



Operativni Sistemi

II Semestar – 2022/23 – Vježbe

Sedmica 2

Handout za Vježbe

Agenda:

- System Clean Install
 - Načini Clean install operacije (Windows)
- Pravljenje bootable USB
 - Dualboot
 - Multiboot
- Virtualizacija
- Dockerizacija

Kontakt:

Narcisa.hadzajlic@size.ba (A Grupa)

Adin.jahic2019@size.ba (B Grupa)

Rekapitulacija operativnih Sistema.

Glavni dio operativnih sistema jeste njihovo jezgro ("KERNEL") koji predstavlja dio operativnog sistema koji se bavi apstrakcijom kompleksnih hardverskih implementacija koje bi naš život kao softverskom inženjeru dosta otežalo za razvoj aplikacija i kompleksnih sistema. Mi nikada ne komuniciramo direktno sa KERNELOM. Komunikacija se vrši preko takozvanih sistemskih poziva I trapova.

Trap

Izraz "**trapovi**" u kontekstu operativnih sistema obično se odnosi na **posebne mehanizme ugrađene u sistem kako bi se prekinuo izvršavanje programa u određenim situacijama**, uključujući greške u programu, pokušaje pristupa zabranjenim resursima ili pozive sistema.

Trapovi, takođe poznati kao prekidi ili izuzeci, su mehanizmi koje operativni sistemi koriste kako bi se prekinulo normalno izvršavanje programa u određenim situacijama. U osnovi, trapovi su posebni događaji koji se generišu kada se dogodi nešto neočekivano ili neželjeno u programu, kao što je pokušaj pristupa nedozvoljenom dijelu memorije ili pokušaj izvršavanja nedozvoljenog sistema. Kada se dogodi trap, procesor prebacuje kontrolu u kernel modu, gdje operativni sistem može preuzeti kontrolu nad situacijom i reagovati na nju.

Ovi prekidi se razlikuju po prioritetu i namjeni, a operativni sistem obično ima posebnu rutinu za obradu svakog prekida. Ova rutina obično ima pristup privilegovanim nivoima sistema i može izvršiti određene akcije kako bi rešila situaciju.

Neki od najčešćih **tipova trapova** uključuju:

- Prekid lošeg koda (eng. segmentation fault) - ovo se događa kada program pokuša pristupiti memoriji koja nije dodijeljena ili koja nije dostupna za čitanje i pisanje
- Prekid dijeljenja nulom (eng. divide-by-zero exception) - ovo se događa kada se pokuša podijeliti nešto sa nulom.
- Prekid pristupa zabranjenom resursu (eng. access violation) - ovo se događa kada program pokuša pristupiti dijelu memorije ili resursu koji mu nije dozvoljen.
- Prekid hardverske greške (eng. hardware exception) - ovo se događa kada dođe do problema sa hardverom, kao što je greška u radu sa memorijom ili drugim uređajim

Ukratko trapovi su ključni za stabilnost i sigurnost operativnih sistema, jer omogućavaju operativnom sistemu da preuzme kontrolu nad situacijama koje bi inače mogle dovesti do kraha sistema ili ugroziti ga.

Sistemske Pozive

Sistemske pozive su način na koji aplikacije komuniciraju sa operativnim sistemom. Kada aplikacija želi pristupiti resursima kao što su datoteke, mrežni soketi ili ulaz/izlaz, mora uputiti zahtjev operativnom sistemu. Ovo se postiže pozivanjem funkcija koje su dio operativnog sistema i koje se nazivaju sistemski pozivi.

Sistemske pozive su definisani kao **funkcije koje omogućavaju korisničkim aplikacijama da komuniciraju sa jezgrom operativnog sistema**. Kada se sistemski poziv pozove, procesor prebacuje kontrolu u kernel modu, gdje operativni sistem može obaviti traženu operaciju i vratiti kontrolu aplikaciji. Sistemske pozive su obično implementirani kao bibliotečke funkcije koje se nalaze u sistemskim bibliotekama operativnog sistema, a korisničke aplikacije ih pozivaju preko API-ja (Application Programming Interface).

Neki od najčešćih sistemskih poziva uključuju:

- otvaranje, čitanje i pisanje datoteka
- upravljanje procesima i nitima
- upravljanje mrežnim soketima
- upravljanje uređajima kao što su štampači i monitori
- upravljanje memorijom
- upravljanje vremenom i datumom

Sistemi pozivi su ključni za interakciju između korisničkih aplikacija i operativnog sistema. Bez njih, aplikacije bi morale da implementiraju sopstvene mehanizme za pristup resursima, što bi bilo veoma teško i neefikasno. Sistemi pozivi omogućavaju aplikacijama da koriste resurse operativnog sistema na standardizovan način, što čini razvoj aplikacija lakšim i efikasnijim.

System Clean Install

CLEAN INSTALL je instalacija softvera koja podrazumijeva uklanjanje svake prethodne verzije. Alternativa je nadogradnja (**UPGRADE**), u kojoj ostaju elementi prethodne verzije. Svaki put kada se OS instalira na novom računaru ili se aplikacija instalira po prvi put, instalacija se smatra "čistom" instalacijom.

Clean install uklanja postojeće OS i korisničke fajlove, lične podatke, aplikacije, drajvere i sve promjene u postavkama te formatira hard disk i briše u potpunosti. Stoga, korisnik bi trebao unaprijed napraviti sigurnosnu kopiju svih podataka i datoteka na eksternom tvrdom disku, drugom računaru ili na lokaciji na mreži/oblaku ili uraditi image recovery stabilne verzije sistema.

UPGRADE se također naziva instalacija na mjestu gdje se zadržava postojeća verzija OS-a ili aplikacije sa svim podacima, datotekama i postavkama, ali se ažurira dodavanjem novih elemenata.

Ako se vrši **CLEAN INSTALL** za noviju verziju postojećeg sistema na računaru, onda se radi o **CLEAN UPGRADE**.

Obično „installation wizard“ daje izbor na početku procesa instalacije da se odabere ili standardna nadogradnja (podrazumevana opcija) ili čista instalacija.

Clean install metode

- **Clean install preko USB (Windows 10)**

Najbolji način da je kreiranje **BOOTABLE USB** pomoću alata za kreiranje medija, komandne linije ili alata treće strane kao što je *Rufus*. Potrebno je osigurati da se računar može pokrenuti sa USB-a prilagođavanjem postavki u BIOS-u, te pokretanje **boot-loadera (program koji pokreće OS kada se upali računar)** zavisi od proizvođača matične ploče, a tu su funkcijske tipke na tastaturi (npr. f6 ili f10)

Za početak treba pokrenuti računar sa Windows 10 USB fleš diskom. Nakon nastavka se pojavljuju opcije za instalaciju. Ukoliko se vrši ponovna instalacija bira se "I do not have product activation key". U tom slučaju, ponovna aktivacija bi se dogodila automatski.

Ako imate ključ, treba odabrati izdanje "Windows 10" licencni ključ aktivira. Za čistu instalaciju treba izabrati opciju **CUSTOM**.

Time se nudi mogućnost odabira particije na hard disku na koju želite instalirati Windows 10 i kliknite na dugme **DELETE**. Obično "Drive 0" je disk koji sadrži sve instalacione datoteke.

NAPOMENA: *Brisanjem particije brišu se i svi podaci na disku. Nije potrebno brisati particije sa sekundarnog hard diska.*

Odaberite hard disk (Disk 0 je obično kao neraspoređeni prostor) da biste instalirali Windows 10.

Nakon instalacije se pojavljuju opcije za dalje prilagođavanje novoinstaliranog sistema kao što je biranje regiona, tastature, konfiguracija izgleda i mreže (system se povezuje/pronalazi automatski Ethernet mrežu).

Clean install Windows 10 from Media Creation Tool

Alat za kreiranje medija je dostupna za uređaje koji već koriste Windows 10, Windows 8.1 ili Windows 7 i pruža opciju za vraćanje na prethodnu instalaciju. Na web stranici Microsoft podrške u odeljku „Kreirajte instalacijski medij za Windows 10“ ([Download Windows 10 \(microsoft.com\)](https://www.microsoft.com/en-us/software-download/windows10)) kliknite na dugme „Download

now". Dvoklikom na datoteku MediaCreationToolXXXX.exe se pokreće alat. Odaberite opciju "Upgrade this PC now".

Nakon toga treba izabrati opciju "Change what to keep".

Ostatak proces instalacije je identičan (konfiguracija novoinstaliranog Sistema).

Postoje i sljedeći načini (za samostalnu aktivnost):

- **Clean install Windows 10 from Reset this PC (local)**
- **Clean install Windows 10 from Reset this PC (cloud)**
- **Clean install Windows 10 from WinRE**
- **Clean install Windows 10 from ISO file**

Clean Install	
Pros	Cons
Fixes most performance-related problems.	Software licenses need to be manually accounted for.
Clears spyware and registry file.	Setting up the new operating system will take longer.
Faster startup due to lack of startup programs and applications.	Need to manually backup documents, applications, and settings in old OS.

Upgrading	
Pros	Cons
Less setup time required for the new operating system.	Old performance-sapping applications and registry errors will remain.
Easier setup with fewer steps.	Slower startup due to existing startup programs and applications.
Preserves applications, data, and settings.	

Pravljenje bootable USB

Potrebni alati:

- USB Drive te bilo kakav externi storage drive ili disk koji se može pokreniti prilikom bootanja.
- **ISO** Fajl željenog operativnog sistema.
- Program za pravljenje Bootabilnih uređaja (Rufus, Yumi, Ventroy...)

Ako u trenutku nemate USB, odabirom **ISO** datoteke možete da je sačuvate na računaru kako biste je mogli snimiti na DVD ili USB. Iz ponuđene liste odaberite svoj USB (ako ga nema kliknite Refresh list opciju). U zavisnosti od brzine interneta, alat će preuzeti sve što je potrebno i snimiti na USB. U toku preuzimanja se računar može koristiti bez smetnji.

Bootable USB je spreman. U budućnosti ako ikada budete imali potrebu za clean instalacijom jednostavno možete koristiti kreirani USB drive na vašem računaru i pokrenuti ga. Ako se računar ne pokrene preko USB-a, onda treba preko BIOS-a promijeniti disk za pokretanje računara na USB disk.

NAPOMENA: kreiranje medija za pokretanje formatira USB te je potrebno izabrati fajl sistem NTFS, a ne FAT32.

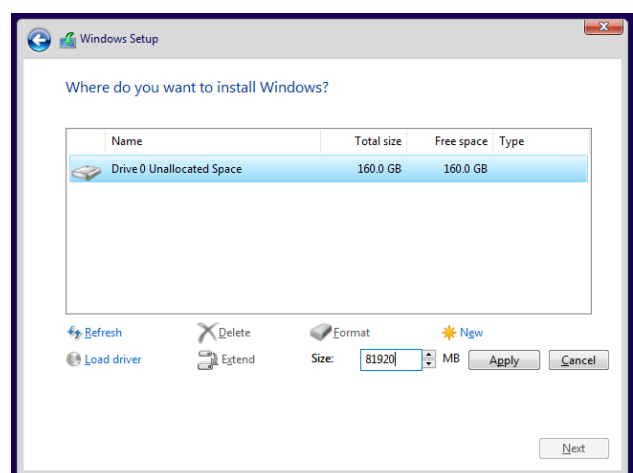
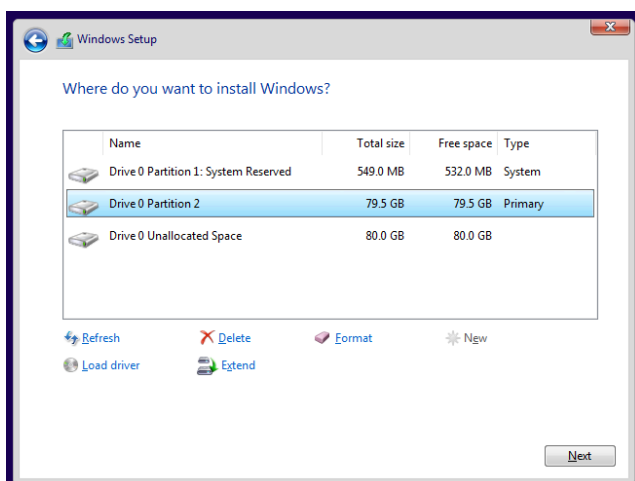
Dualboot

DUALBOOT je način korištenja dva ili više različitih operativnih sistema (OS) na jednom računaru gdje je svaki operativni sistem instaliran na zasebnoj "particiji" hard disk-a. Kod ovog pokretanja korisnik može da odabere operativni sistem koji želi da se pokrene. Za promjenu OS-a biće potrebno ponovo pokrenuti računar.

Potrebni alati su isti kao kod pravljenja bootabilnih stikova.

Proces instalacije drugog operativnog sistema će zavisi od baznog operativnog sistema koji već postoji. Ako linux sistem je bazni ,instalacije će se vršiti preko **GRUB Loadera**, a ako je bazni windows operativni sistem koristi se **Windows Boot Loader** . Instrukcije su otprilike iste ali je redosljed i layout drugaciji.

Prije instaliranja Sekundarnog operativnog sistema provjeriti Diskove , te obezbjediti potrebne velicine particija.

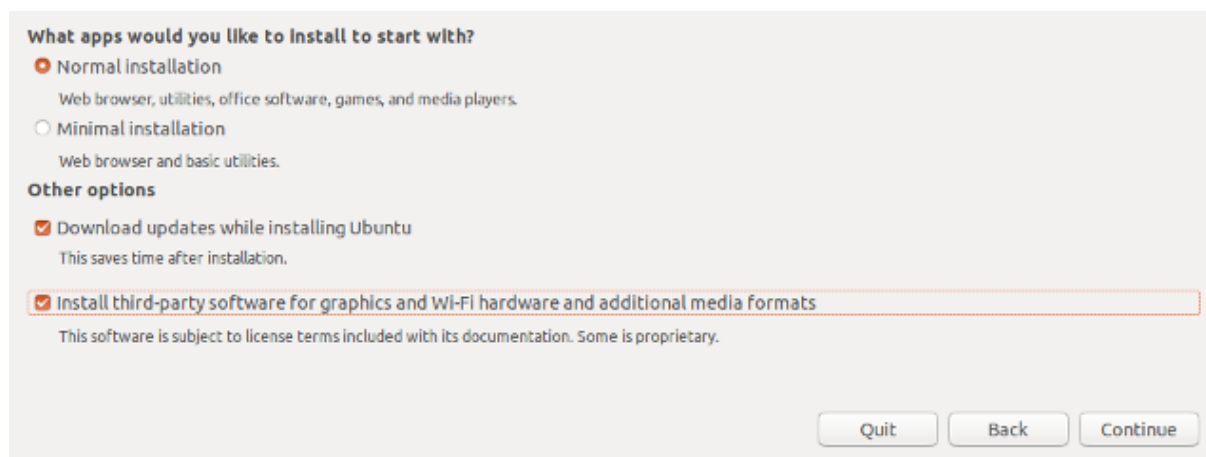


Ako već imate Windows 10 na svom računaru, preskočite instalaciju Windowsa i nastavite na Instaliranje Ubuntu instalacionim medijom. Umetnite medij i pokrenite računar sa njega. Ako sve prođe kako treba, vidjet ćete sljedeći ekran kada se medij završi učitavanjem:

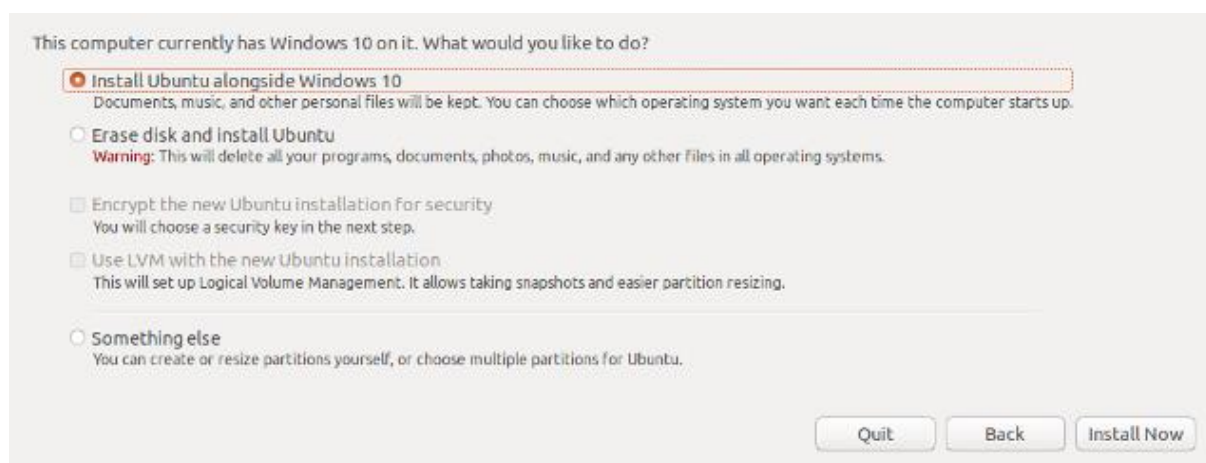


Klikom na Isprobaj Ubuntu imate prednost testirati ga prije instalacije. Provjerite možete li pristupiti internetu i dobiti audio i video reprodukciju. Ako trebate da se povežete na bežičnu mrežu, kliknite na ikonu umrežavanja u gornjem desnom uglu ekrana. Tamo možete pronaći listu bežičnih mreža i povezati se na svoju. Ako sve radi izaberite Install Ubuntu 20.04 LTS na radnoj površini da biste pokrenuli instalater.

Na sljedećem ekranu imate nekoliko opcija. Možete odabrati normalnu ili minimalnu instalaciju. Za većinu ljudi normalna instalacija je idealna. Napredni korisnici mogu umjesto toga htjeti napraviti minimalnu instalaciju, koja ima manje instaliranih softverskih aplikacija prema zadanim postavkama.

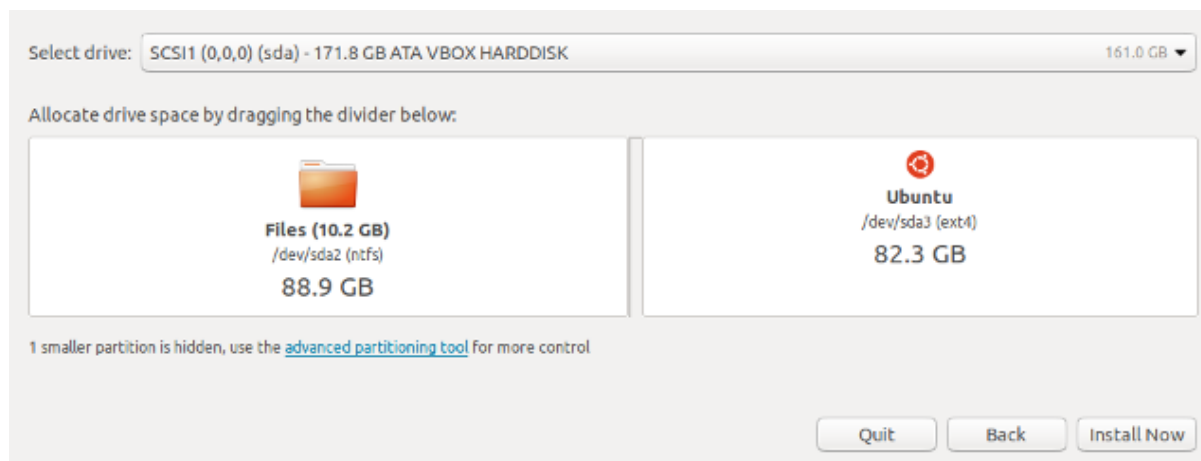


Sledeći ekran vas pita da li želite da obrišete disk ili da podesite dvostruko pokretanje odnosno **DUALBOOT**. Budući da se pokrećete dvostruko, odaberite **Install Ubuntu alongside Windows 10**.

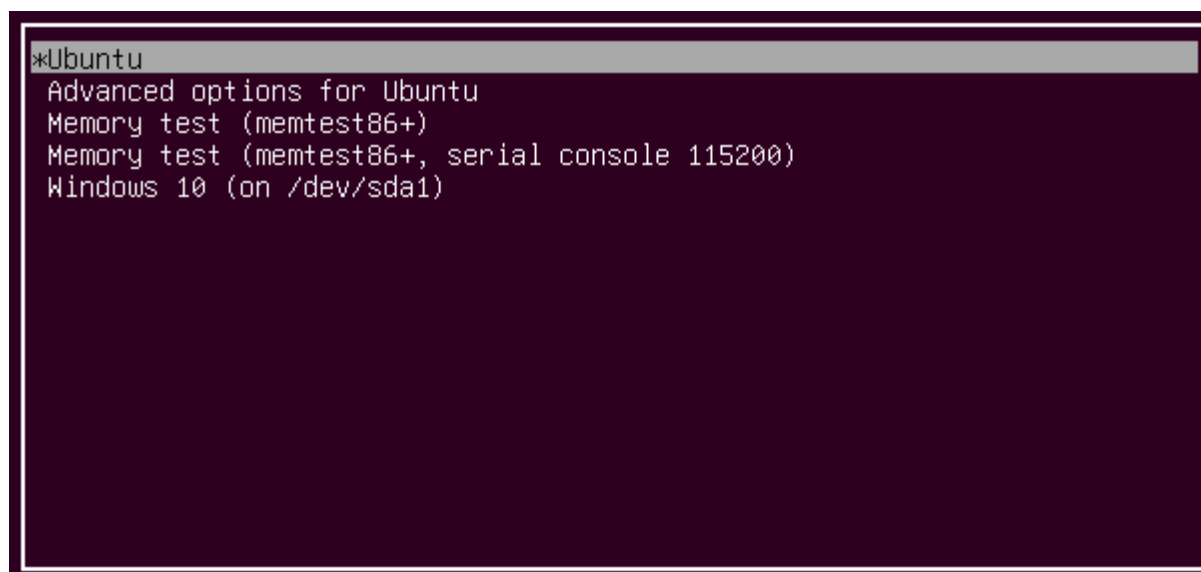


Ako ste instalirali Windows od nule i ostavili **UNALLOCATED** prostor na disku, Ubuntu će se automatski postaviti u prazan prosto. Ako ste već imali instaliran Windows 10 i on zauzima cijeli disk, pojavit će se novi prozor s opcijama za izbor diskova. Ako imate samo jedan disk, možete odabrati koliko prostora želite „ukrasti“ iz Windowsa i primijeniti na Ubuntu. Mišom možete povući okomitu liniju u sredini lijevo i desno da odvojite prostor jednom i date ga drugom. Podesite ovo tačno onako kako želite, a zatim kliknite na Instaliraj odmah.

NAPOMENA: Za Windows poželjno je ostaviti minimalno 40GB prostora.



Ako je sve išlo po planu, trebalo bi da vidite ekran sličan ovom kada se računar ponovo pokrene:



Odaberite Ubuntu ili Windows 10; ostale opcije su za rješavanje problema.

Multiboot Operativnih Sistema (LINUX)

Najbolja opcija za **multiboot** sistema je USB za više pokretanja koji omogućava pokretanje nekoliko ISO datoteka sa jednog uređaja. **ISO datoteke su datoteke koje sadrže sve potrebne instalacione datoteke za određeni program.**

Multiboot USB je koristan jer objedinjuje sve što vam je potrebno za ponovno pokretanje ili popravku vašeg sistema na jednom fleš disku. Možete učitati operativne sisteme, kao i antivirusne alate ili alate za pravljenje rezervnih kopija na jedan USB. Osim toga, uz niz dostupnih besplatnih alata, to je izuzetno isplativ i jednostavan način obavljanja stvari. Postoji niz besplatnih Windows programa koji pojednostavljaju kreiranje USB-a za višestruko pokretanje.

Jedan od najpoznatijih alata je **YUMI** s primarnim fokusom na Linux distribucije, ali se može koristiti i za druge programe.

Unutar programa se odabere USB fleš disk te iz liste ISO datoteka koju želite da sačuvate na disku. Pregledajte sve svoje izbore i kliknite na "Create" Kada završite, bit ćete upitani da li želite dodati još ISO datoteka. Kliknite Da i ponovite postupak.

Drugi programi za kreiranje MULTIBOOT USB-a su: Ventroy, WinSetupFromUSB; itd.

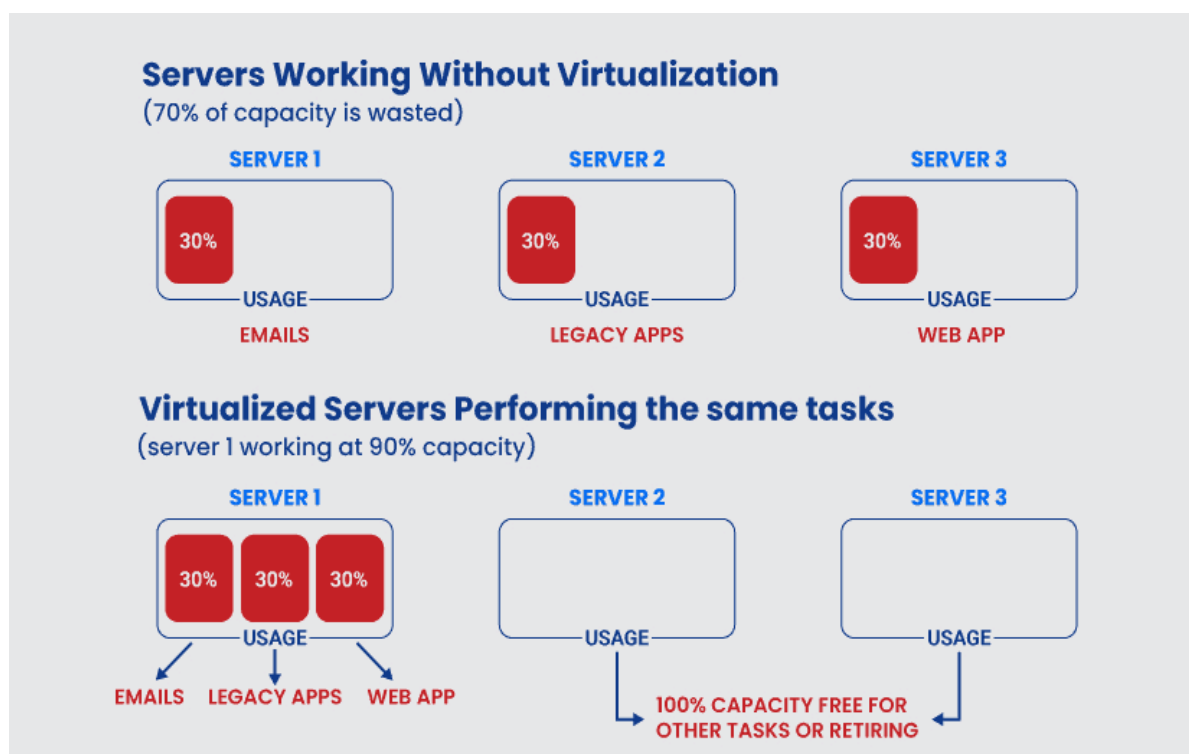
Ako želite da pokrenete sistem pomoću MULTIBOOT USB-a trebate ga obavezno ubaciti prije nego što pokrenete računar. Kada se računar uključi, odmah idite na meni za pokretanje. Odaberite svoj fleš disk i pokrenite proces pokretanja. Moraćete da izaberete koji operativni sistem želite da koristite.

Virutalizacija

Virtualizacija je jedna od posljedica koja je nastala razvijanjem Cloud Computinga. U idealnim situacijama omogućava efikasnu upotrebu računarskih resursa. U tradicionalnim sistemima do sada kada je potreban neki resurs da se osposobi na nekoj mašini. Za njega se fizički odvaja nova mašina na kojoj će se instalirati sve što je potrebno za rad takvog sistema. Problem u tome je što taj način kada se skalira na veliki broj servisa stvara ogromnu količinu neiskorištenog resursa unutar te mašine.

Pošto znamo da svaki računar ima glavne performansne parametre (CPU, Memory, Storage, OS), **virtualizacija uzima fizičke parametre i od njih pravi minijature „particije“ koje služe kao mali neovisni fizički resursi unutar jednog glavnog.** Tom tehnikom ako naprimjer naša glavna mašina ima resurse (4CPU, 8GB Ram, 1TB Disk) možemo rastaviti na 4 „virtualno“ neovisne manje mašine koje su resursa (1CPU, 2GB Rama i 256GB Disk), ova 4 računara dijele zajedničke resurse iz mašine koje su nastale ali predstavljaju izolovan sistem na kojem svaki servis može da radi.

Primjenjujući ovu implmentaciju na naš prvobitni problem možemo na jednoj mašini provizionirati 2 VM dati im 20% i 60% resursa glavne mašine potrebne za rad što nam kao rezultat ostavlja 20% glavne mašine koju možemo upotrijebiti za dalji rad.



Kao što se može na slici vidjeti postoji veliki broj benefita upotrebe virtualizacije.

- Manji troškovi,
- veća efikasnost,
- brzina konfiguracije,
- lakši rad.

Svaka virtuelna mašina koja se napravi predstavlja potpuno neovisan sistem koji se može tretirati neovisno o ostalim sistemima. Primjer toga je testiranje sistema gdje naprimjer u razvoju aplikacije se mogu napraviti 3 testna okruženja (VM) gdje će svaki da testira djelovanje aplikacije na različitom operativnom sistemu. Jedna virtuelna mašina može da testira aplikaciju na Windows okruženju dok druga može da radi to isto testiranje na Linuxu, te treća može da simulira Mac OS operativni sistem i gleda ponašanje aplikacije na tom sistemu.

U praksi razlikujemo 2 tipa hipervizora;

Dijele se na:

- Hipervizor tipa 1
- Hipervizor tipa 2

Hipervizor

Hipervizor predstavlja softverski sloj koji se nalazi između naših virtuelnih mašina i hardvera iz kojih se provizionira. On omogućava da svaka virtuelna mašina dobija dovoljno resursa koji su joj potrebni za dalji rad. Preko njega se vrši sva manipulacija virtuelnim mašinama koja je moguća. Isto tako Hipervizori služe za sinhroniziraju virtuelne mašine na način da ne dozvoljavaju da resursi provizionirani za jednu mašinu budu upotrebljeni od strane druge mašine. Oni vode računa oko sistema umrežavanja, te odlučuju o njihovoj vidljivosti.

Virtuelna mašina koju provizionira Hypervizor se naziva „Guest OS“, dok računar sa kojeg se provizionira ova mašina se naziva „Host OS“.

Naziv Hypervizor je nastao kao izvedenica iz naziva „supervizor“. Supervizor je kolokvijalni naziv za kernel Linux operativnih sistema. Simbolično značenje koje treba ovo da predstavlja jeste da hipervizor predstavlja koncept supervizora supervizoru. Koji omogućava klijentu da sa „Host OS“, provizionira proizvoljan broj „Guest OS“ kojima korisnik pristupa, a sav taj proces se vrši grafički preko platforme za virtualizaciju.

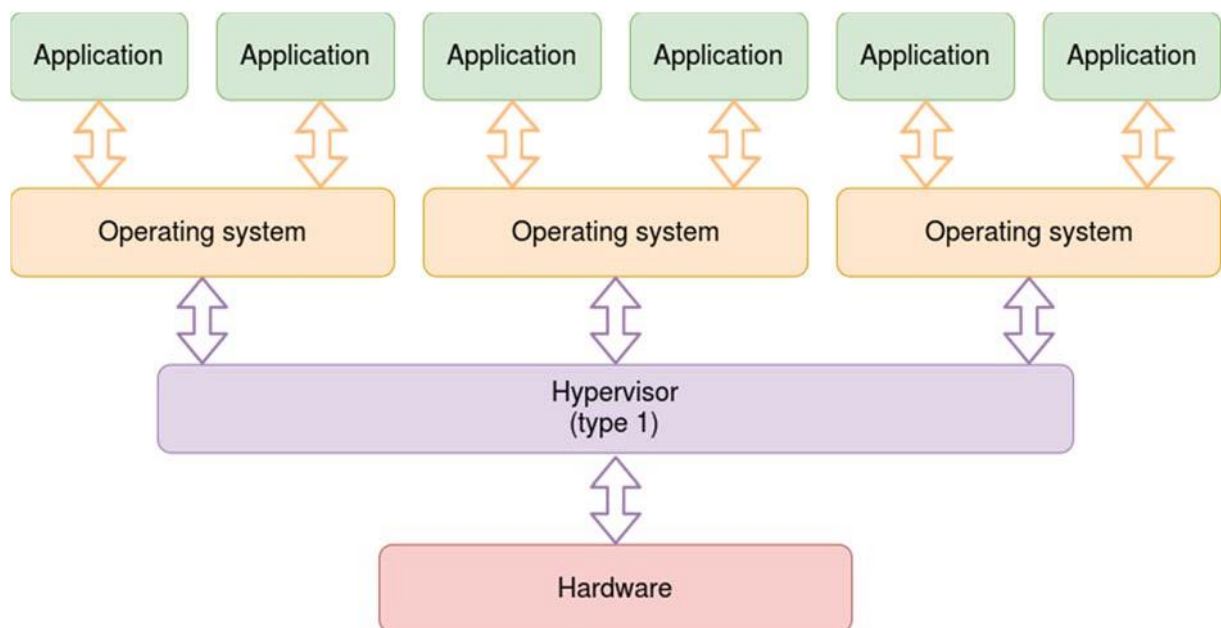


Hipervizor tipa 1

Ovaj tip hipervizora se često još naziva i „Bare-metal hypervisor“ zato što on predstavlja sistem gdje nema operativnog sistema između hardvera i virtualiziranog operativnog sistema.

Sva virtualizacija se vrši direktno instalacijom Hypervizora na Hardver i iz njega se provizioniraju virtuelne mašine potrebne za rad. Ovaj način jeste komplikovaniji ali daje veću brzinu zato što eliminišemo posrednika koji nam nije potreban ako koristimo ovaj način provizioniranja.

Neki bare-metal hipervizori se instaliraju u firmware direktno na BIOS matične ploče. Ovo je potrebno u slučaju da korisnik želi da omogući ili onemogući upotrebu virtualizacije na svoj računaru. Manipulisanje ove opcije se vrši u BIOSu datog računara.



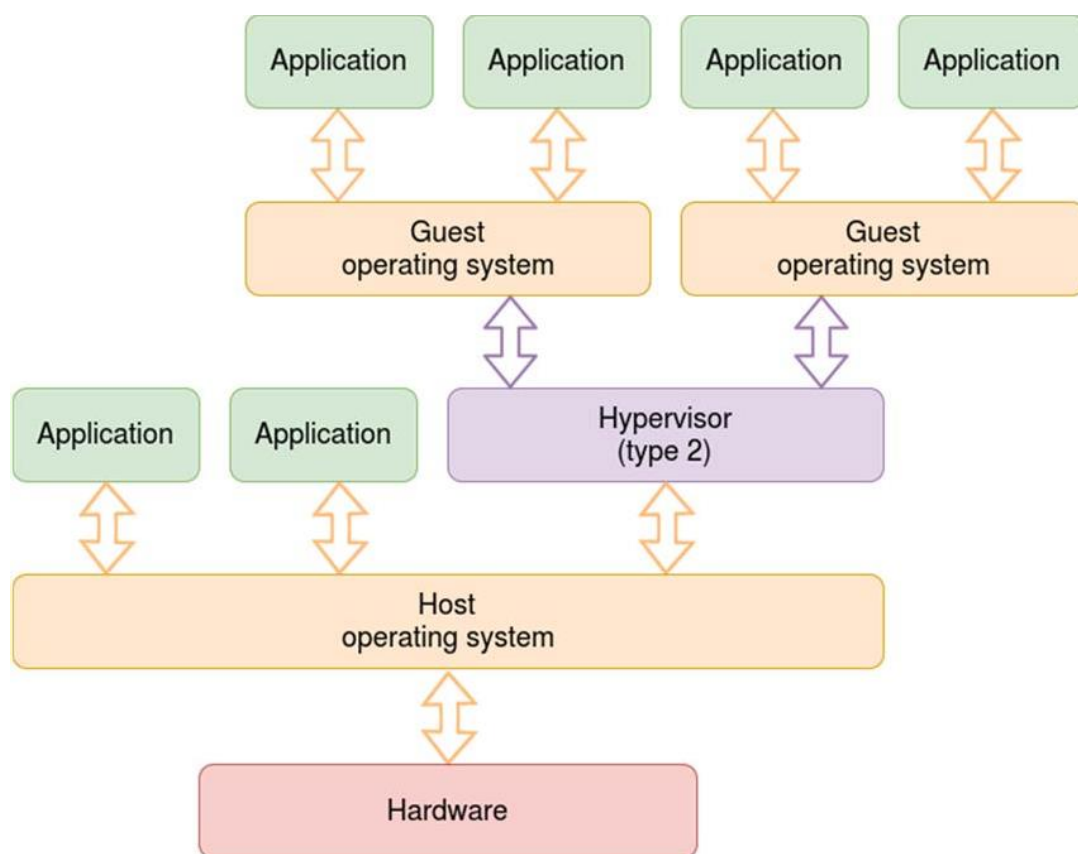
Poznati Hypervizori Tipa 1:

- Microsoft Windows Server Hyper-V,
- Vmware Sphere,
- Red Hat Enterprise Virtualization (RHEV).

Hipervizor tipa 2

Ovaj tip Hipervizora se još naziva i „Hosted“ tip. Ovaj tip karakteriše posredstvo operativnog sistema na koji se instalira Hypervizorski sofver na kojem se tek onda provizioniraju virutelne mašine za rad. Ovaj sistem nema efikasnost i brzinu koju imaju Hipervizori tipa 1 iz razloga što svaki zahtjev se ne vrši direktno na hardveru glavnog računara nego prolazi kroz Operativni sistem koji vrši taj zahtjev umjesto te virtuelne mašine te vraća odgovor kroz taj opertivni sistem nazad prema virtuelnoj mašini.

Ovaj proces sam po sebi je komplikovaniji i koristi se u situacijama gdje je virtualizacija privremena a ne dugotrajna i gdje opterećenje i upotreba nije velika da može uzrokovati problem na sistemu.



Neki od primjera Hypervizora tipa 2:

- Oracle VirtualBox,
- Vmware Workstation,
- Hyper-V Manager.

Kontejnerizacija

Virtualizacija nam pomaže u stvaranju virtuelnih okruženja iz konkretnih fizičkih resursa koje posjedujemo. Ta virtuelna okruženja predstavljaju minijaturnu ili odsječenu verziju glavnog fizičkog resursa koji samostalno posjeduje sve funkcionalnosti koje ima i glavni fizički resurs. Virtualizacijom rješavamo problem dodjele aplikacije koja zauzima malo prostora na većem fizičkom resursu, ali kao što vidimo ogroman broj aplikacija, servisa itd., ima neke univerzalne kategorije po kojima se može filtrirati. Zajednički faktor koji će biti predmet fokusa jeste **operativni sistem** na kojem se vrše aplikacije.

Vratimo se na prijašnji primjer oko razvojnih okruženja koje koriste firme (DEV,STG,PROD,UAT). Problem koji možemo predvidjeti u ovom slučaju jeste da jedina stvar po kojoj bi se u idealnom slučaju trebali da razlikuju ova okruženja su **podaci i kod koji aplikacija sadrži**. Ostale sve stvari kao što su:

- Operativni Sistem,
- Middleware,
- Runtime

se ponavljaju i stvaraju nepotrebnu redudanciju prilikom konfiguracije. U ovim situacijama dolazimo do prave snage koncepta nazvanog **Kontejnerizacija**.

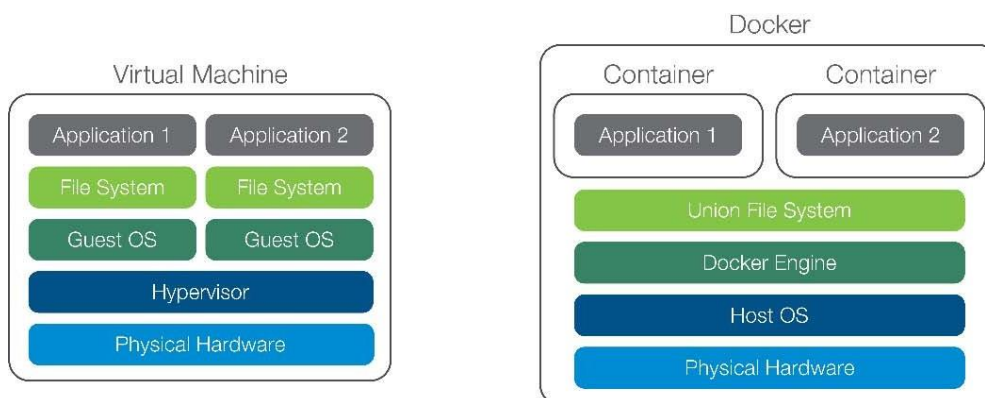
Na isti način kako virtualizacija rješava problem nezavisnih serverskih okruženja za aplikaciju. Kontejnerizacija koristi isti princip ali ga primjenjuje na operativne sisteme koji su potrebni za rad zadane aplikacije. Kontejnerizacija nam daje mogućnost da unutar jednog fizičkog ili virtualnog resursa napravimo instancu operativnog sistema koji se veže za taj resurs i može predstavljati neovisnu minijaturnu cijelinu unutar tog resursa. Glavna razlika u odnosu na koncept virtualizacije je što kontejneri dijele resurse (fizičke ili virtualne) ali stvari koje su neovisne su operativni sistem te programi koji su na njemu.

Kontejneri se vežu za aplikacije, više aplikacija može neovisno raditi u izolovanom okruženju u svom kontejneru koristeći samo potrebne resurse vezane za tu aplikaciju.

Upotreba

Jedni od glavnih razloga za upotrebu ovog koncepta jeste:

- Portabilnost – Kontejneri predstavljaju gotove upakovane aplikacije u izvršnom formatu koji u sebi ima skup svih potrebnih instrukcija i zahtjeva koji su mu potrebni za izvršavanje te aplikacije. Ovo omogućava da se ova upakovana aplikacija izvrši na bilo kojem sistemu i garantuje rad te aplikacije uvijek.
- Agilnost – Rješava probleme cross-platform developmenta.
- Brzina – Zbog toga što dijele zajednički kernel operativnog sistema kontejneri su veoma brzi zato što dupliciraju dijelove operativnog sistema koji su mu eksplicitno potrebni za rad s aplikacijom.
- Tolerancija na greške – pošto su ovo izolovana okruženja, greška na jednom kontejneru neće uticati ninašta drugo izvan tog kontejnera.
- Lagano korištenje.
- Sigurnost.



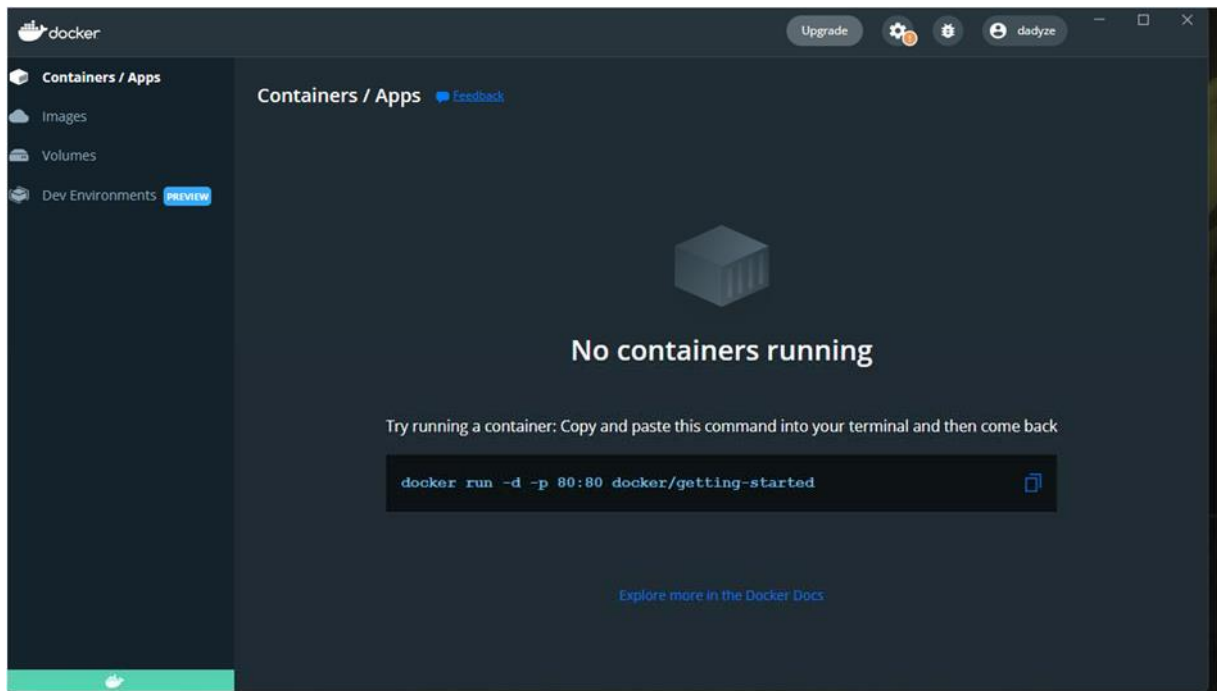
Docker

Trenutno najveći alat po upotrebi koji se koristi za kontejnerizaciju se naziva docker. To je alat koji se instalira u 2 oblika.

- GUI – Docker desktop,
- LUI – Kao dependency preko terminala.

Način na koji docker funkcioniše je da, poput hipervizorskog sofvera u virtualizaciji , instalira takozvani “Docker Engine” koji služi za upravljanje i manipulaciju kontejnera.

GUI interface za Docker Desktop



Korištenje grafičkog interface-a olakšava upotrebu ali u produkcijskom okruženju preferira se upotrebom komandi u terminalu. Primjer nekih koje su jako česte su:

- Docker ps – izlistava procese,
- Docker images – izlistava snimljenje predloške,
- Docker run -d -p 80:80 docker/getting-started – pokreće container,
- Docker stop <ID>
- Docker rm <ID>.

Za sve eventualne primjedbe, komentare, sugestije obratiti se na mail:
narcisa.hadzajlic@dl.unze.ba , **adin.jahic2019@size.ba**