მეტია, მისამართების რაოდენობის გასადიდებლად იყენებენ მომდევნო – მეორე ბაიტსაც. მთლიანობაში **B** კლასის ქსელების რიცხვი 16 384-ით განისაზღვრება, თითოეულ მათგანში კი შეიძლება ფუნქციონირებდეს 65 534-მდე კომპიუტერი.

C კლასი ლოკალურ ქსელებზეა ორიენტირებული. კომპიუტერის იდენტიფიცირებისათვის საჭირო მეხსიერების კიდევ უფრო გაზრდით **C** კლასის ქსელების რიცხვი 532 676 608-ს აღწევს, თითოეულში 254 კომპიუტერის გაერთიანების შესაძლებლობით.

სამივე ტიპის ქსელები, ცხადია, ინტერნეტში მუშაობისთვისაცაა განკუთვნილი და ზემოთ აღწერილი წესით დამისამართება მსოფლიოს ნებისმიერ კუთხეში არსებულ კომპიუტერებს შორის "გაუგებრობებს" გამორიცხავს.

ისეთ შემთხვევაში კი, როცა დარწმუნებული ვართ, რომ ქსელს ინტერნეტთან კავშირი არ ესაჭიროება, შეიძლება მივმართოთ კომპიუტერების ე.წ. ~~კერძო დამისამართებასაც.~~

კერძო დამისამართება

თუ ლოკალურ კომპიუტერულ ქსელს ინტერნეტში გასვლა არ სჭირდება, დასაშვებია მასში გაცილებით მეტი კომპიუტერები ჩავრთოთ, ვიდრე ეს შესაძლებელია ხელმისაწვდომი პროვაიდერის მიერ გამოყოფილ სეგმენტში. ერთი შეხედვით, ასეთ შემთხვევაში თავისთავად უნდა მოიხსნას მსოფლიო მასშტაბით კომპიუტერის მისამართის უნიკალურობაზე ზრუნვის საკითხი. მაგრამ, რადგანაც მომავალში გამორიცხული არ არის ჩვენი ქსელის ინტერნეტთან მიერთება, ლოკალურ ქსელში არსებული კომპიუტერების იდენტიფიცირებისათვის იყენებენ კერძო დამისამართების ხერხს. მისი არსი შემდგომში მდგომარეობს:

კომპიუტერის მისამართის პირველი ნაწილი *(რომელიც, ფაქტობრივად, თვით ქსელის დამისამართებას ახდენს)* ამ შემთხვევაში წარმოადგენს თავისებურ, ამოქოლილ გასასვლელს, რისთვისაც გამოიყენება სპეციალური მისამართები:

A კლასისთვის 10.0.0.0 – 10.255.255.255 დიაპაზონში; ქვექსელის ნიღაბი 255.0.0.0 (იხ. ქვემოთ) საშუალებას იძლევა გამოიყოს 16 მლნ-ზე მეტი მისამართი.

B კლასს ეთმობა 172.16.0.0 – 172.31.255.255 დიაპაზონი. ქვექსელის ნიღაბი 255.255.0.0 ითვალისწინებს **B** კლასში შემავალი 16 ქსელიდან თითოეულს გამოეყოს დაახლოებით 65 ათასი მისამართი.

C კლასისათვის გამოიყენება 192.168.0.0 – 192.168.255.255 დიაპაზონი. ქვექსელის ნიღაბი 255.255.255.0 საშუალებას იძლევა **C** კლასის 256 ქსელიდან თითოეულში დამისამართდეს 254 კომპიუტერი.

ასეთი მისამართები კონფლიქტებს გამორიცხავს მაშინაც კი, როცა ლოკალური ქსელი ინტერნეტს მიუერთდება, რადგანაც კერძო მისამართი ინტერნეტით არ გადაიცემა.

სტანდარტული და მომხმარებლის მიერ დანიშნული ქვექსელის ნიღბები

ამრიგად, სტანდარტული ნიღბებია:

A კლასისთვის – 255.0.0.0;

B კლასისთვის – 255.255.0.0;

C კლასისთვის – 255.255.255.0.

ქვექსელის ნიღაბი ზღუდავს იმ კომპიუტერების რაოდენობას, რომლებისადმიც მოცემულ კომპიუტერს უშუალოდ მიმართვა შეუძლია. როდესაც ქსელში კომპიუტერს ვამატებთ, მისთვის უნდა შევირჩიოთ ისეთივე ნიღაბი, რომელიც გამოიყენება ამ ქსელში უკვე ჩართული კომპიუტერებისათვის.

სტანდარტული ნიღბების გარდა, შეიძლება გამოვიყენოთ მომხმარებლების მიერ შექმნილი ნიღბებიც.

ამ საკითხში უკეთ გასარკვევად რომელიმე ნიღაბი, მაგალითად, 255.255.255.0, ორობით სისტემაში წარმოვადგინოთ:

11111111.11111111.11111111.00000000

ბოლო ბაიტში, როგორც ვიცით, შეიძლება განვათავსოთ 254 მისამართი, მაგრამ თუ ჩვენ მიზანშეწონილად მიგვაჩნია 254 კომპიუტერის შემცველი ერთი ლოკალური ქსელი სეგმენტებად დავყოთ, შეგვიძლია ბოლო ბაიტში თანრიგების ნაწილი ქსელის მისამართს დავუთმოთ. მაგალითად, ავირჩიოთ შემდეგი ნიღაბი:

11111111.11111111.11111111.11110000

ანუ

255.255.255.240

შედეგად, ერთი ლოკალური ქსელი დაიყოფა $2^4 = 16$ სეგმენტად, რომელთაგან თითოეულში შესაძლებელი იქნება $2^4 - 2 = 14$ კომპიუტერის განთავსება.

შევნიშნოთ, რომ ზოგჯერ ნიღაბს სხვაგვარადაც წარმოადგენენ, კერძოდ, მის ორობით გამოსახულებაში არსებული ერთიანების რიცხვით. მოცემული შემთხვევისთვის ნიღბის წარმოდგენის კომპაქტურ ფორმას ექნებოდა რიცხვ 28-ის სახე.

TCP/IP კლიენტის პარამეტრები

Windows 2000 Server-ის ინსტალირებისას **TCP/IP** პროტოკოლი უკვე დავაყენეთ. ამჟამად განვიხილოთ მისი პარამეტრების განსაზღვრის საკითხი.

სამუშაო მაგიდაზე განლაგებულ **My Network Places** *(ჩემი ქსელური შეერთებები)* იარლიყისთვის გამოძახებულ კონტექსტურ მენიუში ავირჩიოთ **Properties** პუნქტი, ორჯერ დავაწკაპუნოთ **Local Area Connection** *(ლოკალური შეერთება)* ნიშნაკზე და შემდეგ ერთხელ **Properties**-ზე.

ავირჩიოთ **TCP/IP** ინტერნეტ-პროტოკოლი და დავაწკაპუნოთ **Properties**-ღილაკზე.

General ჩანართში **Obtain IP address automatically** *(IP-მისამართის ავტომატურად განსაზღვრა)* პოზიციიდან გადამრთველი შესაძლებელია გადავიყვანოთ **Use the following** *(IP-მისამართი თვითონვე უჩვენეთ)* პოზიციაში. ასეთ შემთხვევაში შესაბამის ტექსტურ ველებში უნდა შევიტანოთ ინფორმაცია **IP**-მისამართის, ქვექსელის ნიღბის და დუმილით დანიშნული რაბის შესახებ.

უნიკალური **TCP/IP** მისამართის განსაზღვრისათვის უნდა მივმართოთ ჩვენს ინტერნეტ-პროვაიდერს ანდა მოვინახულოთ **Web**-კვანძი

http://www.arin.net

რაბის როლში შეიძლება გამოვიდეს მარშრუტიზატორი (**router**) ან სხვა რომელიმე სერვერი. ასეთ შემთხვევაში მიეთითება მათი **IP**-მისამართი.

აღვნიშნოთ, რომ რაბი მაშინ გამოიყენება, როდესაც მიმართვა ხდება ისეთ კომპიუტერისადმი, რომლის მისამართი მოცემული ქვექსელისათვის გამოყოფილ **IP**-მისამართების დიაპაზონს სცილდება.

რაბს *(მარშრუტიზატორს)* ქსელის ერთი ნაწილიდან მეორეში პაკეტების გადასაგზავნად ესაჭიროება მინიმუმ ორი **IP**-მისამართის განსაზღვრა *(თითოსი თითოეული სეგმენტისათვის).*

ამრიგად, სხვა სეგმენტში გადასაგზავნი პაკეტები გაემართებიან დუმილით გათვალისწინებული რაბისაკენ, რომელიც, თავის მხრივ, მათ გადააგზავნის მომდევნო რაბზე და ეს პროცესი გაგრძელდება მანამ, სანამ პაკეტი დანიშნულების ადგილს არ მიაღწევს.

ამგვარივე წესით, ავტომატურად ან ჩვენ მიერ, განისაზღვრება **DNS**-სერვერის **IP**-მისამართიც *(იხ. ქვემოთ).*

ვაწკაპუნებთ **Cancel**-ღილაკზე, რათა ჯერჯერობით ყველაფერი უცვლელად დავტოვოთ.

TCP/IP პროტოკოლის გაწყობა საბრძანებო სტრიქონიდან

ზოგჯერ **TCP/IP** ოქმის მუშაობაში წარმოიშვება რაიმე პრობლემები. მათ გადასაწყვეტად მიმართავენ საბრძანებო სტრიქონის უტილიტებს. გავეცნოთ ზოგიერთ, ყველაზე მნიშვნელოვან ბრძანებას.

Ipconfig

ipconfig/all/more ბრძანება ინფორმაციას გვაწვდის ქსელური ადაპტერების შესახებ.

more პარამეტრი უზრუნველყოფს ეკრანზე მონაცემების ნაწილ-ნაწილ გამოტანას *(როცა ინფორმაცია მასზე მთლიანად ვერ ეტევა),* **all**-ით კი ვეცნობით ყველა ქვემოთ განხილული პარამეტრის მნიშვნელობას.

Ping

ამ ბრძანებით შეგვიძლია გავარკვიოთ, არსებობს თუ არა კონკრეტულ კომპიუტერთან კავშირი. სახელი მოიცემა რიცხვითი ან დომენური ფორმით. მაგალითად:

ping 207.55.56.4

ping boutell.com

მისამართების მითითება შეიძლება **Web**-ბროუზერიდანაც *(მისამართის სტრიქონში აკრებით):*

http://www.boutell.com

თუ "მუშაობს" ციფრული მისამართი, დომენური კი – არა, შეიძლება დავასკვნათ, რომ **DNS**-სამსახური არ ფუნქციონირებს.

ping ბრძანება შესაძლებელია საკუთარ კომპიუტერსაც გავუგზავნოთ, რისთვისაც ყოველთვის უნდა ავკრიბოთ შემდეგი მისამართი:

127.0.0.1

თუ პასუხი ვერ მივიღეთ, მიზეზი გახლავთ **TCP/IP** ოქმის დაზიანება ან ჩაუტვირთაობა. ასეთ შემთხვევაში ჯერ ამოვავაგდებთ არსებულ ოქმს და შემდეგ მას ხელახლა ჩავტვირთავთ *(ან პარამეტრებს შევუცვლით).*

გამოვიყენოთ **ping** ბრძანება დუმილით გათვალისწინებული რაბისათვის. პასუხის მიღებისას ვრწმუნდებით, რომ პაკეტები, სულ ცოტა, ქსელის ლოკალური სეგმენტის ფარგლებში შეიძლება გავრცელდეს.

Tracert

ამ ბრძანების სინტაქსიც იგივეა, რაც წინასი:

Tracert 207.55.56.4

Tracert boutell.com

მისი მეშვეობით შეიძლება გავიგოთ, რამდენი მარშრუტიზატორის გავლა უწევთ პაკეტებს დანიშნულების ადგილის მიღწევამდე, მაშასადამე, გამოვარკვიოთ, არის თუ არა ეს მარშრუტი ინფორმაციის დაბალი სიჩქარით გადაცემის მიზეზი.

NetStat

ეს ბრძანება გვაწვდის **TCP/IP** ოქმის ყველა პორტის სიას. აღვნიშნოთ, რომ პორტის მეშვეობით ხდება საჭირო გამოყენებასთან შეერთების დამყარება.

ყველა შეერთების სიის ჩათვალიერება ხდება **NetStat –a** ბრძანებით.

Nbstat

ვიცით, რომ **TCP/IP** ოქმი შეიცავს *(იყენებს)* **NetBIOS** პროტოკოლსაც. ამ უკანასკნელის პარამეტრების ნახვა, სწორედ, **Nbstat** ბრძანებით არის შესაძლებელი.

მას შემდეგ, რაც გაიშვება **TCP/IP** პროტოკოლი, სპეციალური პროგრამების მეშვეობით შეგვიძლია უზრუნველვყოთ სხვა კომპიუტერებთან შეერთებები. ერთ-ერთი ასეთი პროგრამა არის **FTP (File Transfer Protocol –** *ფაილების გადაცემის პროტოკოლი***).** იგი

გამოიყენება ინტერნეტში განთავსებულ **FTP-** სერვერებზე მყოფ ფაილებთან შეღწევისა და კოპირებისათვის.

DNS სამსახური

რით არის გამოწვეული **DNS** სამსახურის არსებობა?

სასურველი კომპიუტერის ინტერნეტში მოძებნა **IP**-მისამართის ჩვენებით ამოცანის არცთუ ისე მოხერხებული გადაწყვეტაა. ბუნებრივია, ადამიანს ურჩევნია კომპიუტერს *(საიტს, ვებ-ფურცელს)* სიტყვიერი სახელით მიმართოს.

ვიცით, რომ ლოკალურ ქსელებში კომპიუტერების სახელ-დებისათვის გამოიყენება ერთნაწილიანი **NetBIOS** სახელები. მათ შეთანადებაზე **TCP/IP** მისამართებთან ზრუნავს **WINS** სამსახური *(იხ. ქვემოთ).* გლობალურ ქსელებსა და ინტერნეტში კი ამ მიზნით გამოიყენება **DNS (Domain Name System) -** დომენური სახელების სისტემა.

დომენური სტრუქტურის თავში იმყოფება ფესვური დომენი, რომელიც აღინიშნება "." სიმბოლოთი. მას მოსდევს ზედა დონის დომენები:

.com, .org, .edu, .gov… ან ქვეყნების აღმნიშვნელი დასახე-ლებები:

.au (ავსტრალია), **.ge** (საქართველო), **.uk** (დიდი ბრიტანეთი), **.ru** (რუსეთი)… და ა.შ.

თუ დომენის დასახელებაში ქვეყნის კოდი არ გვხვდება, იგი აშშ-ში არის დარაგისტრირებული.

ზედა დონის დომენებს მოსდევს მეორე დონის დომენები და ა.შ.

ბოლო პოზიციაზე ფიგურირებს რესურსის *(კომპიუტერის, მასზე არსებული რომელიმე პროგრამის)* სახელი.

მთლიანად განსაზღვრული დომენის სახელი **FQDN (Fully Qualified Domain Name)** არის მიმდევრობა, რომელიც მთავრდება კომპიუტერის სახელით. სწორედ, **FQDN**–ზე დაყრდნობით, **DNS** სამსახურს ტექსტური დასახელება გადაჰყავს **TCP/IP** ციფრულ მისამართში.

DNS სამსახურის დაყენება

დომენის ფესვური სერვერის გარდაქმნისას **W2kserver1** სახელის მქონე დომენის კონტროლერად ჩვენ უკვე დავაყენეთ **DNS** სამსახური. მაგრამ მისი დაყენება შეიძლება დომენში ჩართული სხვა, მაგალითად, **AZDC1** სერვერისთვისაც. ამ მიზნით:

1. **Control Panel**-ში მივმართავთ **Add/Remove Programs** აპლეტს და გადავდივართ **Add/Remove Windows Components** ჩანართზე.
2. გამოსულ ფანჯარაში ვაწკაპუნებთ **Next**-ზე. ავირჩევთ **Networking Services** კომპონენტს და ვაწკაპუნებთ **Details** ღილაკზე.
3. ვაყენებთ **Domain Name Service (DNS)** ალამს, **OK, OK, Finish.**
4. ჩატვირთული **DNS**-სამსახურის გასაშვებად გავცემთ ბრძანებას:

Start → Programs → Administrative Tools

ვაწკაპუნებთ **DNS**-ზე.

DNS-სერვერის ტესტირებისთვის:

1. ვაწკაპუნებთ **AZDC1** სერვერის კონტექსტურ მენიუში **Properties** პუნქტზე.
2. გადავდივართ **Monitoring** ჩანართზე. ვაყენებთ ალმებს.:

Simple Query *(მარტივი მოთხოვნა)*

Recursive Query *(რეკურსიული მოთხოვნა)*

ვაწკაპუნებთ ღილაკზე **Test Now** *("აბა, ჰე!")*

თუ მარტივი მოთხოვნა ვერ შესრულდა, უნდა შევამოწმოთ, სწორად განვსაზღვრეთ თუ არა სერვერზე ქსელური ფირფიტის მისამართი. რეკურსიული მოთხოვნის შეუსრულებლობის შემთხვევაში *(ეს მოთხოვნა უფრო მაღალ დონეზე მყოფ **DNS**-სერვერზე გადაიგზავნება)* ვამოწმებთ ჩვენს სერვერზე **DNS**-სერვერების სიის სისწორეს *(ვთქვათ, ხომ არ მოძველდა იგი)* და **ping** ბრძანებით ვცდილობთ მათთან კავშირზე გასვლას.

DNS სამსახურის ფუნქციონირების პრინციპები

დავუშვათ, რომელიმე მომხმარებელს **widget.corp.com** დომენიდან სურს **www.boutell.com** კვანძზე არსებული რაიმე **Web**–ფურცლის ჩათვალიერება.

თავდაპირველად **widget.corp.com** სერვერის პროტოკოლი გამოარკვევს, რომელ **DNS**-სერვერს უნდა გაეგზავნოს მოთხოვნა. ბუნებრივია, ჯერ ეს იქნება ამავე დომენში შემავალი **DNS**-სერვერი, რომელიც შეამოწმებს, ხომ არ ყოფილა ბოლო ერთი საათის განმავლობაში *(ამ დროის შემდეგ ინფორმაცია იკარგება)* **widget** დომენიდან **boutell** კვანძისადმი მიმართვა. თუ ეს ასეა, სასურველი პუნქტის **IP**-მისამართი ადგილზევე გამოიძებნება. წინააღმდეგ შემთხვევაში მოთხოვნა გადაეგზავნება ზედა დომენის (აქ **corp.com** დომენის) **DNS**-სერვერს, რომელიც იმავე სტილში გააგრძელებს მოთხოვნაზე მუშაობას. ცხადია, ასეთი შესრულების გამო იწოდება ეს პროცესი რეკურსიულად.

მოთხოვნამ შესაძლოა "უმაღლეს ინსტანციასაც" მიაღწიოს – ეს გახლავთ ფესვური დომენის **DNS**-სერვერი. იგი კი მოთხოვნას **.com** დომენის იმ სერვერთან გადააგზავნის, რომლისთვისაც ცნობილია **boutell.com** დომენის ერთ-ერთი **DNS**-სერვერის მისამართი. ამ სერვერიდან უკუგზით საჭირო ინფორმაცია გაეგზავნება ძებნის ინიციატორ კომპიუტერს.

რა თქმა უნდა, **IP**-მისამართების განსაზღვრა შესაძლებელია პროვაიდერების **DNS**-სერვერებისათვის გვერდის ავლითაც – უშუალოდ ფესვური სერვერებისადმი მიმართვით, მაგრამ ამ შემთხვევაში ძებნის პროცესი ძალიან ჭიანურდება. ამის გამო, როგორც წესი, მიმართავენ ინტერნეტ-პროვაიდერების მომსახურებას: საკმაოდ მაღალია მათი სერვერების კეშ-ფაილებში ჩვენთვის საჭირო ინფორმაციის მოძიების ალბათობა.

ამ მიზნით:

1. გავცემთ ბრძანებას:

   Start → Programs → Administrative Tools

   ვაწკაპუნებთ **DNS**-ზე.
2. **AZDC1** სერვერისათვის გამოვიძახებთ **Properties** ფანჯარას.
3. გადავდივართ **Forwarders** *(მონაცემების გადაგზავნა)* ჩანართზე.
4. ვაწკაპუნებთ **Add**-ღილაკზე და შეგვაქვს ჩვენი ინტერნეტ-პროვაიდერის **DNS**-სერვერის **IP**-მისამართი.

5. **Operate as a slave server** ალამი ჩამოგდებულ მდგომარეობაში დავტოვოთ, რათა საჭირო **IP**-მისამართის მოძებნა ჯერ ჩვენი დომენის ძალებით ვცადოთ. წინააღმდეგ შემთხვევაში მოთხოვნა ავტომატურად გადაეგზავნება პროვაიდერის **DNS**-სერვერს. **OK.**

WINS სამსახური

Windows 2000-ის მართვის ქვეშ მომუშავე კომპიუტერებისათვის ერთმანეთისადმი სახელებით მიმართვას შესანიშნავად ართმევს თავს **DNS** სამსახური. უფრო დაბალი დონის კლიენტ-კომპიუტერებისათვის კი ამჯობინებენ **WINS (Windows Internet Name Service)** ინტერნეტ-სახელების სამსახურს **Windows**-სთვის.

WINS სამსახურის დანიშნულება გაცილებით ლოკალურია, ვიდრე **DNS** სამსახურის. იგი კომპიუტერის **NetBIOS** სახელს შეუთანადებს დომენში მის ნომერს და პირიქით. ამასთან, საჭიროების შემთხვევაში ახერხებს მარშრუტიზატორის გავლასაც, რაც მის გარეშე ლოკალურ ქსელში კომპიუტერებისადმი ე.წ. ფართომაუწყებლობრივი მიმართვისას არ ხდება.

შევნიშნოთ, რომ ფართომაუწყებლობრივი მიმართვისას მოცემულ სეგმენტში არსებულ ყველა კომპიუტერს ეგზავნება მიმართვა იმ ანგარიშით, რომ მას გამოეხმაურება მხოლოდ მოთხოვნილი სახელის კომპიუტერი. ამ პროცესს ტრანსლირებასაც უწოდებენ.

მარშრუტიზატორის მეორე მხარესაც, ცხადია, უნდა ფუნქციონირებდეს მეწყვილე **WINS** სერვერი. მტყუნებებისადმი მდგრადობის გაზრდის მიზნით, მოცემულ სეგმენტშიც ამჯობინებენ რამდენიმე ასეთი სერვერის ჩართვას.

WINS სამსახურის დაყენება

WINS სამსახურის დასაყენებლად ავირჩიოთ **AZDC1** სერვერი:

1. **Control Panel**-ში ორჯერ ვაწკაპუნებთ **Add/Remove Programs**-ზე და გადავდივართ **Add/Remove Windows Components** ჩანართზე.
2. მომდევნო ფანჯარაში ვაწკაპუნებთ **Nexts**-ზე. ვირჩევთ **Networking Services** კომპონენტს და ვაწკაპუნებთ **Details**-ღილაკზე.

3. ვაყენებთ ალამს **Windows Internet Name Service (WINS), OK, OK, Finish.**

ამის შემდეგ ქსელში ჩართული კლიენტისათვის *(რომელიც **Windows 98**–ით იმართება)* **WINS** სერვერზე ხელახლა განსაზღვრავთ პარამეტრებს.

გავუშვათ **WINS** სამსახურის კონსოლი **AZDC1** კონტროლერზე, რომელიც **WINS** სერვერის ფუნქციებსაც ასრულებს:

Start → Programs → Administrative Tools

ვაწკაპუნებთ **WINS**-ზე.

WINS სერვერის ტესტირებისთვის კლიენტ-კომპიუტერის **(Buster)** გადატვირთვის შემდეგ, დავაწკაპუნოთ **Active Registrations** *(აქტიური რეგისტრაციების)* საქაღალდეზე და გავცეთ ბრძანება **Action → View Records.** გამოდის დიალოგური ფანჯარა, რომელშიც ვაწკაპუნებთ ღილაკზე **Show records for the selected owner** *(ჩანაწერების ჩვენება ამორჩეული მფლობელისათვის).* **OK** და ეკრანზე აისახება შესაბამისი სია.

აქ მფლობელად იგულისხმება **WINS** სერვერი, რომლის მეშვეობითაც დარეგისტრირდა კლიენტი-კომპიუტერი.

აღვნიშნოთ, რომ **WINS** სერვერის მეშვეობით დარეგისტრირებული თითოეული კლიენტისა თუ დომენისათვის რამდენიმე პარამეტრია ხელმისაწვდომი. კლიენტისა და დომენის **WINS** სერვერთან ურთიერთობისათვის ეს პარამეტრები უნდა დარეგისტრირებულ იქნეს.

ამ მიზნით:

1. **Active Registrations** საქაღალდისათვის გამოძახებულ კონტექსტურ მენიუში **Wiew Records** ბრძანებით ეკრანზე გამოგვყავს შესაბამისი დიალოგური ფანჯარა.
2. გადავდივართ **Record Types** ჩანართზე.
3. სიაში გამოსაყვანი ჩანაწერებისათვის ვაყენებთ ალმებს.

WINS რეპლიკაციების პარტნიორები

WINS სერვერებს შორის საჭიროა მოხდეს მონაცემთა ბაზების გაცვლა – რეპლიკაცია. ეს პროცესი მეორდება დროის წინასწარ განსაზღვრული ინტერვალის შემდეგ და მასში მონაწილეობს რეპლიკაციის სქემაში გაწევრიანებული ყველა პარტნიორი.

მაგრამ პროცესის განხორციელებისათვის აუცილებელია რეპლიკაციის ბაზაში შესაბამისი ინფორმაციის შეტანა.

AZDC1 სერვერისათვის *(იგი **WINS** სერვერის ფუნქციებსაც ასრულებს)* ამოვირჩიოთ რეპლიკაციის პარტნიორი:

1. გავცემთ ბრძანებას:

   Start → Programs → Administrative Tools

   ვაწკაპუნებთ **WINS**-ზე.

2. **Replication Partners** საქაღალდის კონტექსტურ მენიუში გავცემთ ბრძანებას

   New → Replication Partner

3. ვუჩვენოთ პარტნიორი სერვერის სახელი (აქ **W2kserver1**). მასზე **WINS** სამსახური უკვე გაშვებული უნდა გვქონდეს.

DHCP სამსახური

DHCP *(**Dynamic Host Configuration Protocol** – ჰოსტის დინამიკური კონფიგურაციის პროტოკოლი)* სამსახური კომპიუტერ-კლიენტებს ავტომატურად უნიშნავს **TCP/IP** მისამართებს და გადასცემს მათ მონაცემებს *(მისამართებს)* დუმილით გათვალისწინებული რაბის, **WINS** და **DNS** სერვერების შესახებ. შედეგად, კლიენტ-კომპიუტერების გაწყობის პროცესი ფრიად მარტივდება.

DHCP სამსახურის დაყენება

W2kserver1 სერვერზე **DHCP** სამსახური უკვე ფუნქციონირებს. ამჯერად იგი **AZDC1** სერვერზე დავაყენოთ:

1. კვლავ გამოვიძახოთ ოპერაციული სისტემის კომპონენტების დაყენება/ამოგდების აპლეტში **Networking Services** კომპონენტი და დავაწკაპუნოთ **Details** ღილაკზე.
2. ამჯერად ვაყენებთ **DHCP**-ალამს, **OK, OK, Finish.**
3. **DHCP**-სერვერის დაყენების შემდეგ შეიძლება დაგვჭირდეს კომპიუტერის გადატვირთვა.
4. **DHCP**-სამსახურის გასაშვებად გავცემთ ბრძანებას:

   Start → Programs → Administrative Tools და ვაწკაპუნებთ **DHCP**-ზე.

DHCP სერვერის გაწყობა

DHCP-სერვერმა რომ ქსელში თავისი ფუნქციები შეასრულოს, იგი ქსელში უნდა დავარეგისტრიროთ, რისთვისაც:

1. **Administrative Tools** ფანჯარაში გავუშვებთ **DHCP**-სამსახურის კონსოლს **DHCP**-ზე დაწკაპუნებით.
2. კონსოლის მარცხენა პანელში მოვნიშნავთ **DHCP**-საქადალდეს და გავცემთ ბრძანებას: **Action → Browse Authorized Servers** *(ავტორიზებული სერვერების ჩათვალიერება).*
3. ვაწკაპუნებთ **Add**-ღილაკზე და **DHCP**-სერვერის სახელად შეგვაქვს **AZDC1, OK.**

   *(მანამდე ეს პროცედურა **W2kserver1** სერვერისთვის უნდა გვქონდეს ჩატარებული).*
4. პროცესის გასაგრძელებლად ვაწკაპუნებთ **Yes**-ღილაკზე *(შესაბამისი მოთხოვნის გამოსვლისას).*
5. ვხურავთ დიალოგურ ფანჯარას.

უბნების შექმნა

შემდეგი ნაბიჯია ახალი **DHCP**-სერვერისათვის მისამართებით მომსახურების უბნის შექმნა. განვსაზღვროთ მისამართების დიაპაზონი, მაგალითად, ამგვარად:

24.0.64.18 - 24.0.64.24

და ქვექსელის ნიღაბი - **255.255.0.0**

ქსელის მომხმარებლებს ავტომატურად გადაეგზავნებათ შემდეგი ინფორმაცია:

- რაბის მისამართი – **24.0.64.2**
- **DNS** ძირითადი სერვერის მისამართი – **24.0.64.35** *(**AZDC1** სერვერია)*
- **WINS** ძირითადი სერვერის მისამართი – **24.0.64.35** *(იგივე სერვერია)*
- **WINS** დამატებითი სერვერის მისამართი – **24.0.64.61** *(**W2kserver1** სერვერია)*

DHCP-სერვერისათვის უბნის შესაქმნელად:

1. **Administrative Tools** ფანჯარაში გავუშვებთ **DHCP**-სამსახურის კონსოლს **DHCP**-ზე დაწკაპუნებით.
2. გამოვყოფთ **azdc1.corp.com** სერვერის სახელს და გავცემთ ბრძანებას: **Action → New → Scope** *(შეიქმნეს ახალი უბანი).*
3. გამოდის ოსტატის დიალოგური ფანჯარა. **Next.**
4. შეგვაქვს უბნის სახელი. **Next.**
5. განვსაზღვრავთ მისამართების დიაპაზონის საზღვრებს. **Next.**
6. ვუჩვენებთ ქვექსელის ნიღაბს.
7. მომდევნო, **Add Exclussions** *(გამონაკლისების დამატება)* დიალოგურ ფანჯარაში მივუთითებთ იმ მისამართებზე, რომლებიც დაფიქსირებული დიაპაზონიდან უნდა გამო-ირიცხოს. **Next.**

   აღვნიშნოთ, რომ დიაპაზონიდან გამოვრიცხავთ იმ კომპიუტერების მისამართებს, რომელთა შეცვლა არ დაიშვება. ესენი შეიძლება იყოს როგორც სერვერები, ასევე ზოგიერთი მნიშვნელოვანი კლიენტ-კომპიუტერიც. მოცემულ შემთხვევაში გამონაკლისებს არ ვაკეთებთ.
8. **Lease Duration** *(არენდის ძალაში ყოფნის პერიოდი)* ფანჯარაში ამჯერად არ ვცვლით დუმილით დადგენილ 8 დღის ვადას. **Next.**

   შევნიშნოთ, რომ ამ დროის გასვლის შემდეგ კლიენტს მისამართი ჩამოერთმევა.
9. შემდგომ დიალოგურ ფანჯარაში გვთავაზობენ უბნისათვის პარამეტრების შერჩევას. **Yes, Next.**
10. პირველი პარამეტრია რაბის მისამართი. მას მნიშვნელობად განვუსაზღვროთ **24.0.64.1. Next.**
11. მომდევნო დიალოგურ ფანჯარაში დომენის სახელად შეგვაქვს **corp.com,** რის შემდეგაც ვუჩვენებთ **DNS** სერვერის მისამართს – **24.0.64.35.** ეს გახლავთ **AZDC1** სერვერის მისამართი. **Next.**
12. შეგვაქვს ძირითადი და დამხმარე სერვერების მისამართები:

    24.0.64.35 (AZDC1)

    24.0.64.61 (W2kserver1)

13. **Next**-ით გადავდივართ მომდევნო ფანჯარაში. აქ გვთხოვენ, გავააქტიუროთ ახალი უბანი. **Yes, Next, Finish.**

DHCP-სერვერის უბანი მზადაა გამოყენებისათვის.

საინტერესოა, რომ კლიენტი-კომპიუტერები მზადყოფნაშია, ავტომატურად განესაზღვროთ **IP**-მისამართი. მაგრამ თუ მათ უკვე მინიჭებული აქვთ მუდმივი მისამართი, ის დინამიკურით უნდა შევცვალოთ.

ვნახოთ, როგორ მუშაობს მისამართების არენდირების მექანიზმი:

1. კლიენტ-კომპიუტერზე გავუშვებთ **DHCP**-სამსახურის კონსოლს.
2. კლიენტის გადატვირთვის შემდეგ მივმართავთ **DHCP**-სერვერის უბნის ხეს. ორჯერ ვაწკაპუნებთ **Address Leases** *(მისამართების არენდა)* საქაღალდეზე, რათა ვნახოთ, რომელი მისამართი აქვს დანიშნული კლიენტს. მოცემულ შემთხვევაში ეს გახლავთ პულიდან *(დიაპაზონიდან)* პირველი მისამართი - **24.0.64.18.**
3. შეგვიძლია ასევე ვისარგებლოთ საბრძანებო სტრიქონით. კერძოდ, **Windows NT** ან **Windows 2000** კლიენტისათვის მასში შეგვყავს **cmd** ბრძანება, **OK** და გამოსულ ფანჯარაში ვკრებთ ბრძანებას **ipconfig/all,** ხოლო **Windows 98** კლიენტის საბრძანებო სტრიქონში პირდაპირ შეგვყავს ბრძანება – **winipcfg, OK.**
4. იღება დიალოგური ფანჯარა **IP Configuration.**
5. ვრწმუნდებით, რომ **IP**-მისამართები ერთმანეთს ემთხვევა. თუ გვსურს, კლიენტს ახალი მისამართი მივაკუთვნოთ:
   ა) ზემოთ მოყვანილი წესებით გამოყვანილ ფანჯრებში **Windows NT** და **Windows 2000** კლიენტებისათვის ვკრებთ ბრძანებას **ipconfig/release,** ხოლო **Windows 98** კლიენტებისათვის კი ჯერ ვაწკაპუნებთ საბრძანებო სტრიქონში **Release** და მერე **Renew** *(განახლება)* ღილაკებზე.
   ბ) კლიენტს დაენიშნება ახალი **TCP/IP** მისამართი. ძალიან ხშირად ეს მისამართი წინას იდენტურია – ჩანს, დროის მოკლე პერიოდში იგი ჩვენთვის სხვა კომპიუტერს არ წაურთმევია.

დასასრულ, აღვნიშნოთ, რომ **DHCP** პაკეტები შეიცავს არა მარტო კლიენტების **IP**-მისამართებს, არამედ **WINS, DHCP** და **DNS** სერვერების მისამართებსაც. ამრიგად, **DHCP** სამსახური მაკავშირებელი რგოლის ფუნქციას ასრულებს სხვადასხვა სამსახურებისათვის.

ქსელის წარმოებადობის დიაგნოსტიკა და კონტროლი

ქსელის და სერვერების წარმოებადობის დიაგნოსტიკა-კონტროლის ინსტრუმენტები შესაძლებლობას იძლევიან, გამოვავლინოთ ოპერაციული სისტემის მუშაობის პროცესში "ვიწრო" ადგილები, მოვახდინოთ მათი აღმოფხვრა, უფრო ოპტიმალური გავხადოთ სისტემის ფუნქციონირება და დავიცვათ მისი უსაფრთხოება.

ამ მიზნით, კონტროლდება აპარატურულ და პროგრამულ უზრუნველყოფებში ხდომილობათა დიდი სპექტრი, წარმოებს მათი რეგისტრაცია სპეციალურ ჟურნალებში, რის საფუძველზეც მიიღება შესაბამისი გადაწყვეტილებები მოსალოდნელ საფრთხეთა თავიდან ასაცილებლად და ქსელის წარმოებადობის ასამაღლებლად.

სერვერის ჩართვისას ავტომატურად ხდება **Event Log** სამსახურის გაშვება. უტილიტა **Event Log** *(ხდომილობების ჩათვალიერება)*-ის მეშვეობით ვეცნობით შემდეგი სამი, სისტემაში ჩაშენებული ჟურნალის შემცველობას:

- გამოყენებებში წარმოქმნილ ხდომილობათა რეგისტრაციის ჟურნალი – **Application Log**. საკითხს, თუ რომელი ხდომილობები უნდა დარეგისტრირდეს მასში, წყვეტს გამოყენების დამპროექტებელი.
- სისტემის უსაფრთხოების ჟურნალი – **Security Log**. არეგისტრირებს სისტემაში შეღწევის ყველა მცდელობას, წარმატებულს თუ წარუმატებელს. დუმილით, ეს ჟურნალი ჩართული არ არის. მას საჭიროების შემთხვევაში ააქტიურებს სისტემის ადმინისტრატორი **Group Policy** *(ჯგუფური პოლიტიკა)* კონსოლის მეშვეობით.
- სისტემის ჟურნალი – **System Log**. მასში რეგისტრირდება ოპერაციულ სისტემაში მომხდარი ხდომილობები.

შევნიშნოთ, რომ საკუთარი ჟურნალია გათვალისწინებული ზოგიერთი დამატებითი კომპონენტისთვისაც *(მაგალითად, **DNS** სამსახურისთვის).*

მომხმარებლებს შეუძლიათ, გაეცნონ **Application Log** და **System Log** ჟურნალების შემცველობას, **Security Log**-ის ნახვის უფლება კი მხოლოდ ადმინისტრატორს აქვს.

ჩავათვალიეროთ ხდომილობები **Event Viewer** უტილიტის მეშვეობით:

1. **Administrative Tools → Computer Management** ფანჯრის მარცხენა პანელში ორჯერ დავაწკაპუნოთ **System Tools**-ზე.
2. გამოდის ქვეხე. ორჯერ ვაწკაპუნებთ ჯერ სასურველ ჟურნალზე, შემდეგ კი მასში ჩვენთვის საინტერესო ხდომილობაზე. იმავე მიზნით, შეიძლება მივმართოთ კონტექსტურ მენიუში **Properties** პუნქტსაც.
3. ვეცნობით ინფორმაციას *(შეგვიძლია მისი წარმოდგენის სახის არჩევაც).* **OK.**

ჟურნალის ჩათვალიერების პროცესში ხშირად ინფორმაცია უკვე ძველდება. მისი განახლებისთვის გავცემთ ბრძანებას:

Action → Refresh

<u>მუშაობა ჟურნალების არქივებთან</u>

არქივის შესაქმნელად ასე ვიქცევით:

1. **Event Viewer** უტილიტის გაშვების შემდეგ გავცემთ ბრძანებას: **Save Log File As** *(**Action** მენიუდან).*
2. ვირჩევთ შესანახი ფაილის ტიპს *(ორობითი, ტექსტური, თუ გამყოფი მძიმეებიანი ტიპის).*
3. ვაწკაპუნებთ **Save**-ზე.
4. შენახულ ფაილს მოვძებნით იმავე ფანჯარაში, **Action** მენიუდან **Open Log File** ბრძანების გაცემის შედეგად.

<u>მუშაობა უსაფრთხოების ჟურნალთან</u>

სისტემის ჟურნალისაგან განსხვავებით, საჭიროა უსაფრთხოების ჟურნალი ხელით გავააქტიუროთ:

1. **Start → Run** და ვკრებთ **mmc /a** ბრძანებას. **OK.**
2. შემდეგ **Console** მენიუში ვირჩევთ **Add/Remove Snap-in** პუნქტს *(აღჭურვილობათა დაყენება-ამოგდება).* ვაწკაპუნებთ **Add**-ზე.
3. აღჭურვილობათა სიაში ამოვირჩევთ ჯგუფური პოლიტიკის ობიექტს. **Add.**
4. დიალოგურ ფანჯარაში **Select Group Policy Object Wizard** შემოთავაზებულ პარამეტრებს არ ვცვლით. **Finish,Close.**

5. ვბრუნდებით მართვის კონსოლის ფანჯარაში, ვაწკაპუნებთ **Audit Policy**-პარამეტრზე.
6. კონსოლის მარჯვენა პანელში ამოვირჩევთ იმ ხდომილებას, რომლის აუდიტის ჩატარებაც გვსურს.
7. გავცემთ ბრძანებას **Action → Security**
8. დიალოგურ ფანჯარაში **Audit Account Logon Events** შევირჩევთ ჩვენთვის საჭირო ხდომილებებს, რომელთა აუდიტი უნდა ჩატარდეს სისტემაში შესვლისას, **OK**.

დომენის კონტროლერზე სისტემის უსაფრთხოების აუდიტის განსახორციელებლად:

1. გავუშვებთ კონსოლს **Active Directory Users and Computers.**
2. კომპიუტერების სიიდან ამოვირჩევთ დომენის კონტროლერს.
3. გამოვიძახებთ მისთვის **Group Policy** ფანჯარას და ვაწკაპუნებთ **Audit Policy**-ზე.
4. მარჯვენა პანელში ავარჩევთ პარამეტრებს, რომელთა შეცვლაც გვსურს.
5. გავცემთ ბრძანებას **Action → Security**
6. დიალოგურ ფანჯარაში შევირჩევთ ჩვენთვის საჭირო ხდომილებებს. **OK**.

ასეთივე წესით ხდება ფაილებისა და საქაღალდეებისათვის აუდიტის გააქტიურება, რის შემდეგაც უნდა მივუთითოთ კონკრეტული ფაილები და საქაღალდეები:

1. ფაილის *(საქაღალდის)* კონტექსტურ მენიუში გამოვიძახებთ **Properties** ფანჯარას.
2. გადავდივართ **Security** ჩანართზე და ვაწკაპუნებთ **Advanced**-ზე.
3. მომდევნო ფანჯარაში გადავდივართ **Audit** ჩანართზე, ვაწკაპუნებთ **Add**-ზე და ვირჩევთ მომხმარებელს ან ჯგუფს, რომლისთვისაც გვსურს აუდიტის ჩატარება. **OK.**
4. **Auditing Entry** *(აუდიტის ობიექტი)* დიალოგურ ფანჯარაში შევირჩევთ ჩვენთვის საინტერესო ხდომილებებს. **OK**.

Windows-ის მთვლელები, Performance Monitor უტილიტა

Windows 2000 ოპერაციულ სისტემაში ჩართულია რიგი ხდომილობებისა, რომელთა დანიშნულებაა სისტემის წარმადობის კონტროლი. მთვლელი შეიძლება წარმოვიდგინოთ სიგნალების გენერატორად, რომლებიც ეგზავნება განსაზღვრულ მოწყობილობებს, რათა დადგინდეს შემდეგი პარამეტრები:

რა დრო სჭირდება სიგნალის ამოცნობას, დამუშავებას, უკან დაბრუნებას.

ზოგიერთი მთვლელი *(მაგალითად, პროცესორის წარმადობის მაკონტროლებელნი)* მუდმივადაა ჩართული, ნაწილი მომხმარებლის მიერ აქტიურდება *(მაგალითად, ხისტი დისკოების მუშაობის შემმოწმებელნი).*

პროგრამები და სამსახურები ხშირად თვითონ ამატებენ სისტემაში საკუთარ მთვლელებს *(მაგალითად, **WINS** სამსახური, **SQL Server** პროგრამა).*

მთვლელებიდან მიღებულ მონაცემებს წარმადობის ობიექტებს უწოდებენ. ზოგიერთი მთვლელი თითო გაითვალისწინება რაიმე პროცესისათვის, ზოგიერთი კი ერთდროულად რამდენიმე მოწყობილობას ემსახურება *(მაგალითად, ხისტ დისკოებს).*

მთვლელისათვის შეიძლება არსებობდეს "შვილობილი" მთვლელებიც.

მთვლელს სახელი შემდეგნაირი წესით შეიძლება მიენიჭოს:

კომპიუტერის-სახელი \ ობიექტის-სახელი \ მთვლელის-სახელი

მთვლელებთან ურთიერთობას ძალიან აადვილებს სისტემაში არსებული **Performance Monitor** უტილიტა, რომლის მეშვეობითაც წარმოებს სისტემის მუშაობის კონტროლი, სუსტი ადგილების გამოვლენა და პრობლემების გადაჭრა. ამ უტილიტის გრაფიკული ინტერფეისი სისტემის წარმადობას მომხმარებელს წარმოუდგენს დიაგრამის სახით. შესაძლებელია ხდომილობათა ჟურნალების წარმოებაც, ოღონდ სისტემა რომ არ გადაიტვირთოს, რეკომენდებულია, ჟურნალების განახლება 15 წუთზე ნაკლები დროის ინტერვალით არ განხორციელდეს.

"საბრძოლო მზადყოფნაში" მოვიყვანოთ უტილიტის რომელიმე მთვლელი:

1. **Administrative Tools → Perfomance** *(წარმადობა)*
2. **Perfomance Monitor** განყოფილებაში ვაწკაპუნებთ **Add**-ზე. ეკრანზე აისახება **Add Counters** *(მთვლელის დამატება)* დიალოგური ფანჯარა. აქვე შევნიშნოთ, რომ ამორჩეული მთვლელის შესახებ ინფორმაცია შეიძლება მივიღოთ **Explain**-ღილაკზე დაწკაპუნებით.
3. ვირჩევთ კომპიუტერს *(ნებისმიერს).*
4. მოვნიშნავთ იმ ობიექტებს, რომელთა წარმადობის შემოწმებაც გვსურს.
5. **Perfomance Counter** *(წარმადობის მთვლელები)* ფანჯარაში შევირჩევთ მთვლელებს. **Add, Close.**

ჟურნალის მონაცემების ასახვისათვის:

1. გავუშვებთ **Perfomance Monitor** უტილიტას.
2. მარჯვენა პანელში გამოძახებულ კონტექსტურ მენიუში ვაწკაპუნებთ **Properties**-ზე.
3. გადავდივართ **Source** *(წყარო)* ჩანართზე და ვაყენებთ **Log file** *(ჟურნალი)* გადამრთველს.
4. ვაწკაპუნებთ **Browse**-ზე, რათა მოვნახოთ საჭირო ფაილი. **OK.**

სხვა უტილიტებიდან აღსანიშნავია **Logs and** და **Alerts Network Monitor.**

ოპერაციულ სისტემაში ფუნქციონირებს აგრეთვე მომხმარებლისათვის შეტყობინებების გაგზავნის მექანიზმი. როცა არჩეული პარამეტრის მნიშვნელობა წინასწარ განსაზღვრულ რაიმე მნიშვნელობას გადააჭარბებს, ხდება ამის შესახებ მომხმარებლის ინფორმირება.

დანართი:

Windows NT Server4

სპეციალისტების აზრით, **Windows NT Server4**–ის შექმნა ახალ ერას მოასწავებს ქსელური ოპერაციული სისტემის განვითარებაში.

Windows NT ქსელი იყენებს "კლიენტ-სერვერ" არქიტექტურას. აქ სერვერი-კომპიუტერები მომსახურებას უწევენ კლიენტ-კომპიუტერებს. ცხადია, პირველთ უფრო რთული ფუნქციების შესრულება უწევთ. სერვერის ოპერაციულ სისტემას, ინდივიდუალური კომპიუტერისაგან განსხვავებით, დამატებით უნდა შეეძლოს:

ა) დაშორებულ ფაილურ სისტემასთან მუშაობა;
ბ) საერთო გამოყენებათა შესრულება;
გ) საერთო ქსელურ დისკოებზე ინფორმაციის შეტანა-გამოტანა;
დ) ცენტრალური პროცესორის დროის განაწილება ქსელურ პროცესებს შორის;
ე) ქსელის უსაფრთხოების უზრუნველყოფა.

სერვერის ოპერაციული სისტემა ფრიად საიმედო უნდა იყოს. არამცთუ მწყობრიდან მის გამოსვლას, გადატვირთვისას მუშაობის შენელებასაც კი მეტად არასასურველი შედეგების მოტანა შეუძლია.

კლიენტებზე სხვადასხვა ოპერაციული სისტემა შეიძლება დაყენდეს, მაგრამ, ცხადია, უკეთეს ვარიანტს **Windows NT Workstation** წარმოადგენს.

Windows NT Server4–ის მიერ საკუთარი ფუნქციების შესრულება ქსელის შემდეგ შესაძლებლობებს ეფუძნება:

1. <u>აპარატურული დამოუკიდებლობა</u> - აღნიშნავს იმ ფაქტს, რომ **Windows NT Server4** შეიძლება დაყენებულ იქნას კომპიუტერზე, რომელზეც გვაქვს DEC Alpha RISC, MIPS RISC, PowerPC ან Intel 80386-ზე უფრო მძლავრი პროცესორი.
2. <u>მრავალპროცესორულობა</u> - **Windows NT Server** შეიძლება დაყენდეს კომპიუტერზე, რომელიც 4-მდე პროცესორს შეიცავს. **Windows NT Workstation**-ს კი მაქსიმუმ ორ პროცესორთან შეუძლია მუშაობა.
3. <u>მრავალამოცანიანობა და მრავალნაკადურობა</u>. მრავალამოცანიანობა მხოლოდ იმ ფაქტს აღნიშნავს, რომ პროცესორი დროის ქვანტებს ანაწილებს რამდენიმე ამოცანას შორის, რაც მათი ერთდროულად შესრულების ილუზიას ქმნის. აქ მთავარი ის გახლავთ, რომ ერთი ამოცანის "ჩამოკიდება" ხელს არ უშლის დანარჩენებზე სისტემის მუშაობას. რაც შეეხება მრავალნაკადიანობას, მრავალპროცესორული კომპიუტერისათვის სისტემას მართლაც ძალუძს პროგრამის კოდის რამდენიმე ფრაგმენტის ერთდროულად შესრულება.
4. <u>უსაფრთხოება</u> - ქსელმა ინფორმაციის დაცვა უნდა უზრუნველყოს როგორც აპარატურული, ასევე პროგრამული ნაწილის მტყუნებისას. მან არ უნდა დაუშვას გარეშე პირების მიერ სისტემაში შეღწევა, ხოლო საკუთარ მომხმარებლებს კი მხოლოდ იმის გაკეთების ნება დართოს, რისი უფლებაც მათ გააჩნიათ.

5. **RAID** (დისკოთა მასივის) ტექნოლოგიის გამოყენება – ამაღლებს მტყუნებებისას სისტემის მდგრადობას. უკეთესი შედეგები მიიღწევა, როცა სპეციალურ მოწყობილობას შევიძენთ ამ ტექნოლოგიის სრულყოფილად განხორციელებისათვის. მაგრამ შესაძლებელია მისი პროგრამული განხორციელებაც. ამ შემთხვევაში ვიყენებთ სტანდარტულ ხისტ დისკოებს და **SCSI** ადაპტერს.
6. **NTFS - NT-**ის ფაილური სისტემა. ჩვენთვის ცნობილი **FAT (File Allocation Table -** ფაილების განლაგების ცხრილი) ფაილური სისტემისაგან განსხვავებით, **NTFS** გაცილებით უფრო მიესადაგება ქსელში მუშაობას. იგი უზრუნველყოფს:
   ა) 255 სიმბოლომდე სიგრძის მქონე ფაილთა სახელების არსებობას;
   ბ) **MS-DOS -** ისათვის უფრო მოკლე სახელების ავტომატურ გენერირებას;
   გ) დაზიანებული სექტორებიდან საიმედო უბანში ინფორმაციის ავტომატურ გადაწერას;
   დ) ცალკეულ ფაილებსა და საქაღალდეებში შეღწევის ნებართვის გაცემით უსაფრთხოების დაცვას;
   ე) დისკოს მტყუნების შემთხვევაში ფაილების აღდგენას ჟურნალის მეშვეობით.

Windows NT Server4-ს ბევრი სხვა ახალი შესაძლებლობაც გააჩნია, როგორიცაა მაგალითად: ინტერნეტში მუშაობა, ელექტრონული ფოსტით სარგებლობა და სხვ.

კომპიუტერი, რომელზეც ყენდება **Windows NT Server4** სისტემა, უნდა აკმაყოფილებდეს შემდეგ მოთხოვნებს:

პროცესორი სასურველია, პენტიუმზე ნაკლები შესაძლებლობების არ იყოს; ხისტ დისკოზე სისტემურ ფაილებს უნდა გამოეყოს არანაკლებ 150 მგბაიტი ზომის განყოფილება (შემდგომ განვმარტავთ); მონიტორი **VGA-**ზე უარესი არ უნდა იყოს; უნდა გვქონდეს კომპაქტ-დისკოების წაკითხვის მოწყობილობა (თუ სისტემის დაყენება ქსელიდან არ ხდება) და დისკოწამყვანი 3,5 დუიმიანი დისკეტებისათვის. ოპერაციული მეხსიერება სასურველია იყოს არანაკლებ 32 მგბაიტისა, ქსელური ადაპტერი უნდა ფიგურირებდეს **HCL (Hardware Compatiblity List)** სიაში.

დომენები

თავდაპირველად ლოკალური ქსელები იგებოდა ერთრანგიანი ქსელის პრინციპზე. აღნიშნულ ქსელში შემავალი კომპიუტერები ერთიანდებიან ე.წ. სამუშაო ჯგუფებში. თითოეულ ასეთ ჯგუფში კომპიუტერები თანაბარი უფლებებით სარგებლობენ. მათ შეუძლიათ ერთმანეთის რესურსებით სარგებლობა. თითოეული მომხმარებელი თვითონ განსაზღვრავს, რომელი რესურსი (ფაილი, საქაღალდე, დისკო, პრინტერი) გადასცეს საერთო მოხმარებაში. მაგრამ როცა კომპიუტერების (მომხმარებლების) რაოდენობა საკმაოდ დიდი ხდება, ასეთი ქსელის ფუნქციონირება დიდ სირთულეებს აწყდება. ცენტრალიზებული მართვის არარსებობა მოითხოვს თითოეული

კომპიუტერის ინდივიდუალურ გაწყობას. მომხმარებელს სჭირდება უამრავი პაროლის დამახსოვრება (თუ ნებართვის გაცემისას პაროლური სისტემა გამოიყენება).

ყოველივე აქედან გამომდინარე, შედარებით უფრო მსხვილი ლოკალური ქსელების შექმნისას იყენებენ "სერვერ-კლიენტს" არქიტექტურას, რომელიც, თავის მხრივ, ეყრდნობა კომპიუტერების დომენებში გაერთიანების კონცეფციას. ასეთი გადაწყვეტა უზრუნველყოფს ქსელის ადმინისტრირების გამარტივებასა და მოქნილობას.

დომენი Windows NT Server4-ის უმთავრესი ერთეულია. იგი წარმოადგენს კომპიუტერების ჯგუფს, რომლებიც იყენებენ მონაცემების საერთო ბაზას და უსაფრთხოების საერთო პოლიტიკას.

დომენში გაერთიანებული კომპიუტერები ორ ჯუფად შეიძლება დავყოთ:

- **სერვერები** – მმართველი კომპიუტერები. მათზე ყენდება **NT Server** ოპერაციული სისტემა;
- **კლიენტები** – სამუშაო სადგურები. ამ კომპიუტერებზე შესაძლებელია დაყენდეს **Windows**-ის ოჯახის ნებისმიერი წარმომადგენელი, თუმცა, ცხადია, უკეთესი ვარიანტია **Windows NT Workstation-**სადმი მიმართვა.

დომენის ცენტრალური ელემენტია ის სერვერი-კომპიუტერი, რომელსაც დაკისრებული აქვს **დომენის ძირითადი კონტროლერის როლი. (Primary domain controller - PDC).** ძირითადი კონტროლერი დომენის ყველაზე მნიშვნელოვანი კომპიუტერია. იგი ახორციელებს დომენის უსაფრთხოების პოლიტიკას მომხმარებელთა სააღრიცხვო ბარათების ცენტრალიზებლად შენახვის მეშვეობით. ამ მიზანს ემსახურება სპეციალური პროგრამა **User Manager for Domains** (დომენების მომხმარებელთა მენეჯერი).

დომენის ძირითად კონტროლერს შეუძლია იყოს ფაილური თუ აპარატურული რესურსების მიმწოდებელიც ქსელში ჩართული კომპიუტერებისათვის, თუმცა ზოგჯერ ეს სასურველი არც არის. ეს როლი, როგორც წესი, სხვა სერვერებს [1] ეკისრებათ. მაგალითად, დომენში რომელიმე სერვერი შეიძლება გამწესებული იყოს დოკუმენტების ბეჭდვაზე და ა.შ.

გარდა ძირითადი კონტროლერისა, დომენში არსებობს მინიმუმ ერთი ცალი **დომენის სარეზერვო კონტროლერი (Backup domain controller - BDC)** და ასევე მინიმუმ ერთი ცალი სამუშაო სადგური (კლიენტი).

---

[1] *სერვერი* ***Windows NT*** *ქსელში ეწოდება კომპიუტერს, რომელიც მუშაობს* ***Windows NT Server*** *ოპერაციული სისტემის მართვის ქვეშ, მაგრამ დომენის კონტროლერს არ წარმოადგენს.*

სარეზერვო კონტროლერზე ინახება მომხმარებელთა სააღრიცხვო ბარათების ასლები.

სააღრიცხვო ბარათებში ცვლილებების შეტანა შეიძლება მოხდეს მხოლოდ დომენის ძირითად კონტროლერზე და ამის შემდეგ გავრცელდეს სარეზერვო კონტროლერებზე. უშუალოდ ასლებში ცვლილებების შეტანა არ ხდება, შესაძლებელია მხოლოდ მათში ჩაწერილი ინფორმაციის წაკითხვა ძირითადი კონტროლერის გამორთვისას, რაც მომხმარებლებს საშუალებას აძლევს გაიარონ რეგისტრაცია და იმუშაონ სისტემაში. მაგრამ თუ ძირითადი კონტროლერი მწყობრიდან გამოვიდა, სარეზერვოს შეუძლია მომხმარებლების რეგისტრაციის თავის თავზე აღება და ამა თუ იმ რესურსზე შეღწევის უფლების შემოწმება.

აღვნიშნოთ, რომ საჭიროების შემთხვევაში ადმინისტრატორს შეუძლია სარეზერვო კონტროლერი ძირითადად აქციოს.

<u>სამუშაო სადგური</u>. მასზე აყენებენ **Windows NT Workstation**-ს.

შესაძლოა ზოგიერთი სადგური სამუშაო ჯგუფში შედიოდეს და არ ეკუთვნოდეს დომენს, მაგრამ ეს ნიშნავს იმ უპირატესობებზე უარის თქმას, რომლებიც დამახასიათებელია "სერვერი-კლიენტი" არქიტექტურისათვის.

<u>ნდობითი დამოკიდებულებანი</u>

მრავალდომენიან ქსელში თითოეული დომენი ფუნქციონირებს როგორც ცალკეული ქსელი სააღრიცხვო ბარათების საკუთარი მონაცემთა ბაზით. მაგრამ ხშირად ხდება საჭირო, რომ ამა თუ იმ დომენის ზოგიერთმა მომხმარებელმა სხვა დომენის რესურსებითაც ისარგებლოს. ეს საკითხი წყდება დომენებს შორის (უფრო ზუსტად, დომენების კონტროლერებს შორის) ე.წ. **ნდობითი დამოკიდებულების** დამატებით. დამოკიდებულებანი სხვადასხვა სახის შეიძლება იყოს:

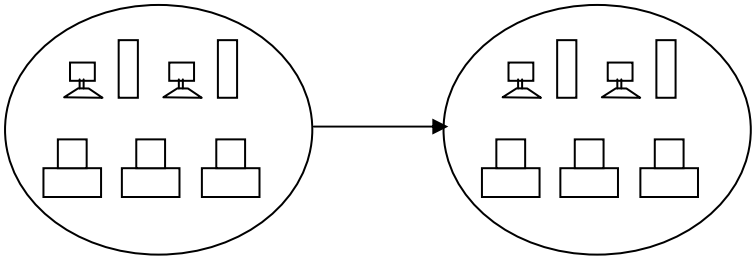
1. <u>დომენებს შორის ნდობითი დამოკიდებულება არ არსებობს.</u>

ასეთ შემთხვევაში, თუ მომხმარებელს სურს რამდენიმე დომენის რესურსებით ისარგებლოს, იგი თითოეულში უნდა დარეგისტრირდეს. ადმინისტრატორმა მას ყოველ დომენში უნდა გაუხსნას სააღრიცხვო ბარათი. ამასთან, მომხმარებელს შეიძლება დომენებში ერთიდაიგივე ან სხვადასხვა სახელები ჰქონდეს.

ვთქვათ, მომხმარებელი მუშაობს რომელიმე დომენში და მას დასჭირდა სხვა დომენის რესურსებით სარგებლობა. ასეთ შემთხვევაში იგი უნდა გამოვიდეს ერთი დომენიდან და შევიდეს მეორეში, რაც მუშაობის ეფექტიანობას ამცირებს. გარდა ამისა, ყოველ დომენში მომხმარებლებს დასჭირდებათ თითო სააღრიცხვო ბარათი, რაც ადმინისტრატორს მათში კორექტივების შეტანას უძნელებს.

2. <u>ცალმხრივ მიმართული ნდობითი დამოკიდებულებანი.</u>

გრაფიკულად ასეთ დამოკიდებულებებს აქვს შემდეგი სახე:



დომენი "ფინანსები" დომენი "მარკეტინგი"

დომენი "ფინანსები" ენდობა დომენ "მარკეტინგს"

(ანუ მის მომხმარებლებს).

თუ მომხმარებელს შეუძლია დომენ "მარკეტინგში" შესვლა, მას ეძლევა დომენ ფინანსებში" შესვლის შესაძლებლობაც, მაგრამ ეს არ ნიშნავს, რომ მომხმარებელი ავტომატურად იღებს ამ დომენის რესურსებით სარგებლობის უფლებას. ამისათვის ადმინისტრატორმა სპეციალური ნებართვები უნდა გასცეს.

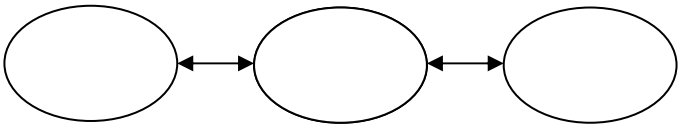
ცალმხრივ მიმართული ნდობითი დამოკიდებულებებისას, ცხადია, ნდობა საწინააღმდეგო მიმართულებით არ ვრცელდება, ე. ი. ის მომხმარებლები, რომლებიც შევიდნენ დომენ "ფინანსებში", ვერ შევლენ დომენ "მარკეტინგში". თუ ეს აუცილებელია, მაშინ დამატებით უნდა იქნას ორგანიზებული კიდევ ერთი ცალმხრივ მიმართული ნდობითი დამოკიდებულება – ამჯერად, დომენი "მარკეტინგი" უნდა ენდოს დომენ "ფინანსებს".

3 <u>ორმხრივ მიმართული ნდობითი დამოკიდებულებანი</u>

ეს დამოკიდებულებანი უფრო მეტი მოქნილობით ხასიათდებიან და მნიშვნელოვნად ამარტივებენ ქსელში ადმინისტრირებას. ასეთი დამოკიდებულება, ფაქტიურად, შეიძლება შეიქმნეს ორი ცალმხრივ მიმართული დამოკიდებულების ორგანიზებით.

ამ დამოკიდებულების მთავარი ღირსება ის გახლავთ, რომ მომხმარებელთა სააღრიცხვო ბარათები მხოლოდ ერთხელ უნდა შევქმნათ (უმჯობესია ძირითად დომენში).

<u>ნდობითი დამოკიდებულებების ტრანზიტი არ ხდება.</u>



დომენი "წარმოება" დომენი "ფინანსები" დომენი "მარკეტინგი"

მაგალითად, თუ შევედით პირველ დომენში, ვერ მოვხვდებით მესამეში და პირიქით. თუ ეს აუცილებელია, ორგანიზებულ უნდა იქნას დამატებითი ორმხრივი ნდობითი დამოკიდებულება ამ დომენებს შორის.

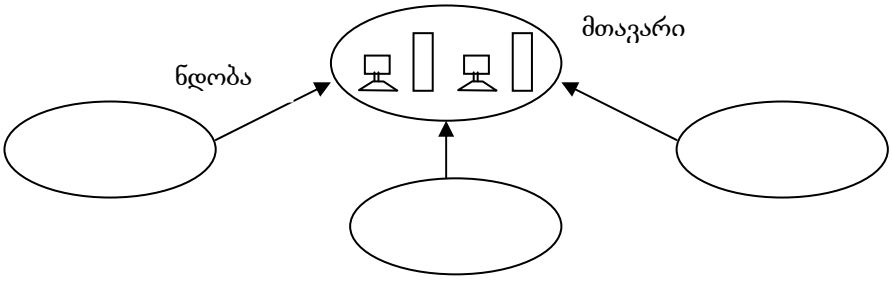
Windows NT Server4-ში არსებობს დომენებს შორის ნდობითი დამოკიდებულებების სტრუქტურის ოთხი მოდელი:

- *ერთდომენიანი;*
- *ერთი მთავარი დომენით;*
- *რამდენიმე მთავარი დომენით;*
- *სრული ნდობითი დამოკიდებულებით.*

ეს მოდელები და მათი მოდიფიკაციები ერთდროულად შეიძლება გამოვიყენოთ დიდ ქსელებში.

ერთდომენიანი მოდელი. მარტივი მოდელია. სერვერები და კლიენტები ერთ დომენში შედიან, ამიტომ ლოკალური და გლობალური ჯგუფები ერთმანეთს ემთხვევა [2]. ყოველ ადმინისტრატორს შეუძლია თითოეული სერვერის მართვა. ცხადია, საჭირო აღარაა ნდობითი დამოკიდებულებების ორგანიზებაც. ამ მარტივ დომენზე უარს მაშინ ამბობენ, როცა მომხმარებელთა ჯგუფებს იშვიათად სჭირდებათ ერთმანეთი, ან როდესაც საწარმო (ორგანიზაცია) დიდია და/ან განტოტვილი სტრუქტურის.

მოდელი ერთი მთავარი დომენით. ეს მოდელი უზრუნველყოფს მომხმარებელთა ცენტრალიზებულ მართვას და ახდენს რესურსების სტრუქტურიზაციას განყოფილებების მიხედვით. ამ მოდელში გვაქვს ერთი *მთავარი დომენი* (master domain) და, როგორც წესი, რამდენიმე *რესურსების დომენი* (secondary domain), რომელთაგან თითოეული ცალმხრივ მიმართულ ნდობით დამოკიდებულებაშია მთავარ დომენთან.



ეს მოდელი განსაკუთრებით მისაღებია ისეთი ორგანიზაციისათვის, რომელშიც მომხმარებლების რიცხვი არც ისე დიდია, ხოლო რესურსების რიცხვი საკმაოა და აზრი აქვს მათ ჯგუფებად დაყოფას.

[2] ლოკალურ ჯგუფში გაერთიანებულ მომხმარებლებს მხოლოდ საკუთარი დომენის რესურსებით შეუძლიათ სარგებლობა, ხოლო გლობალური ჯგუფის წევრებს - იმ დომენის რესურსებითაც, რომლებიც ნდობას უცხადებენ მათ დომენს (იხ. ქვემოთ).

ყოველი მომხმარებლის სააღრიცხვო ბარათი რეგისტრირდება მთავარ დომენში. მთავარ დომენშივე იქმნება გლობალური ჯგუფები, სხვა დომენებში – ლოკალური ჯგუფები.

ამრიგად, ამ მოდელში მთავარი დომენის უპირველესი ამოცანაა სააღრიცხვო ბარათების ცენტრალიზებული გაძღოლა. ცხადია, საიმედოობის თვალსაზრისით, აუცილებელია მთავარ დომენში თუნდაც ერთი სარეზერვო კონტროლერის არსებობა. სხვა დომენები კი რესურსების მომწოდებელთა როლში გამოდიან.

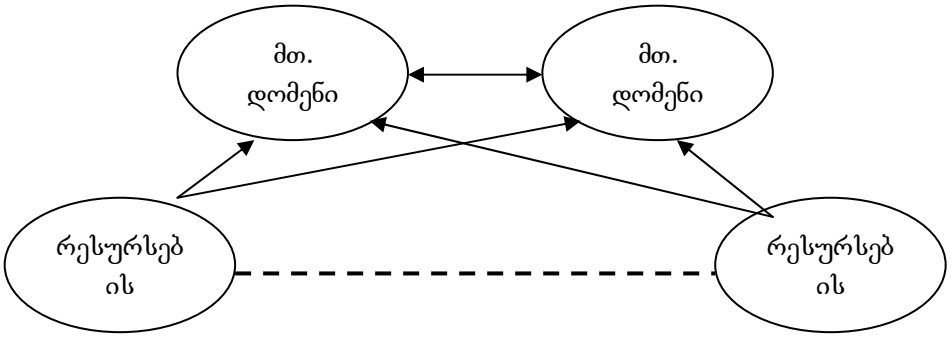
ასეთი გადაწყვეტის ღირსებაა ის, რომ მომხმარებელი თავდაპირველად მხოლოდ საკუთარ რესურსებს ხედავს, რაც ამარტივებს და აჩქარებს საჭირო რესურსის მოძებნას.

<u>მოდელი რამდენიმე მთავარი დომენით.</u> როცა მომხმარებელთა რიცხვი იზრდება, წინა მოდელი ვეღარ უზრუნველყოფს ქსელის ეფექტიან მუშაობას. კერძოდ, ქსელის წარმადობა ეცემა იმის გამო, რომ თითოეული სააღრიცხვო ბარათის ნამდვილობას ერთი მთავარი დომენი ამოწმებს.

ასეთ შემთხვევაში, თუ სურთ შეინარჩუნონ სააღრიცხვო ბარათების ცენტრალიზებულად გაძღოლის შესაძლებლობა, გადადიან რამდენიმე მთავარდომენიან მოდელზე.

სააღრიცხვო ბარათები მთავარ დომენებს შორის განაწილდება ან ფორმალურად, (მაგალითად, ალფაბეტის მიხედვით დალაგებულ ჯგუფებად) ან ლოგიკური კავშირების მიხედვით. პირველ შემთხვევაში მეტი შრომა იხარჯება გლობალური ჯგუფების შექმნა–გაძღოლაზე. მეორე შემთხვევაში კი სირთულეებს ვაწყდებით, როცა ორგანიზაციის სტრუქტურა მოქნილია – თანამშრომლები ხშირად იცვლიან სამუშაო ადგილებს.

მოდელს აქვს ასეთი სქემური სახე:



მოდელი რამდენიმე მთავარი დომენით და სრული ნდობითი დამოკიდებულებებით. ნებისმიერ ორ დომენს შორის მყარდება ორმხრივი ნდობითი დამოკიდებულება, მომხმარებლის სააღრიცხვო ბარათები კი ინახება ერთ-ერთ, ე.წ. Home Domain-ში.

ამ მოდელს იშვიათად იყენებენ, რადგანაც ნდობითი დამოკიდებულებების რიცხვი n*(n-1) დიდ მნიშვნელობას ღებულობს არცთუ დიდი n-სათვის.

მომხმარებლის სააღრიცხვო ბარათის არსი.

იგი შეიცავს: მომხმარებლის გვარს, პაროლს, შეზღუდვებს ქსელში მუშაობაზე. ბარათი ერთ-ერთ დომენში ინახება. თუ მომხმარებელი ისეთ დომენში შესვლას ცდილობს, რომელშიც მისი ბარათი რეგისტრირებული არ არის, ამის ნება დაერთვება მხოლოდ იმ შემთხვევაში, თუ ეს დომენი ნდობას უცხადებს მომხმარებლის სააღრიცხვო ბარათის შემცველ დომენს.

მომხმარებლები შეიძლება გაერთიანებული იქნან ***ლოკალურ*** და ***გლობალურ*** ჯგუფებში, რომელთაც ერთნაირი ნებართვები და შეღწევის უფლებები გააჩნიათ.

ჯგუფში გაერთიანება საშუალებას იძლევა საჭიროების შემთხვევაში სწრაფად მოხდეს მისი ყველა წევრის პრივილეგიების კორექტირება.

ჯგუფები – მომხმარებელთა პრივილეგიების მართვის ინსტრუმენტი.

Windows NT Server შეიცავს ჩაშენებული გლობალური და ლოკალური ჯგუფების მთელ რიგს.

ლოკალური ჯგუფი მხოლოდ იმ დომენში ფუნქციონირებს, რომელშიც იგი შეიქმნა. ***გლობალური ჯგუფის*** გავლენა კი ყველა იმ დომენზე ვრცელდება, რომლებიც მის მშობლიურ დომენს ნდობას უცხადებენ.

ჯგუფის არსებობა შემდეგ სერვისს უზრუნველყოფს:

ა) შესაძლებელი ხდება პრივილეგიების ერთბაშად კორექტირება მისი ყველა წევრისთვის;

ბ) ჯგუფში ახალი წევრის დამატებისას ეს წევრი ავტომატურად ღებულობს ჯგუფისთვის განკუთვნილ პრივილეგიებს;

ჯგუფების შექმნა მომხმარებელსაც შეუძლია, მაგრამ ჯერ უმჯობესია, კარგად შევისწავლოთ ის შესაძლებლობები, რომლებსაც გვაწვდის ჩაშენებული ჯგუფების წევრობა.

უსაფრთხოება Windows NT-ში.

Windows NT-ში უსაფრთხოების დონე შეესაბამება აშშ-ის თავდაცვის სამინისტროს მიერ დაწესებულ სტანდარტებს.

ქსელის დაცვის მექანიზმი გულისხმობს 4 კატეგორიას:

1. *ქსელის ფიზიკური დაცვა;*
2. *მომხმარებლების დაცვა;*
3. *ფაილების დაცვა;*
4. *გარედან შეღწევისაგან ქსელის დაცვა.*

ფიზიკური დაცვა გულისხმობს არა მარტო რკინის კარებს და რკინის კარადას, არამედ სერვერის ცალკე, კარგად დაცულ ოთახში მოთავსებასაც.

მომხმარებლის დაცვა ხორციელდება:

ა) მისთვის საჭირო რესურსებზე შეღწევის ნებართვის გაცემით;

ბ) სხვა რესურსებზე შეღწევის შეუძლებლობით, ზოგჯერ მათი ჩათვალიერების აკრძალვითაც კი.

ყოველივე ეს მიიღწევა პაროლების მეშვეობით. იშვიათ შემთხვევაში განსაკუთრებით საიდუმლო ინფორმაციის გასაცნობად მომხმარებელს შეიძლება მოუხდეს რამდენიმე პაროლის დამახსოვრებაც. მაგრამ ამ გზას უნდა ვერიდოთ, რათა მომხმარებელმა არ მიმართოს პაროლების დასამახსოვრებლად ადვილ ხერხს – მათ ჩაწერას.

უფრო დაწვრილებით განვიხილოთ ფაილების დაცვის მექანიზმი. ისიც ორ ასპექტს მოიცავს:

ა) ფაილთან შეღწევის მართვა;

ბ) ფაილის მთლიანობის დაცვა.

თავდაპირველად აღვნიშნოთ, რომ **Windows NT**-ის მიერ მოწოდებული დაცვის მექანიზმის სრულყოფილად გამოყენებისთვის ინფორმაცია დისკოებზე ჩაწერილი უნდა იყოს ახალ ფაილურ სისტემა **NTFS**-ში. დასაშვებია ზოგიერთ ტომზე ძველი – **FAT (File Allocation Table)** სისტემის გამოყენებაც. მაგრამ ამ შემთხვევაში შესაძლებელია ვარეგულიროთ მხოლოდ საქაღალდეების დონეზე შეღწევა და არა ცალკეული ფაილებისადმი მიდგომა. **NTFS**-სისტემას ის უპირატესობაც გააჩნია, რომ ფაილების და საქაღალდეების დაცვა შესაძლებელია საკუთარ კომპიუტერზე მომუშავე მომხმარებლებისგანაც.

NTFS-სისტემის სხვა ღირსებებია:

1. *დიდ ხისტ დისკოებზე მონაცემების უფრო ეფექტიანი შენახვა;*
2. *ფაილებთან უფრო სწრაფი შეღწევა;*
3. *დისკოზე პრობლემების შექმნისას მონაცემების უკეთ აღდგენა (ტრანზაქციების ჟურნალის შემოღების შედეგად);*
4. *ფაილების შეკუმშვა (Dos-ში ცნობილი დისკოების შეკუმშვის პროგრამები Windows NT-ში არ მუშაობენ).*

NTFS-სისტემის ნაკლია ის, რომ მასში ჩაწერილი ინფორმაცია სხვა ოპერაციულ სისტემებში ვერ დამუშავდება (თუმცა ქსელთიდან მიმართვის შემთხვევაში ეს შეზღუდვაც იხსნება).

აღსანიშნავია, რომ შესაძლებელია **FAT** ტომების **NTFS** ტომებად გარდაქმნა. **FAT**-სისტემაში თითოეული ფაილისათვის ინახება სახელი, ზომა, ბოლო ცვლილების დრო. **NTFS** –ში, გარდა ამ ცნობებისა, თითოეული ფაილისა და საქაღალდისათვის მოცემულია შეღწევის მართვის სიაც – **ACL (access control list)** (იხ. ქვემოთ). თვისებების ფანჯარაში მათთვის დამატებულია ჩანართი *"უსაფრთხოება"*. თუ ჩვენ გვაქვს სათანადო უფლება, შეგვიძლია დავათვალიეროთ ეს ჩანართი **Permission** (ნებართვები) ღილაკზე დაწკაპუნებით კი გამოვიყვანოთ **ACL**-სია და გავიგოთ, რომელ ჯგუფს თუ მომხმარებელს რა უფლებები აქვს ამ ობიექტზე – შეღწევის რომელი ტიპია მისთვის დანიშნული.

შენიშვნა: ამ სიაში კორექტირების შეტანის უფლება მაშინ გვეძლევა (რისთვისაც გათვალისწინებულია Add და Remove ღილაკები), თუ ჩვენ ამ ფაილის მფლობელი ვართ (იხ. ქვემოთ).

ამასთან, შესაძლებელია, რომ მომხმარებელს დაერთოს საქაღალდესთან მუშაობის ნება, მაგრამ არ ჰქონდეს მასში არსებულ რომელიმე ფაილში შეღწევის უფლება.

ეს ნებართვებია (ფაილისთვის):

READ (R), WRITE (W), EXECUTE (E), DELETE (D), CHANGE PERMISSION (P) და TAKE OWNERSHIP (O) – მფლობელის შეცვლა.

ამ ელემენტარული ნებართვების საფუძველზე ფაილებისთვის გათვალისწინებულია შემდეგი შეღწევის ტიპები:

1. ***No Access*** *– ნებართვები გაცემული არაა;*
2. ***READ*** *– წაკითხვა და შესრულება* ***(RX)****;*
3. ***Change*** *– წაკითხვა, ჩაწერა, შესრულება და ამოგდება* ***(RWXD)****;*
4. ***Fulle Control*** *–ყველა ნებართვა* ***(All)****.*

ზოგჯერ ფაილთან შეღწევის ეს 4 წინასწარ განსაზღვრული ტიპი საკმარისი არ არის. მაგალითად, მაშინ, როცა გვსურს მომხმარებელს მივცეთ ფაილის მხოლოდ მოდიფიცირების, მაგრამ არა მისი ამოგდების უფლება. ასეთ შემთხვევაში **ACL**-სიაში მოცემული მომხმარებლისთვის შეღწევის ტიპების სიიდან ავირჩევთ მნიშვნელობას – **Special Access** (სპეციალური შეღწევა) და იქ მოვახდენთ საჭირო არჩევანს.

საქაღალდეებში შეღწევის ტიპის განსაზღვრისას ცვლილებანი შეიძლება გავავრცელოთ მასში მყოფ ფაილებსა და საქაღალდეებზეც. ამიტომ ყოველი მომხმარებლისა თუ ჯგუფისათვის გამოდის ორი სია: პირველი განსაზღვრავს საქაღალდეებთან მუშაობის ნებართვებს, ხოლო მეორე – ამ ნებართვებს საქაღალდეებში მყოფი ფაილებისთვის. თუ ფაილებისთვის განკუთვნილ სიაში წერია **Not Specified**, ეს ნიშნავს, რომ მომხმარებელს თუ ჯგუფს არ შეუძლია ფაილებით

სარგებლობა (ეს ფაილები შექმნილია სხვა მომხმარებლის მიერ). მათთან მუშაობის ნებართვა ცალკე უნდა გაიცეს.

ჩამოვთვალოთ შეღწევის ტიპები საქაღალდეებისთვის:

1.	***No Access***	***(None)***	***(None)***
2.	***List***	***(RX)***	***(Not Specified)***[3]
3.	***Read***	***(RX)***	***(RX)***
4.	***Add***	***(WX)***	***(Not Specified)***
5.	***Add&READ***	***(RWX)***	***(RX)***
6.	***Change***	***(RWXD)***	***(RWXD)***
7.	***Full control***	***(All)***	***(All)***[4]

აქაც გვეძლევა ფაილებთან შეღწევის სპეციალური ტიპის არჩევის საშუალება. იგივე საშუალება ემატება საქაღალდეებისთვისაც.

როცა ვახდენთ საქაღალდეში შეღწევის ნებართვების ცვლილებას, იმისათვის, რომ მოხდეს ცვლილებები მასში არსებული ფაილებისთვისაც, მოვნიშნავთ ოფციას**–Replace Permissions On Existing Filles.** იგივე შეიძლება გავაკეთოთ ქვესაქაღალდეებისათვისაც და მათში არსებული ფაილებისათვის მეორე ოფციის მონიშვნით – **Replace Permissions On Subdirectories.**

ნებართვების შეცვლა შეიძლება ერთდროულად მოვახდინოთ რამდენიმე ფაილისთვის ან საქაღალდისთვის, რისთვისაც მათ წინასწარ მოვნიშნავთ.

ნებართვების შეცვლის გარდა, შეიძლება ახალი წევრების დამატებაც. იმავე ფანჯარაში, რომელშიც გამოდის **ACL**-სია, ვაწკაპუნებთ **Add** ღილაკზე, ჩამოშლად სიაში ვირჩევთ დომენს ან კომპიუტერს. გამოდის დიალოგის ფანჯარა . . .

ახლა უკვე შეიძლება ზოგიერთი დასკვნის გამოტანა: პროგრამულ ფაილებს და მათ შემცველ საქაღალდეებს უნდა ახასიათებდეს შეღწევის ტიპი **READ**, რაც დაიცავს მათ შემთხვევითი ცვლილებებისა და ვირუსებისაგან. მაგრამ ეს საკმარისი არ გახლავთ იმ მომხმარებლებისაგან თავის დასაცავად, რომელთაც შეუძლიათ ნებართვის შეცვლა.

რაც შეეხება მონაცემთა ფაილებს, უმჯობესია თითოეულ მომხმარებელს ჰქონდეს საკუთარი საქაღალდე სრული შეღწევის ტიპით მისთვის, ხოლო სხვა მომხმარებლებისათვის კი – **No Access** ტიპის მქონე.

საერთო მონაცემებისთვის კი ვქმნით ცალკე საქაღალდეს, სხვადასხვა მომხმარებლისთვის ნებართვების სხვადასხვა ტიპის მქონეს.

გარედან შეღწევისაგან დაცვა ხორციელდება პაროლების შერჩევისა და ცვლილებების სწორად დაგეგმვით. მაგალითად, მცდარი

---

[3] –შეიძლება მხოლოდ საკუთარ ფაილთან მუშაობა.

[4] –შეიძლება შეღწევის ტიპის და ფაილის მფლობელის შეცვლაც.

პაროლის შეტანის ცდების 3-მდე შეზღუდვით, დაშორებულ მომხმარებლებთან სისტემის მხრიდან დაკავშირებით და სხვ.

ზოგჯერ საჭიროა შევცვალოთ ფაილის მფლობელიც. მივმართავთ ჩანართს **Properеties → Securite** და ვაწკაპუნებთ **Ownership** ღილაკზე. ვიგებთ მფლობელის ვინაობას, ვაწკაპუნებთ **Take Ownership** –ზე.

ახალი ფაილებისა და საქაღალდეებისთვის შეღწევის პარამეტრები მშობლის პარამეტრებით განისაზღვრება. იგივე ხდება სხვა ტომზე მათი გადანაცვლება – კოპირებისას, იმავე ტომზე კი პარამეტრები შენარჩუნდება.

მომხმარებლები და მომხმარებელთა ჯგუფები

როგორც უკვე ვნახეთ, **Windows NT** ოპერაციული სისტემის დაყენებისას ავტომატურად იქმნება რამდენიმე ჩაშენებული ჯგუფი და ორი სააღრიცხვო ჩანაწერი: **Administrator** და **Guest**. ადმინისტრატორის სააღრიცხვო ჩანაწერის გამოყენების შედეგად შეგვიძლია სხვა სააღრიცხვო ჩანაწერების შექმნა, თვით ადმინისტრატორისთვისაც კი (რომელთა რიცხვი შეიძლება ერთზე მეტი იყოს). **Guest** ჩანაწერი დუმილით გამორთულია. მისი ამოგდება არ შეიძლება.

დამწყებისთვის სავსებით საკმარისია სისტემაში ***ჩაშენებული ჯგუფებით*** სარგებლობა, თუმცა შემდგომში შეიძლება სურვილი დაგვებადოს, ჩვენ თვითონაც შევქმნათ ჯგუფები.

ჯგუფებში გაერთიანებულ მომხმარებლებს ერთნაირი ***შესაძლებლობები*** და ***უფლებები*** გააჩნიათ. თუ მომხმარებელი რამდენიმე ჯგუფშია გაერთიანებული, მაშინ ეს შესაძლებლობები და უფლებები ჯამდება.

ჩაშენებული ჯგუფისთვის შესაძლებლობა არის ის, რისი წართმევაც, უფლებისგან განსხვავებით, არ ხდება. ამიტომ თუ გვსურს, რომელიმე მომხმარებელს რაიმე კონკრეტული შესაძლებლობა წავართვათ, იგი შესაბამისი ჯგუფიდან უნდა გამოვიყვანოთ.

ამრიგად, ჯგუფები შეიძლება დავყოთ ასეთი ნიშნის მიხედვით: ***ჩაშენებულია*** იგი, თუ ***მომხმარებლის მიერ შექმნილი.***

უკვე ვიცით, რომ ჯგუფები არსებობს აგრეთვე ***გლობალური*** და ***ლოკალური***.

გლობალურია ჯგუფი, თუ მისთვის უფლებების და ნებართვების გაცემა შეიძლება მოხდეს რამდენიმე დომენზე. გლობალური ჯგუფის შექმნა და ადმინისტრირება დასაშვებია მხოლოდ დომენის კონტროლერზე. გლობალური ჯგუფების დასახელებები, ორის გარდა, იწყება **Domain** სიტყვით.

ლოკალურ ჯგუფებზე უფლებები და ნებართვები გაიცემა მხოლოდ იმ დომენის ფარგლებში, რომელშიც იგი შეიქმნა. მისი შექმნა და ადმინისტრირება შეიძლება დომენში შემავალ ნებისმიერ კომპიუტერზე.

გლობალური ჯგუფის მახასიათებლები:

- *მასში შეიძლება შედიოდეს მხოლოდ ამ დომენში აღრიცხვაზე აყვანილი მომხმარებლების საალრიცხვო ჩანაწერები, თან უშუალოდ ანუ არა ჯგუფის მეშვეობით;*
- *გლობალურ ჯგუფს რესურსები შეიძლება დაენიშნოს ნდობის გამომცხადებელ დომენებში;*
- *თვითონ გლობალური ჯგუფი დასაშვებია ლოკალური ჯგუფის შემადგენლობაში შედიოდეს.*

ლოკალური ჯგუფის მახასიათებლები:

- *ლოკალური ჯგუფში შეიძლება შედიოდეს გლობალური ჯგუფები მოცემული და ნდობის გამომცხადებელი დომენებიდან და, ცხადია, მომხმარებელთა ცალკეული საალრიცხვო ჩანაწერებიც;*
- *ლოკალური ჯგუფი ვერ შევა სხვა ჯგუფის შემადგენლობაში (ვერც გლობალურსა და ვერც ლოკალურში).*

ლიტერატურა

1. Освой самостоятельно Microsoft Windows 2000 Server за 24 часа. Барри Сосински, Джереми Московиц, «Вильямс» , Москва, Санкт-Петербург, Киев, 2002.

2. Microsoft Windows NT Server 4.0. Чарльз Рассел, Шерон Кроуфорд, Microsoft Press, Санкт-Петербург, Москва, Харьков, Минск, 1988.

3. Нэд Снелл. Internet, «Вильямс», 2002

4. Учебный курс «Компьютерные сети» , Microsoft Press. Санкт-Петербург, 1999.

შინაარსი