ივანე ჯავახიშვილის სახელობის თბილისის

სახელმწიფო უნივერსიტეტი

დავით ხარაიშვილი

Windows ოპერაციული სისტემის ვირტუალიზაცია Linux - ის გარემოში

სამაგისტრო პროგრამა: ინფორმაციული ტექნოლოგიები

ნაშრომი შესრულებულია ინფორმაციული ტექნოლოგიების მაგისტრის

აკადემიური ხარისხის მოსაპოვებლად

ხელმძღვანელი: ფიზ. მათ მეცნიერებათა

კანდიდატი, აკადემიური დოქტორი ზურაბ

მოდებაძე

თბილისი 2015

ანოტაცია

- წინამდებარე დოკუმენტი წარმოადგენს ერთი ოპერაციული სისტემის ვირტუალიზაციას მეორე ოპერაციული სისტემის გარემოში. ამ შემთხვევაში Windows ვირტუალურად არის წარმოდგენილი Linux - ის გარემოში.
- განხილულია აღნიშნული სპეციფიკის თავისებურება, მოცემული სტრუქტურის პრაქტიკული განხორციელება, ვირტუალიზაციის სახეები და შესაბამისი სტანდარტები.
- მოყვანილია ვირტუალიზაციის მაგალითები და მათი გადაწყვეტის გზები.
- წარმოდგენილია ვირტუალიზაციის მიზანი და მისი პრაქტიკული გამოყენება.
- ჩამოთვლილია ვირტუალიზაციის მექანიზმის ნაკლოვანებები და უპირატესობები, უარყოფითი და დადებითი მხარეები.

Annotation

- Previous document show us one operative system virtualization in environment of second operative system. In this case virtually windows presented in Linux environment.
- We discuss about marked specificity originality, practical realization of given structure, faces of virtualization and corresponding standard.
- This presentation show some cases of virtualization and ways of its resolution.
- Presentation of virtualization target and it's practical usage.
- There are listed the benefits and drawbacks and advantages and disadvantages of virtualization mechanism.

სარჩევი

შესავალი

ვირტუალიზაცია ყველას თავისებურად, განსხვავებულად ესმის, მაგრამ რაც მის ძირითად აზრს აერთიანებს, არის ამ სისტემის ერთი წარმოდგენა: ურთიერთდამოუკიდებელი სერვერებისა და ოპერაციული სისტემების ერთ კომპიუტერზე განთავსება არის მოცემული სისტემის ვირტუალიზაცია. ვირტუალიზაცია ეწოდება ასევე ისეთ სიტუაციას, როდესაც რამდენიმე კომპიუტერი წარმოდგენილია ერთ ცალკეულ კომპიუტერად, ჩვეულებრივ ამას უწოდებენ კლასტერულ სერვერს - Grid Computing.

ვირტუალიზაციის ისტორია 40 წელზე მეტ დროს ითვლის, მისი პირველი განხორციელება გასული საუკუნის 70-იან წლებში მოხდა IBM® 7044 სერიის კომპიუტერებში Compatible Time Sharing System (CTSS), მასაჩუსეცის ტექნოლოგიურ ინსტიტუტში დამუშავებულ კომპიუტერში IBM 704 და პროექტ Atlas-ში მანჩესტერის უნივერსიტეტში (ერთ-ერთი პირველი სუპერკომპიუტერი).

IBM ჯერ კიდევ 1960-იან წლებში მიხვდა ვირტუალიზაციის საჭიროებას, კომპიუტერ System/360(™) Model 67 - ის დამუშავებისას. ამ კომპიუტერის მთელი აპარატურული ინტერფეისი ვირტუალიზირებულ იქნა VMM (the Virtual Machine Monitor)-ის საშუალებით. კომპიუტერული ეპოქის დასაწყისში ოპერაციულ სისტემას უწოდეს **„სუპერვიზორი“** (Supervisor), მას შემდეგ, რაც მოხდა ერთი ოპერაციული სისტემის რეალიზება მეორე ოპერაციულ სისტემაში, გაჩნდა ტერმინი **„ჰიპერვიზორი“** (Hypervisor).

მიღებული წარმადობის უმრავლესობა გამომუშავებულია სერვერების კონსოლიდაციის შედეგად. თუ ჩვენ გავაერთიანებთ ერთ სერვერზე რამდენიმე სისტემას, მივიღებთ სივრცის ეკონომიას, ენერგომოხმარების შემცირებას, გაგრილების სისტემის შემცირებულ რაოდენობას და სამართავად მხოლოდ ერთ სერვერს.

ვირტუალიზაცია ასევე მნიშვნელოვანია პროგრამირებისთვისაც. Linux-ის ბირთვი მუშაობს სივრცის საერთო დამისამართებით, რაც გულისხმობს იმას რომ, თუ

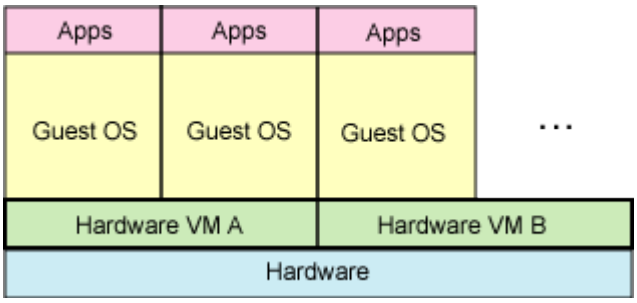
ბირთვში ან რომელიმე დრაივერში მოხდება რამე შეცდომა, მწყობრიდან გამოდის მთელი ოპერაციული სისტემა. ვირტუალიზაციის შემთხვევაში გვაქვს რამდენიმე ოპერაციული სისტემა და თუ ერთ-ერთი მათგანი გამოვა რაიმე შეცდომის გამო მწყობრიდან, ჰიპერვიზორი და დანარჩენი ოპერაციული სისტემები განაგრძობენ მუშაობას უპრობლემოდ.

1. ვირტუალიზაციის სახეები

ვირტუალიზაცია პროგრამული ტექნოლოგიაა, რომელიც სარგებლობს სერვერის ფიზიკური რესურსებით და ანაწილებს მას ვირტუალურ ნაწილებად, რომელთაც ვირტუალური სერვერები ეწოდებათ. როდესაც საქმე ვირტუალიზაციაზე მიდის, არსებობს არაერთი გზა საჭირო ეფექტის მისაღებად. ფაქტიურად არსებობს რამდენიმე ვარიანტი ერთი და იმავე შედეგის მისაღწევად, სხვადასხვა დონის აბსტრაქციის გამოყენებით. ამ ნაწილში განვიხილავთ ვირტუალიზაციის ყველაზე გავრცელებულ სახეებს Linux - ში, განვსაზღვრავთ მათ დადებით და უარყოფით მხარეებს.

1.1 აპარატურის ემულაცია

ვირტუალიზაციის ყველაზე რთულ მეთოდად ითვლება აპარატურული ემულაცია, ამ მეთოდის გამოყენებით ჰოსტ-სისტემაზე იმართება ვირტუალური მანქანა, რომელიც ახდენს სხვა რომელიმე აპარატურის არქიტექტურის მოდელირებას, როგორ ეს შემდეგ სურათზეა ნაჩვენები:



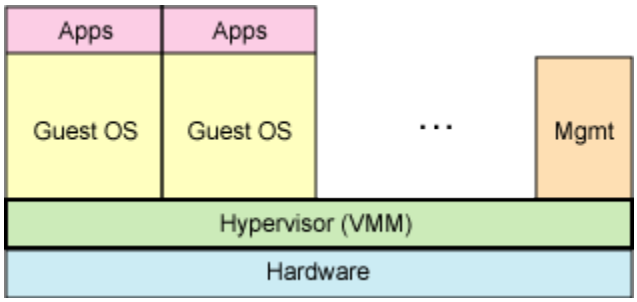
ამ მეთოდში მთავარი პრობლემა არის ის, რომ სისტემა საკმაოდ ნელია. საქმე იმაშია, რომ თითოეული ბლოკი უნდა იყოს მოდელირებული მიახლოებული რეალურ აპარატურულ უზრუნველყოფასთან, რაც იწვევს სისტემის 100-ჯერ შენელებას. სრული მოდელირებისას, რომელიც გულისხმობს სრული ციკლის შესრულებას, პროცესორის კონვეირის და ქეშ მეხსიერების სიმულაციას, შესაძლოა სისტემის სიჩქარე 1000-ჯერ მცირე იყოს, ვიდრე რეალური მოდელირებული აპარატურისა.

აპარატურის ემულაციას აქვს თავისი უპირატესობებიც. მაგალითად, ამ შემთხვევაში ჩვენ შეგვიძლია ნებისმიერი ცვლილების გარეშე ჩავტვირთოთ ოპერაციული სისტემა

განკუთვნილი PowerPC-თვის ჰოსტ-კომპიუტერზე. ჩვენ შეგვიძლია გავუშვათ რამდენიმე ვირტუალური სისტემა განსხვავებული პროცესორებით მოდელირებული.

1.2 სრული ვირტუალიზაცია

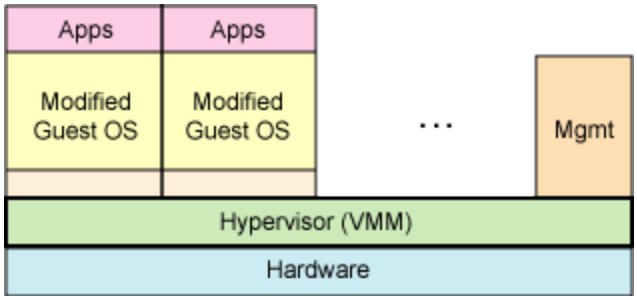
სრული ვირტუალიზაცია, რომელსაც ასევე უწოდებენ ბუნებრივს (Native Virtualization) – ეს განსხვავებული, საინტერესო ვირტუალიზაციის მეთოდია. ეს მოდელი იყენებს ვირტუალურ მანქანას, რომელიც წარმოადგენს შუამავალს ვირტუალურ სისტემასა და რეალურ მოწყობილობას შორის. ამ შემთხვევაში შუამავლის როლს ასრულებს VMM, რომელიც ამასთანავე არის ჰიპერვიზორიც:



სრული ვირტუალიზაცია შედარებით სწრაფია ვიდრე აპარატურის ემულაცია, მაგრამ წარმადობა უფრო დაბალია რეალურ აპრატურაზე. ყველაზე დიდი დადებითობა სრული ვირტუალიზაციის არის ის, რომ ოპერაციული სისტემა ვირტუალურად შეგვიძლია დავაყენოთ ისე რომ ამ სისტემაში არ გავაკეთოთ არანაირი მოდიფიკაცია. ერთადერთი შეზღუდვა არის იმაში, რომ სისტემა რომელსაც ვაინსტალირებთ, უნდა აკმაყოფილებდეს რეალური ფიზიკური მოწყობილობების სტანდარტებს.

1.3 არასრული ვირტუალიზაცია

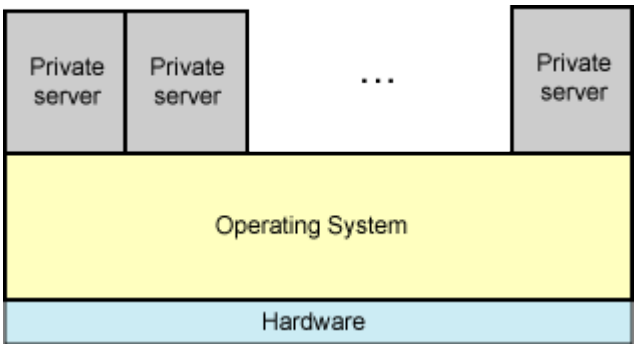
არასრული ვირტუალიზაცია - ეს არის კიდევ ერთი პოპულარული ვირტუალიზაციის სახე, რომელიც წააგავს სრულ ვირტუალიზაციას. ეს მეთოდიც იყენებს ჰიპერვიზორს განცალკევებულ მოწყობილობებთან წვდომისთვის, მაგრამ ვირტუალიზაციის კოდი ინტეგრირდება ოპერაციულ სისტემაში:



ამ შემთხვევაში ვირტუალური სისტემის წარმადობა უახლოვდება არავირტუალური სისტემის წარმადობას. სრული ვირტუალიზირებული სისტემის მსგავსად, აქაც შესაძლოა ერთდროულად წარმოდგენილი იყო რამდენიმე დამოუკიდებელი ვირტუალური სისტემა.

1.4 ვირტუალიზაცია ოპერაციული სისტემის დონეზე

ვირტუალიზაცია ოპერაციული სისტემის დონეზე წარმოადგენს ჩვენს მიერ ზემოთ განხილული ვარიანტების გაუმჯობესებულ სახეობას. ამ შემთხვევაში ვირტუალიზირდება ოპერაციულ სისტემაში ჩატვირული სერვერები. ამ შემთხვევაში ოპერაციული სისტემა არის ერთი და იზოლირებულია სერვერებისგან, რომლებიც შეიძლება იყოს რამდენიმე და ვირტუალიზირებული მოცემულ ოპერაციულ სისტემაზე:



ვირტუალიზაცია ოპერაციული სისტემის დონეზე საჭიროებს ცვლილებებს სისტემის ბირთვში, მაგრამ ეს იძლევა საშუალებას საგრძნობლად გაიზარდოს ვირტუალური სისტემის წარმადობა.

ოპერაციული სისტემის დონეზე ვირტუალიზაციის მატრიცა

მექანიზმი	ოპერაციული სისტემა	ლიცენზია	წარმოების თარიღი	ფაილური სიტემის იზოლაცია	დისკური ქვოტა	I/O-ს განსაზღვრა	მეხსიერების განსაზღვრა	CPU-ს ქვოტა	ქსელის იზოლაცია	მიგრაცია
chroot	Unix	BSD, GNU GPL, CDDL	1982	ნაწილ ობრივ	არა	არა	არა	არა	არა	არა
Containers/Zones	OpenSolaris	CDDL	05/2008	კი	კი	არა	კი	კი	კი	არა
Containers/Zones	OpenSolaris	CDDL	01/2005	კი	კი	არა	კი	კი	არა	არა
FreeVPS	Linux	GNU GPL	-	კი	კი	არა	კი	კი	კი	არა
iCore Virtual Accounts	Windows XP	Proprietary (პატენტი)	06/2008	კი	კი	არა	არა	არა	კი	არა
Linux-VServer	Linux	GNU GPL v.2	-	კი	კი	კი	კი	კი	კი	არა
LXC	Linux	GNU GPL v.2	2008	კი	არა	კი	კი	კი	კი	არა
OpenVZ	Linux	GNU GPL v.2	-	კი	კი	კი	კი	კი	კი	კი
Parallels Virtuozzo Containers	Linux Microsoft Windows	Proprietary (პატენტი)	-	კი	კი	კი	კი	კი	კი	კი
FreeBSD Jail	FreeBSD	BSD	03/2000	კი	კი	არა	კი	ნაწილ ობრივ	კი	არა
SysJail	OpenBSD, NetBSD	BSD	-	კი	არა	არა	არა	არა	კი	არა
WPARS	AIX	Proprietary (პატენტი)	10/2007	კი	კი	კი	კი	კი	კი	კი

2. ვირტუალიზაციის პროექტები Linux - თვის

შემდეგ ცხრილში მოყვანილია ვირტუალიზაციის სახეები და მათი ტიპები. აღნიშნული ჩამონათვალი არასრულია იმ სიიდან რაც არსებობს დღევანდელი მდგომარეობით კომპიუტერულ სამყაროში. განვიხილავთ აქ ჩამოთვლილი პროექტებს:

პროექტი	პროექტის ტიპი
Bochs	ემულაცია
QEMU	ემულაცია
VMware	სრული ვირტუალიზაცია
z/VM	სრული ვირტუალიზაცია
Xen	არასრული ვირტუალიზაცია
UML	არასრული ვირტუალიზაცია
Linux-VServer	ვირტუალიზაცია ოპერაციული სისტემის დონეზე
OpenVZ	ვირტუალიზაცია ოპერაციული სისტემის დონეზე

Bochs (ემულაცია) - ესაა პროგრამა იმიტატორი, მოდელირებული x86 არქიტექტურის კომპიუტერისათვის, რომელიც პორტატულია და შეიძლება გაეშვას ბევრ აპარატურულ პლატფორმაზე, როგორებიცაა x86, PowerPC, Alpha, SPARC და MIPS. მოცემული სისტემა საინტერესოა იმითაც, რომ იგი ახდენს არა მხოლოდ პროცესორის მოდელირებას, არამედ პერიფერიული მოწყობილობებისასაც, როგორებიცაა: კლავიატურა, მაუსი, ვიდეოკარტა, ქსელური კარტა და ასე შემდეგ. Bochs-ის გამოყენებით Linux-ში, ჩვენ შეგვიძლია გავუშვათ Linux-ის ნებისმიერი დისტრიბუცია, Microsoft Windows 95/98/NT/2000, Berkeley Software Distribution (BSD) - FreeBSD, OpenBSD და ა.შ.

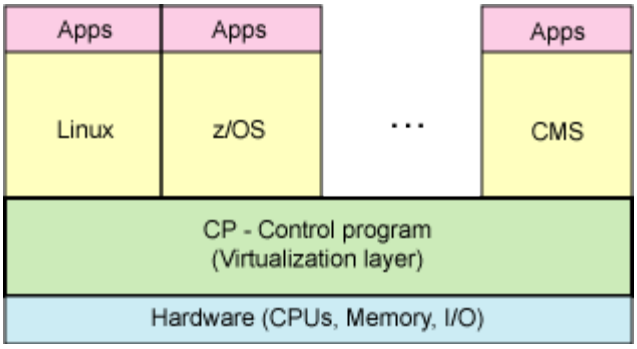
Qemu (ემულაცია) - მას აქვს ემულაციის ორი რეჟიმის მხარდაჭერა:

პირველი - სისტემის სრული ემულაცია (Full System Emulation). აღნიშნული რეჟიმის დროს ხდება პერსონალური კომპიუტერის სრული ემულაცია, პროცესორის და პერიფერიული მოწყობილობების ჩათვლით.

მეორე - მომხმარებლის რეჟიმის ემულაცია (User Mode Emulation). აღნიშნული რეჟიმი მუშაობს მხოლოდ იმ შემთხვევაში როდესაც ბაზისი ოპერაციული სისტემა არის Linux. ამ შემთხვევაში შესაძლებელია ბინარული ფაილების გაშვება, რომლებიც არის განკუთვნილი სხვა არქიტექტურისთვის. მაგალითად, x86 არქიტექტურის Linux-ზე შეიძლება გავუშვათ ბინარული ფაილი, რომელიც არის დაკომპილირებული MIPS არქიტექტურისთვის (Microprocessor without Interlocked Pipeline Stages).

VMware (სრული ვირტუალიზაცია) - მოცემულ შემთხვევაში ჰიპერვიზორი განთავსებულია ვირტუალიზირებულ ოპერაციულ სისტემასა და რეალურ მოწყობილობას შორის, ქმნის ერთგვარ დამაკავშირებელ გარემოს რეალურსა და ვირტუალურ სისტემებს შორის. აღნიშნულ აბსტრაქციას შეუძლია ამუშაოს ოპერაციული სისტემა თავის მოწყობილობებთან ერთად, იმის მიუხედავად აყენია თუ არა რომელიმე ოპერაციული სისტემა ვირტუალურად ან რამდენია ასეთი სისტემა, იგი სრულიად დამოუკიდებლად გამოყოფს რესურსს ცალკეული სისტემისთვის. მთელი ეს ვირტუალური სისტემები ინახება ფაილური სისტემით, რაც საშუალებას გვაძლევს გადავიტანოთ ნებისმიერ ადგილზე ან დავაბალანსოთ რეალურის სისტემის დატვირთვა.

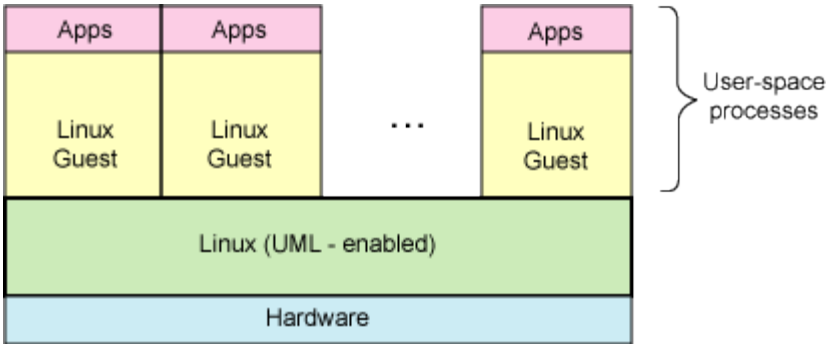
z/VM (სრული ვირტუალიზაცია) - ეს არის ჰიპერვიზორი, ოპერაციული სისტემა “System z”-თვის (System Z) მის ბირთვს წარმოადგენს მართვის პროგრამა, რომელიც უზრუნველყოფს ფიზიკური რესურსების ვირტუალიზაციას ვირტუალური სისტემისათვის, მაგალითად Linux-თვის. ამით შესაძლებელია მრავალპროცესორიანი სისტემის ვირტუალიზაცია და ასევე სხვა ფიზიკური რესურსების ვირტუალიზაცია, სხვადასხვა ვირტუალიზირებული ოპერაციული სისტემისთვის ერთდროულად:



Xen (არასრული ვირტუალიზაცია) - ეს არის თავისუფალი გადაწყვეტა ღია კოდით, კომპანია XenSource-გან. დასამახსოვრებელი ფაქტია ის, რომ არასრული ვირტუალიზაციის დროს ჰიპერვიზორი და ოპერაციული სისტემა ერთიანდებიან, ოპერაციული სისტემის ხარჯზე, მაგრამ სამაგიეროდ ვღებულობთ თითქმის ისეთ წარმადობას, როგორსაც მივიღებდით რეალური მოწყობილობების შემთხვევაში.

რადგან Xen-თვის საჭიროა ჰიპერვიზორის ისეთ ოპერაციულ სისტემასთან გაერთიანება, რომლის ცვლილებაც შესაძლებელია, ამიტომ Linux ღია კოდით არის საუკეთესო ვარიანტი, რისგანაც მივიღებთ მაღალ წარმადობას. ხოლო ისეთი ოპერაციული სისტემის შემთხვევაში, რომელშიც გარკვეული ცვლილებების შეტანა შეუძლებელია, აღნიშნული ტექნოლოგია ვერ იქნება წარმატებული.

UML – User Mode Linux (არასრული ვირტუალიზაცია) - აღნიშნული მეთოდი გვაძლევს საშუალებას ოპერაციული სისტემა Linux-ის ერთი მომხმარებლის ჭრილში გაშვებული გვქონდეს სხვა ოპერაციული სისტემა Linux. ყოველი ვირტუალური სისტემა გაშვებულია ერთ საბაზისო ოპერაციულ სისტემაზე. ვიღებთ სისტემის ორ განსხვავებულ ბირთვს, საბაზისო ოპერაციული სისტემის ბირთვს და მომხმარებლის სივრცეში რეალიზებული ვირტუალური ოპერაციული სისტემის ბირთვს:



Linux – Vserver (ვირტუალიზაცია ოპერაციული სისტემის დონეზე) - ცვლის Linux-ის ბირთვს ისეთნაირად, რომ იქმნება მომხმარებელთა სივრცის სიმრავლე, ვირტუალური სერვერის დამოუკიდებელი ნაწილებით (Virtual Private Servers - VPS). Linux-VServer უზრუნველყოფს ამ მომხმარებელთა სივრცის იზოლაციას Linux-ის ბირთვის ხარჯზე.

Linux-VServer ასევე იყენებს სპეციალური ფორმა უტილიტას Chroot, რაც უზრუნველყოფს ფუძე დირექტორიის იზოლაციას ყოველი VPS-თვის. Croot იძლევა საშუალებას განვსაზღვროთ ახალი ფუძე კატალოგი, მაგრამ ამ შემთხვევაში საჭიროა დამატებითი ფუნქციონალი, რომელსაც უწოდებენ Croot-Barrier, რომ VPS -ს არ შეეძლოს გამოსვლა მისთვის დანიშნული ფუძე კატალოგიდან. გარდა იზოლირებული ფუძე კატალოგისა, ყოველ VPS-ს აქვს საკუთარ მომხმარებელთა სია და ცალკე პაროლი სუპერმომხმარებლისთვის Root.

OpenVZ (ვირტუალიზაცია ოპერაციული სისტემის დონეზე) - ეს არის ოპერაციული სისტემის დონეზე ვირტუალიზაციის მეორე ვარიანტი, გავს Linux-Vserver-ს, მაგრამ აქვს საინტერესო განსხვავებებიც. იგი მორგებულია მოდიფიცირებული ბირთვის ვირტუალიზაციისთვის, რომელიც მხარს უჭერს მომხმარებელთა სივრცის იზოლირებას, აგრეთვე აქვს მომხმარებელთა უტილიტების ნაბორი, სამართავად. მაგალითად, შესაძლებელია მარტივად შევქმნათ ახალი VPS ბრძანებათა ველიდან:

```
$ vzctl create 42 --ostemplate fedora-core-4
Creating VPS private area
VPS private area was created
$ vzctl start 42
Starting VPS ...
VPS is mounted
```

ასევე შესაძლებელია მივიღოთ ინფორმაცია უკვე არსებული VPS-ების შესახებ, შემდეგი ბრძანების გამოყენებით vzlist. პროცესების სამართავად OpenVZ-ში არსებობს დავალებების ორდონიანი დამგეგმავი (დისპეჩერი) CPU, ეს დისპეჩერი პირველ რიგში განსაზღვრავს, რომელმა VPS-მა უნდა მიიღოს ნებართვა CPU-ზე, როცა ეს უკვე განსაზღვრულია, მეორე დონის დამგეგმავი უშვებს პროცესს შესრულებაზე. მისთვის დაწესებული Linux-ის პრიორიტეტების გათვალისწინებით.

OpenVZ-ის უნიკალური შესაძლებლობაა ისიც, რომ საშუალებას იძლევა შევქმანათ საკონტროლო წერტილები (To Checkpoints) და გადავიტანოთ VPS ერთი ფიზიკური სერვერიდან მეორეზე.

ზემოთ ჩამოთვლილის გარდა, არსებობს კიდევ ვირტუალიზაციის სხვადასხვა მექანიზმები, მაგალითად ერთ-ერთი მათგანი არის VirtualBox, რომელსაც დაწვრილებით განვიხილავთ შემდეგ თავში და განვახორციელებთ მისი მეშვეობით ვირტუალურად Windows-ის ინსტალაციას Linux-ზე.

3. Window-ის ვირტუალიზაცია Linux-ის გარემოში

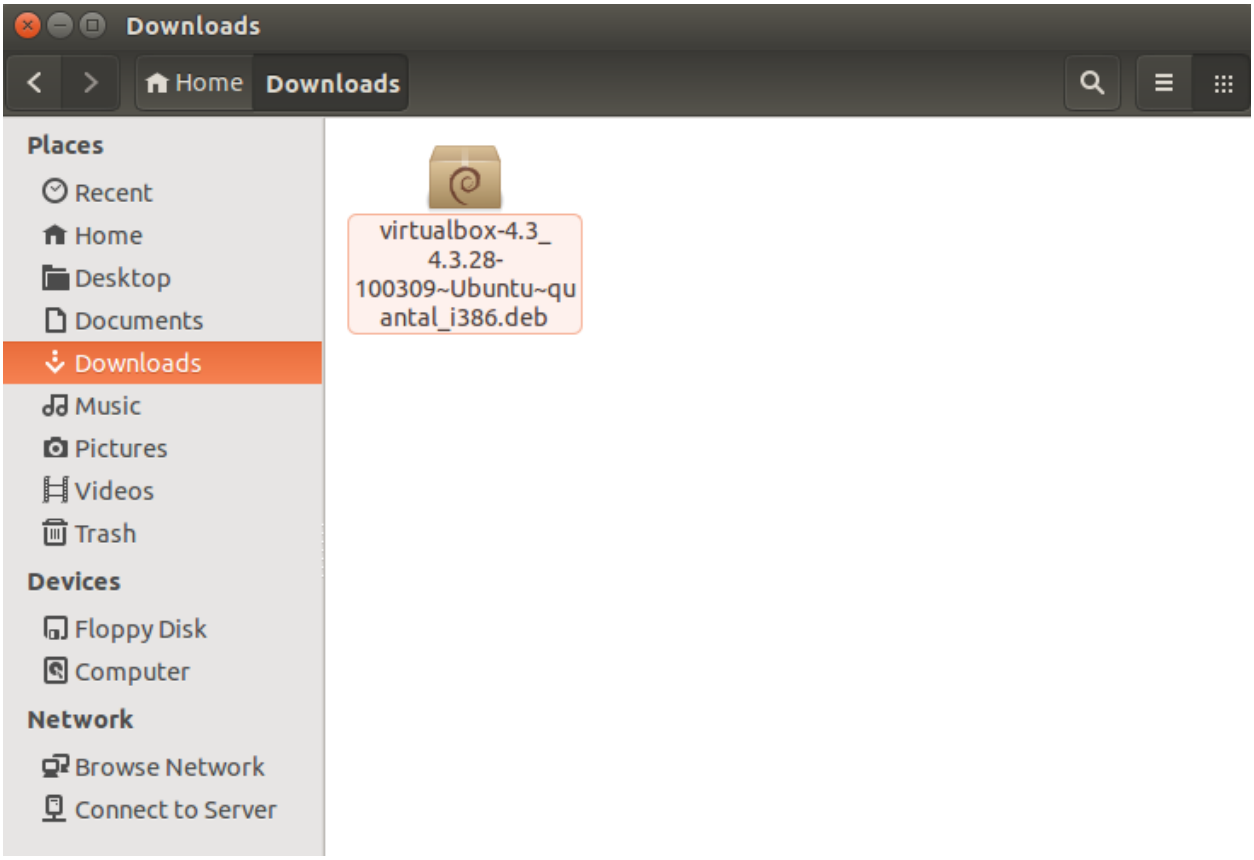
ამ თავში განვიხილავთ Linux-ის გარემოში Windows-ის ვირტუალიზაციის პრაქტიკულ მაგალითს, VirtualBox-ის გამოყენებით.

VirtualBox განეკუთვნება სრული ვირტუალიზაციის სისტემების კატეგორიას, მაგალითად როგორიცაა VMware. იგი შეიძლება გაიმართოს სხვადასხვა ოპერაციულ სისტემებზე, როგორებიცაა Windows, Linux, Mac OS X. ცალკეული მომხმარებლებისთვის პროგრამა არის უფასო, თუმცა ასევე არის კომერციული ვერსიებიც, რომელიც იყენებს სტანდარტულ საერთო ლიცენზიას - General Public License (GPL).

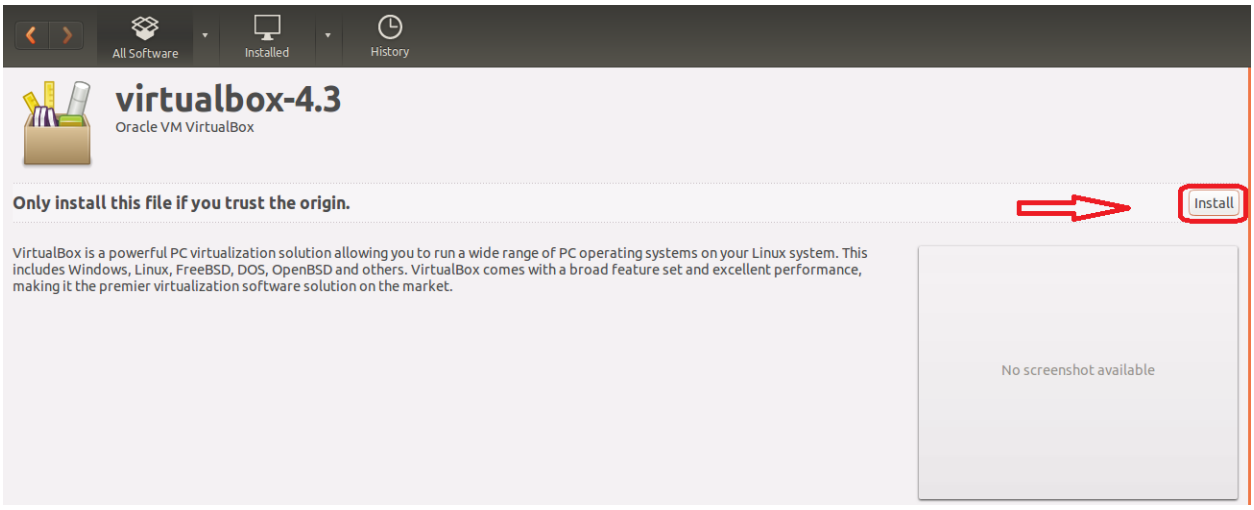
მოცემული პროგრამის ჩამოტვირთვა შესაძლებელია მისი ოფიციალური საიტიდან https://www.virtualbox.org/. მინიმალური სისტემური მოთხოვნები და რეკომენდაციები:

- x86 არქიტექტურის კომპიუტერი, ნებისმიერი Intel ან AMD ტიპის პროცესორი.
- მეხსიერება. პროგრამაში რომ გაეშვას ვირტუალური ოპერაციული სისტემა, ამისათვის საჭიროა მინიმუმ 512 MB RAM. ჩვენს შემთხვევაში რეკომედირებულია მინიმუმ 1 GB.
- მყარი დისკის მოცულობა. ვინაიდან VirtualBox-ს ინსტალაციისთვის ესაჭიროება ძალიან ცოტა მეხსიერება დაახლოებით 30 MB, ამიტომ ვირტუალური სისტემის ინსტალაციისთვისაც არ ითხოვს განსაკუთრებულ დიდ მეხსიერებას, ამისათვის რამდენიმე GB-იც საკმარისია. ინსტალაციის პროცესში სტანდარტული რეკომენდირებული მონაცემები შერჩეულია ავტომატურად, რისი ცვლილებაც ნებისმიერ დროს არის შესაძლებელი.
- VirtualBox-ს აქვს შემდეგი ოპერაციული სისტემების მხარდაჭერა, რომლებზეც შესაძლებელია მისი რეალიზება: Windows XP და Windows-ის შემდეგი ვერსიები, Linux-ის რამდენიმე სახეობა, Mac OS X, Solaris და OpenSolaris.
- რაც შეეხება იმ ოპერაციულ სისტემებს თუ რომლებიც შეიძლება ვირტუალურად დავაინსტალიროთ VirtualBox-ის მეშვეობით, მათი სრული ჩამონათვალი შესაძლებელია ვიხილოთ შემდეგ მისამართზე: https://www.virtualbox.org/wiki/Guest_OSes.

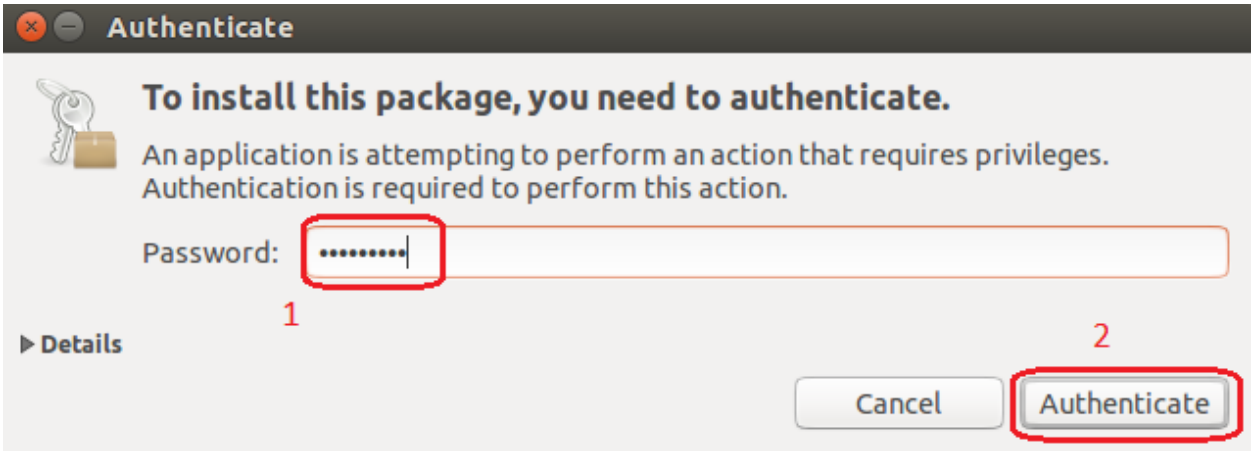
დავიწყოთ ინსტალაციის პროცესი. მას შემდეგ რაც გადმოვწერთ საინსტალაციო ფაილს, ვუშვებთ ინსტალაციის დასაწყებად, აღნიშნული საინსტალაციო ფაილი დაგვხვდება Downloads საქაღალდეში:



ამის შემდეგ გამოდის ინსტალაციის ფანჯარა, სადაც ვაჭერთ ღილაკს Install:

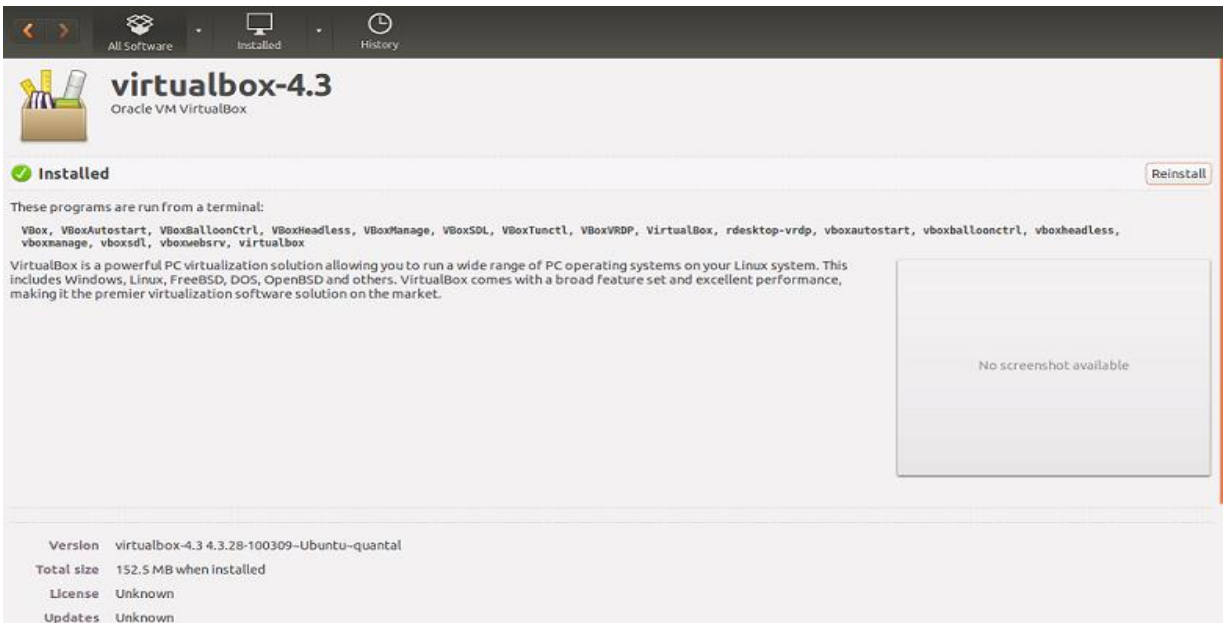


რადგან მოცემული ოპერაციული სისტემა ითხოვს აუთენთიფიკაციას პროგრამული უზრუნველყოფის მომენტში, შემდეგ გამოსულ ფანჯარაში შეგვყავს Linux-ის იუზერის პაროლი და ვაჭერთ ღილაკს Authenticate:

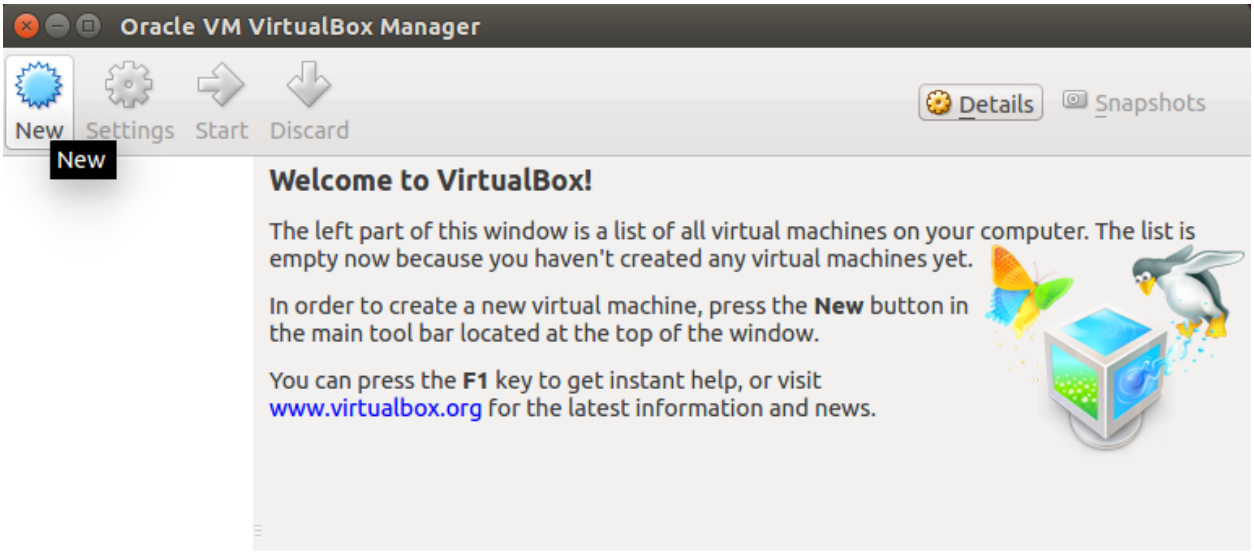


შემდეგ გამოსულ ფანჯარაში მიდის ინსტალაციის პროცესი და ველოდებით:

ინსტალაციის დასრულების შემდეგ გამოდის შეტყობინება, რომ პროგრამა უკვე დაინსტალირებულია, აქვე არის ღილაკი Reinstall, რომლის გამოყენებაც შეგვიძლია, თუ რაიმე არასწორად გავაკეთეთ ინსტალაციის პროცესში, ცვლილების შესატანად:

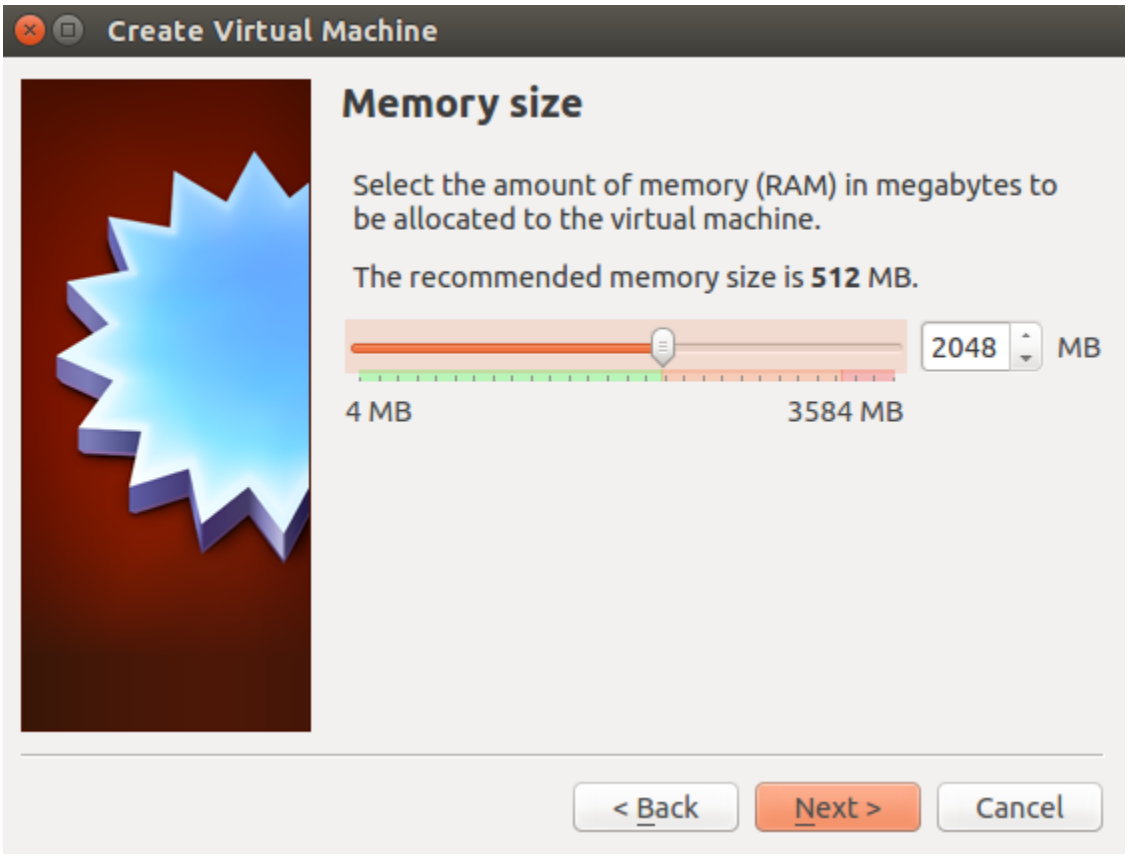


VirtualBox უკვე მზადაა ვირტუალური სისტემის საინსტალაციოდ, რომელ პროცესსაც ვიწყებთ New ღილაკით:



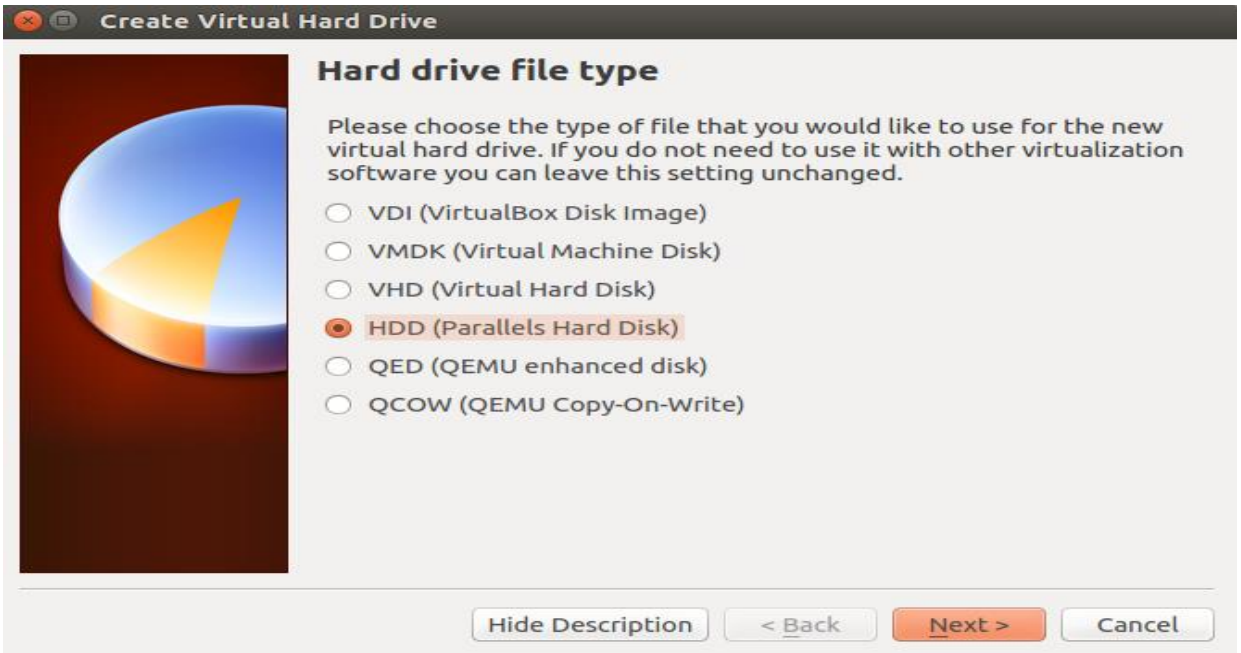
შემდეგ მოცემულ ბიჯზე, ჩამონათვალში ვირჩევთ თუ რომელ ოპერაციულ სისტემას ვაინსტალირებთ ვირტუალურად, ვარქმევთ სახელს და Next ღილაკით გადავდივართ შემდეგზე:

შემდეგში ხდება ოპრაციული სისტემის კონფიგურირება, მისი ზომის განსაზღვრა. ამ სქრინზე ჩანს რომ შესაძლებელია გავზარდოთ ან შევამციროთ ოპერატიული მეხსიერების მოცულობა, მაგრამ ამასთანავე მოცემული სტანდარტული რეკომენდირებული მოცულობა 512 MB. ამ შემთხვევაში ვირჩევთ 2 GB, რადგან გვაქვს ამის საშუალება და ვაგრძელებთ:

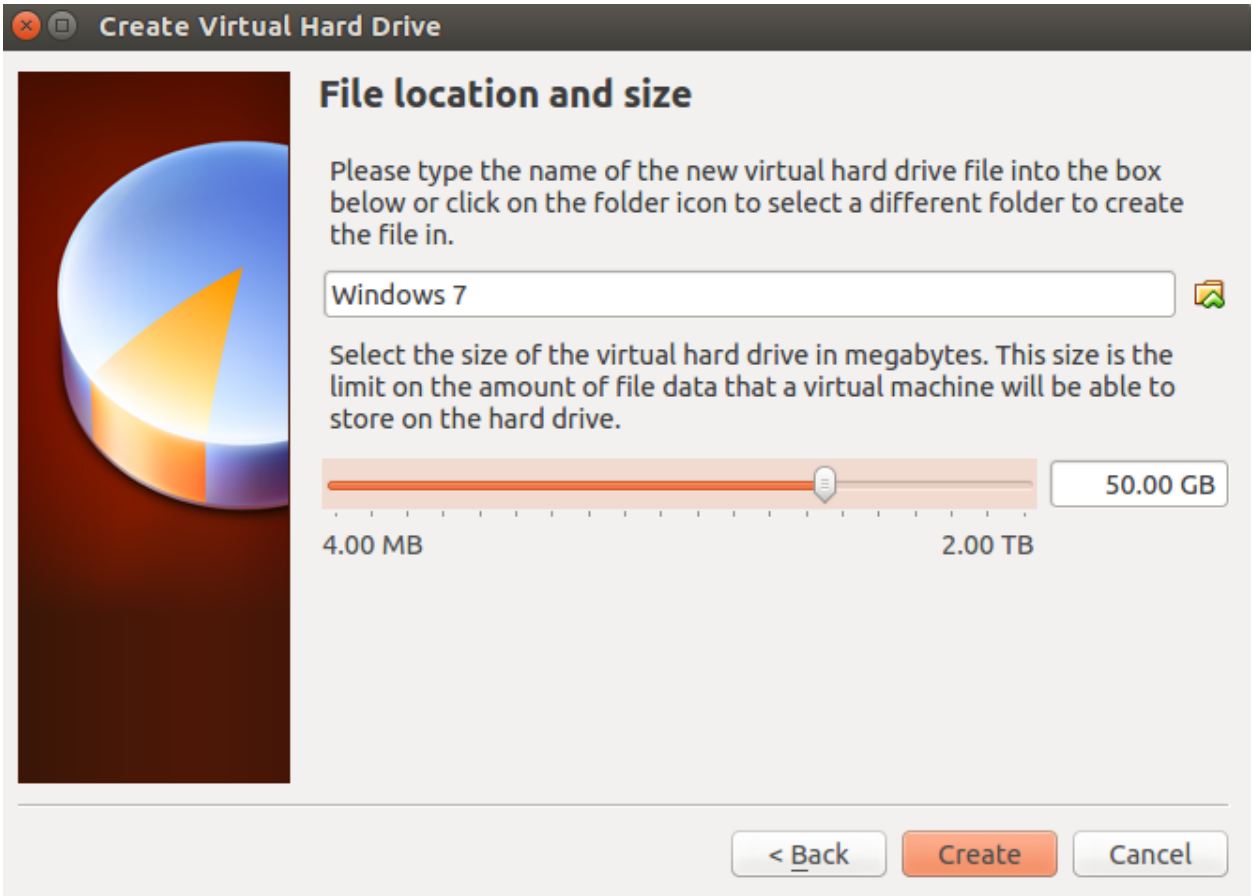


შემდეგ ფანჯარაში ვაკონფიგურირებთ ვირტუალური მყარი დისკის ზომას, აქაც მოცემულია სტანდარტული რეკომენდაცია (25 GB) მოცემული ოპერაციული სისტემისათვის (Windows 7 32 Bit), რომელსაც ვაინსტალირებთ. ვირჩევთ „Create a virtual hard drive now“ (მყარი დისკის შექმნა) და ვაგრძელებთ Create ღილაკით:

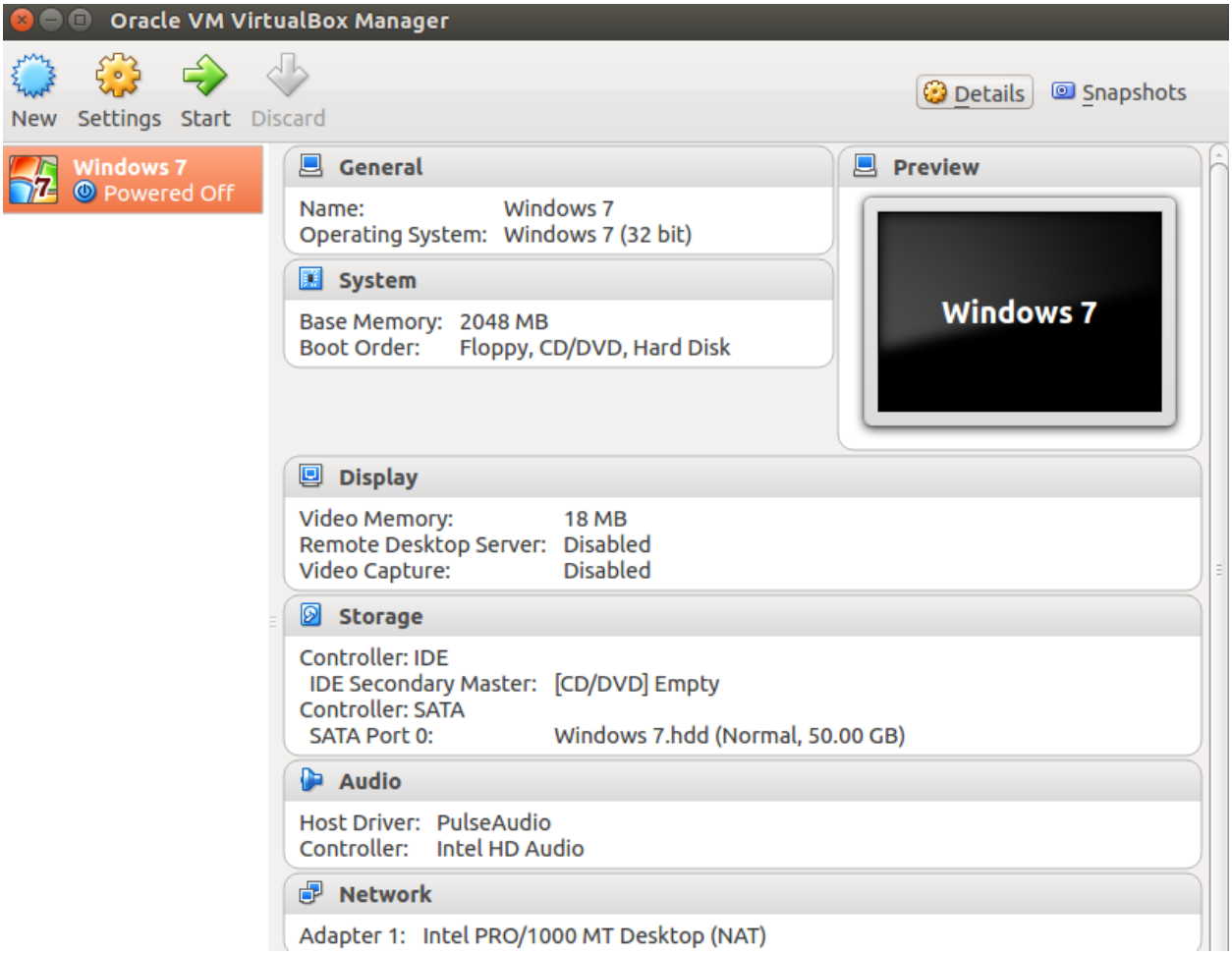
შემდეგ ფანჯარაში ვირჩევთ მყარი დისკის ტიპს ვირტუალიზაციისთვის, მოცემული ჩამონათვალიდან შეგვიძლია ავირჩიოთ ნებისმიერი, იმის მიხედვით თუ რისთვის გვჭირდება:



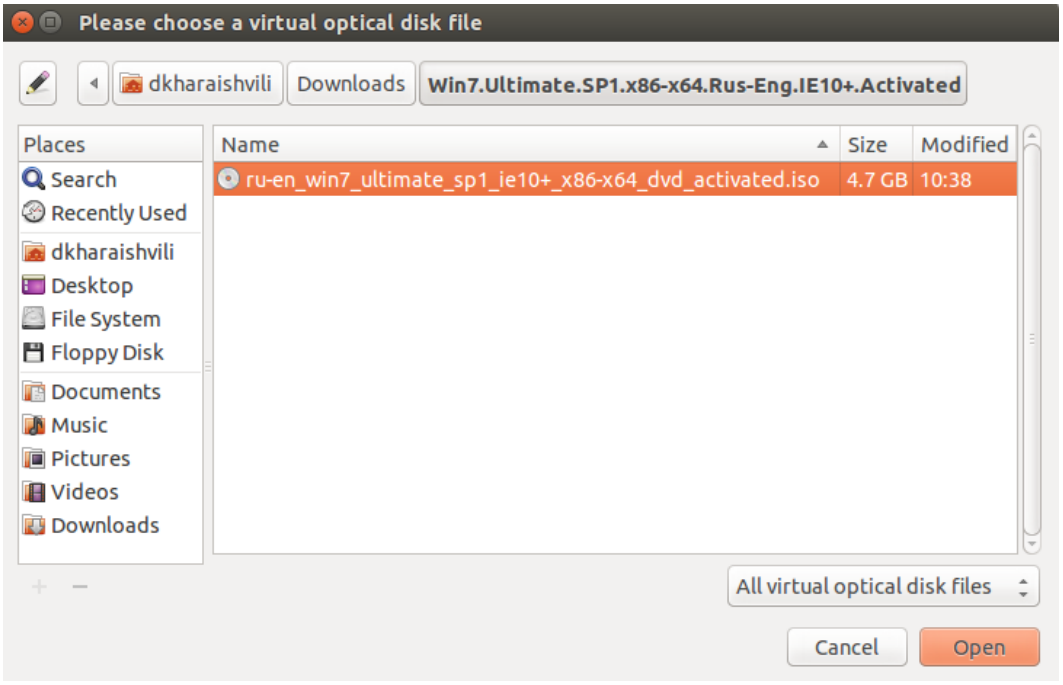
შემდეგ ბიჯზე ვირჩევთ მყარი დისკის ზომას, ვირტუალური სისტემის სახელს და მისამართს, თუ სად უნდა განთავსდეს ფაილურად აღნიშნული სისტემა და Create ღილაკით ვაგრძელებთ პროცესს:



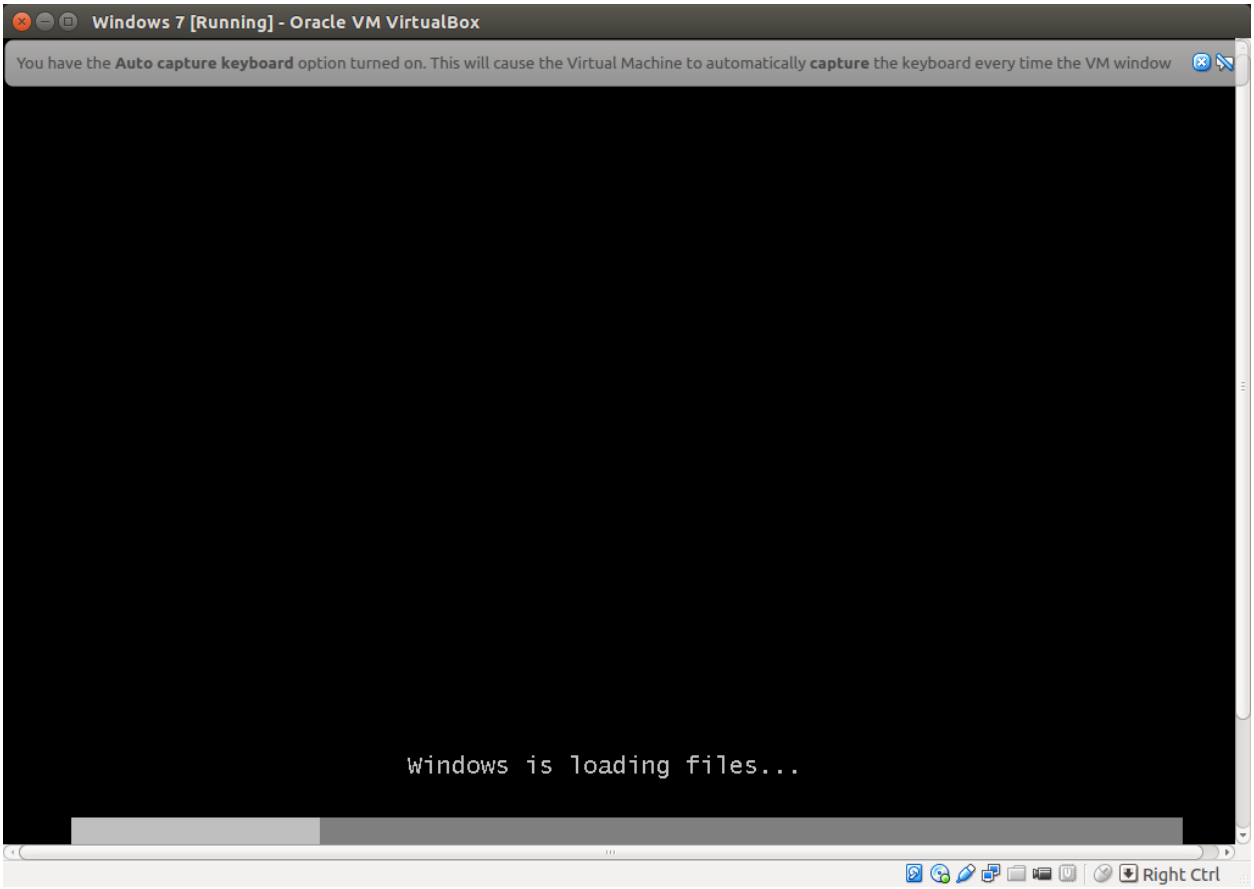
ამის შემდეგ პლატფორმა დაკონფიგურირებულია და მზადაა მასზე ვირტუალური სისტემის დასაყენებლად. მოცემულ ფანჯარაში ჩანს ინფორმაცია თუ როგორაა კონფიგურირებული VirtualBox. Start ღილაკით ხდება უკვე არჩეული სისტემის ინსტალირებისთვის საჭირო ფანჯარაში გადასვლა:



მომდევნო სქრინზე ვირჩევთ .iso ტიპის ფაილს, რომელიც წარმოადგენს ოპერაციული სისტემის „Image“-ს, ვხსნით მას და Start ღილაკით ვიწყებთ, ჩვენს შემთხვევაში Windows - ის ინსტალაციას. თუმცა შესაძლებელია დაინსტალირდეს ნებისმიერი ოპერაციული სისტემა, რომლის მხარდაჭერაც აქვს VirtualBox - ს:


Please choose a virtual optical disk file
dkharaishvili
Downloads
Win7.Ultimate.SP1.x86-x64.Rus-Eng.IE10+.Activated
Places
Search
Recently Used
dkharaishvili
Desktop
File System
Floppy Disk
Documents
Music
Pictures
Videos
Downloads
Name
Size
Modified
ru-en_win7_ultimate_sp1_ie10+_x86-x64_dvd_activated.iso
4.7 GB
10:38
All virtual optical disk files
Cancel
Open


Select start-up disk
Please select a virtual optical disk file or a physical optical drive containing a disk to start your new virtual machine from.
The disk should be suitable for starting a computer from and should contain the operating system you wish to install on the virtual machine if you want to do that now. The disk will be ejected from the virtual drive automatically next time you switch the virtual machine off, but you can also do this yourself if needed using the Devices menu.
ru-en_win7_ultimate_sp1_ie10+_x86-x64_dvd_a
< Back
Start
Cancel



Windows - ის ინსტალაციის შემდეგ, ვირტუალური სისტემა მზადაა სამუშაოდ, რომელიც წარმოადგენს, VirtualBox - ის საშუალებით გაწყობილ, სრულად ვირტუალიზირებულ სისტემას.

დასკვნა

დღევანდელ ტექნოლოგიურ სამყაროში, ვირტუალიზაცია წინ გადადგმული დიდი ნაბიჯია, მისი დადებითი თვისებებიდან გამომდინარე. როგორც უკვე ავღნიშნეთ იგი ფართოდ გამოიყენება კომპიუტერულ ტექნოლოგიებში და საკმაო პოპულარობითაც სარგებლობს. ძირითადი პრობლემა რასაც ორგანიზაციები აწყდებიან ვირტუალურ გარემოზე გადასვლისას არის შემდეგი:

1. ორგანიზაციების 80% რომლებიც ნერგავენ ვირტუალიზაციას საქართველოში სასერვეროს მოცულობა იზრდება 20-30%-ით. ამის ახსნა მარტივია, სერვერის შექმნა ვირტუალურ გარემოში იმდენად მარტივია, რომ სისტემურ ადმინისტრატორს შეუძლია რამდენიმე კლიკით შექმნას სერვერი ვირტუალურ გარემოში, ვიდრე რეალურ სერვერზე დაამატოს შესაბამისი საჭირო აპლიკაცია, რამაც შესაძლოა გარკვეული დროით, სერვისების შეფერხება გამოიწვიოს.

2. რეალური და გაანგარიშებული სიმძლავრეების არ დამთხვევა. ორგანიზაციების საკმაოდ დიდი ნაწილი აწყდება ამ პრობლემას. ეს ნაწილობრივ პირველი პუნქტითაც არის გამოწვეული, 30%-ით დიდ ინფრასტრუქტურას მინიმუმ 15-25%-ით მეტი რესურსი ჭირდება და გარდა ამისა, ხშირად გათვლა ხდება პროცესორული რესურსზე და ოპერატიულ მეხსიერებაზე, დისკური სისტემები კი რატომღაც ავიწყდებათ, რაც ქმნის შემდგომ ძალიან დიდ პრობლემებს და ხანდახან ორგანიზაცია სერიოზულად ფიქრობს ძველ ინფრასტრუქტურაზე დაბრუნებას ან უწევს დამატებით გაუთვალისწინებელი ხარჯის გაღება. რეალურად, ვირტუალიზაციის დანერგვისას ყველაზე დიდი მნიშვნელობა დისკურ სისტემებს აქვს, მარტივი მაგალითისთვის: სტანდარტული დისკური სისტემები (RAID მასივი, კარგი კონტროლერით და საკმარისი ქეშ-მეხსიერებით) სრულიად საკმარისია, რომ გარკვეული აპლიკაცია ამუშაოს, მაგრამ როცა ამ სერვერზე 10 ან მეტი ანალოგური აპლიკაციის გაშვება ხდება, დისკური სისტემა ყველაზე სუსტი რგოლი გამოდის, რადგან თუ პროცესორის და ოპერატიული მეხსიერების დათვლა ძალიან მარტივია, დისკური სისტემის საჭირო სისტემის გამოთვლა საკმაოდ რთული პროცესია.

ვირტუალიზაციას მთელი რიგი უპირატესობები გააჩნია, ეს ძირითადი უპირატესობებია:

1. პროგრამული ჩავარდნების შემცირება და ასევე ერთმანეთთან კონფლიქტური პროგრამების ერთმანეთისგან განცალკევება.
2. ერთმანეთთან მჭიდროდ დაკავშირებული პროგრამების ერთ ჯგუფში გაწევრიანება და ერთ ვირტუალურ მანქანაზე ატვირთვა.
3. უსაფრთხოება
    3.1. ადმინისტრაციული უფლებების განაწილება იძლევა საშუალებას შევზღუდოთ და ვმართოთ ყოველი იუზერის უფლება.
    3.2. შიდა და გარე უსაფრთხოების ამაღლება - დაცვა შიდა და გარე კიბერ შემოტევისგან.
4. რესურსების გადანაწილება - ყოველ ვირტუალურ მანქანაზე გამოიყოფა ის რესურსები, რაც რეალურადაა საჭირო.
    4.1. მანქანებს შორის მოქნილი ქსელური კავშირი
    4.2. დასმული ამოცანებიდან პრიორიტეტების ამორჩევა
    4.3. მყარი დისკის მოდულების ოპტიმალური გადანაწილება
    4.4. მეხსიერების მიზანმიმართული გამოყენება
5. მუდმივი ხელმისაწვდომობა
    5.1. კრიტიკული სერვერების თანმიმდევრული განახლება
    5.2. ვირტუალური მანქანების Online მიგრაციის შესაძლებლობა
6. ადმინისტრირების სიმარტივე და მისი ხარისხის ამაღლება
    6.1. ექსპერიმენტებისა და გამოკვლევების ჩატარების შესაძლებლობა
    6.2. რეგრესიული ტესტების გაკეთების შესაძლებლობა.

გამოყენებული რესურსი

1. РЕКОМЕНДАЦИИ ПО ПРАКТИЧЕСКОМУ ИСПОЛЬЗОВАНИЮ ПРОГРАММЫ VirtualBox ДЛЯ ОПЕРАЦИОННЫХ СИСТЕМ Microsoft Windows, Linux, FreeBSD, Mac OS X, Solaris/OpenSolaris, ReactOS, DOS: http://www.tstu.ru/book/elib2/pdf/2014/nemtinov.pdf

2. Виртуализация операционных систем и приложений: http://www.pcweek.ru/infrastructure/article/detail.php?ID=107230

3. Виртуализация на уровне операционной системы: http://dic.academic.ru/dic.nsf/ruwiki/1449760; http://www.osp.ru/os/2002/01/180946/

4. Какие программы для виртуализации использовать в Linux: http://linuxsetup.ru/kakie-programmy-dlya-virtualizacii-ispolzovat-v-linux/

5. Виртуальный Linux: http://rus-linux.net/kos.php?name=/papers/virtual/virtual-linux.koi