

# Modularizacija SquashFS Linux datotecnog sistema

by

Dajan Brackovic

Submitted to the Odsjek za racunarstvo i informatiku  
in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science and Engineering

at the

ELEKTROTEHNICKI FAKULTET U SARAJEVU

June 2020

© Dajan Brackovic, MMXX. All rights reserved.

The author hereby grants to ETF permission to reproduce and to  
distribute publicly paper and electronic copies of this thesis document  
in whole or in part in any medium now known or hereafter created.

Author .....  
Odsjek za racunarstvo i informatiku  
May 18, 2020

Certified by .....  
Samir Ribic  
Professor  
Thesis Supervisor

Accepted by .....  
Zeljko Juric  
Prodekan za nastavu

# Contents

<b>Abstract</b>	<b>3</b>
<b>Uvod</b>	<b>4</b>
<b>Sistemske zahtjevi</b>	<b>5</b>
<b>Priprema radnog okruženja</b>	<b>6</b>
SquashFS paket . . . . .	6
<b>Modul NodeJS</b>	<b>7</b>
Rezultat Modul NodeJS . . . . .	11
<b>Modul MySQL</b>	<b>12</b>
<b>Modul Docker</b>	<b>13</b>

# Abstract

This work is describing the process of modularization of Linux SquashFS filesystem. Modularization is performed by manual customization of packages in the live system, then extracting that customized system as a separate module. This thesis will show how to create 3 separate modules out of the same base image, which will be the ubuntu-18.04.4-desktop-amd64.iso. We will be using the SquashFS tools to make the modifications inside the ubuntu-18.04.4-desktop-amd64.iso image

# Uvod

Zasto uopce mijenjati instalacioni iso image operativnog sistema? Postoji nekoliko razloga:

1. Da bismo napravili svoju distribuciju mijenjajuci postojeću iso datoteku
2. Da bismo predstavili odredjenu aplikaciju
3. Radi lokalizacije na odredjeni jezik
4. Da bismo uklonili odredjene softverske pakete
5. S ciljem dodavanja novih softverskih paketa
6. U svrhu azuriranja softverkih paketa
7. Radi mijenjanja systemske konfiguracije kao sto su teme, ikone, fontovi, pozadina...

Najlaksi nacin modifikacije iso image-a baziranih na Ubuntu distribuciji je koriscenjem "Ubuntu Customization Kit" alata. Medjutim ovaj rad ce obuhvatiti drukciju princip, manualni.

Svaki od modula koji su kreirani su bazirani na istom base image-u, ubuntu-18.04.4-desktop-amd64.iso. Modifikacijom istog dobit cemo tri modula:

- 1 Modul NodeJS - ubuntu-with-nodejs-18.04-amd64.iso
- 2 Modul MySQL - ubuntu-with-mysql-18.04-amd64.iso
- 3 Modul Chrome - ubuntu-with-chrome-18.04-amd64.iso

# Sistemiški zahtjevi

Da biste se uputili u ovaj zadatak postoji prije svega nekoliko hardverskih minimuma koje vasa radna masina treba da ispunjava:

1. Najmanje 5GB slobodnog prostora na disku, mada poželjno bi bilo mnogo više od 5GB, pogotovo ukoliko pravite razlicitih modula.
2. Najmanje 512MB RAM memorije i 1GB alocirane swap memorije.
3. Linux kernel sa squashfs podrskom.
4. QEMU/KVM || VirtualBox || VMWare - bilo koji od ova 3 alata za testiranje kreiranih modula.
5. genisoimage - paket za generisanje novog iso image-a

# Priprema radnog okruzenja

Instalirati squashfs-tools i genisoimage:

---

```
1 sudo apt-get install squashfs-tools genisoimage
```

---

## SquashFS paket

Paket squashfs-tools implementira 2 funkcije koje se koriste u ovom radu a koje pruža SquashFS <http://tldp.org/HOWTO/SquashFS-HOWTO/whatis.html>. Radi se o funkcijama **mksquashfs** i **unsquashfs**. Prva od navedenih koristi se za kreiranje squashfs dateteke, dok se druga funkcija koristi za raspakivanje kompresovane squashfs datoteke. SquashFS je moguće instalirati kao dodatak na linux jezgro. Prema tome moguće ga je instalirati na različite linux distribucije. Za Debian distribuciju njegov naziv je squashfs-tools.

# Modul NodeJS

NodeJS Modul ce biti kreiran od istog baznog modula kao i svi ostali moduli. To je ubuntu-18.04.4-desktop-amd64.iso datoteka:

---

```
1 mkdir ~/squashfs/livedtmp
2 mkdir ~/squashfs/livedtmp/isoimgs
3 cd ~/squashfs/livedtmp
```

---

Napraviti mnt direktorij unutar livedtmp direktorija u koji ce biti mount-an ubuntu-18.04.4-desktop-amd64.iso image:

---

```
1 mkdir mnt
2 sudo mount -o loop ./isoimgs/ubuntu-18.04.4-desktop-amd64.iso mnt
```

---

Napraviti direktorij extract-cd u kojeg cemo kopirati mnt direktorij izostavljajuci filesystem.squashfs datoteku unutar /casper direktorija:

---

```
1 mkdir extract-cd
2 sudo rsync --exclude=/casper/filesystem.squashfs -a mnt/ extract-cd
```

---

Napraviti direktorij za modul nodejs i kopirati u njega extract-cd direktorij:

---

```
1 mkdir modul-nodejs
2 sudo rsync -a extract-cd/ modul-nodejs
```

---

U ovom trenutku cemo upotrijebiti unsquashfs funkciju iz squashfs-tools paketa. Te cemo kopirati raspakovani squashfs-root direktorij u edit direktorij. Ovaj edit direktorij cemo kasnije koristiti da unutar njega instaliramo nodejs pakete:

---

```
1 sudo unsquashfs mnt/casper/filesystem.squashfs
2 sudo mv squashfs-root/ edit
```

---

Da bi imali mrežnu konekciju unutar edit direktorija jedno rješenje je kopirati /run direktorij unutar edit direktorija. Najbolje manuelno popuniti resolv.conf unutar edit direktorija, isto i za etc/hosts datoteku:

---

```
1 sudo cp /etc/resolv.conf edit/etc/  
2 sudo mount -o bind /run/ edit/run
```

---

Kopirati i hosts direktorij/:

---

```
1 sudo cp /etc/hosts edit/etc/
```

---

Namjestiti edit/dev direktorij kopirajući /dev/ direktorij sa hosta, zatim chroot u edit direktorij. Obaviti mount instrukcije navedene ispod. Ukoliko korisnik odluči da obriše edit direktorij iz nekog razloga, bilo bi potrebno uraditi unmount edit direktorija da sistem domaćin ne bi postao neupotrebljiv:

---

```
1 sudo mount --bind /dev/ edit/dev  
2 sudo chroot edit  
3 mount -t proc none /proc  
4 mount -t sysfs none /sys  
5 mount -t devpts none /dev/pts
```

---

Takodje potrebno je izvršiti sljedeće komande da bi se izbjegli problemi sa lokalizacijom:

---

```
1 export HOME=/root  
2 export LC_ALL=C
```

---

Za ispis svih instaliranih paketa:

---

```
1 dpkg-query -W --showformat='${Installed-Size}\t${Package}\n' | sort -nr | less
```

---

Instalacija nodejs paketa:

---

```
1 apt-get update  
2 apt-get install curl  
3 curl -sL https://deb.nodesource.com/setup_13.x | sudo -E bash -  
4 apt-get install -y nodejs
```

---



Nakon zavrsetka instalacije izvesti unutar chroot:

---

```
1 apt-get clean
2 rm -rf /tmp/* ~/.bash_history
3 rm -rf /tmp/* ~/.bashrc
4 rm /var/lib/dbus/machine-id
5 rm /sbin/initctl
6 dpkg-divert --rename --remove /sbin/initctl
7 umount /proc || umount -lf /proc
8 umount /sys
9 umount /dev/pts
10 exit
11 sudo umount edit/dev
```

---

Ponovno generisati filesystem.manifest:

---

```
1 sudo cp extract-cd/casper/filesystem.manifest extract-cd/casper/filesystem.manifest-
  desktop
2 sudo sed -i '/ubiquity/d' extract-cd/casper/filesystem.manifest-desktop
3 sudo sed -i '/casper/d' extract-cd/casper/filesystem.manifest-desktop
4 sudo umount edit/dev
```

---

Sada cemo upotrijebiti drugu funkciju iz squashfs-tools, a to je mksquashfs. S tom funkcijom cemo kompresovati edit direktorij u novu filesystem.squashfs datoteku. U kodu ispod je potrebno izvesti komandu iz linije 1 i jednu od preostale 3, pri cemu prva (komanda na liniji 2) daje najslabiju kompresiju, ali je najbrza. Druga komanda se duze izvršava ali je veci procenat kompresije u odnosu na prvu komandu. Dok je kod trece komande procenat kompresije najveći, a vrijeme izvršenja najduže:

---

```
1 sudo rm extract-cd/casper/filesystem.squashfs
2 sudo mksquashfs edit extract-cd/casper/filesystem.squashfs -nozma
3 sudo mksquashfs edit extract-cd/casper/filesystem.squashfs -b 1048576
4 sudo mksquashfs edit extract-cd/casper/filesystem.squashfs -comp xz -e edit/boot
```

---

Naredni korak je da azuriramo filesystem.size datoteku:

---

```
1 sudo su
2 printf $(du -sx --block-size=1 edit | cut -f1) > extract-cd/casper/filesystem.size
3 exit
```

---

Nakon toga upisati naziv image-a unutar README.diskdefines. Upisati 'Ubuntu with NodeJS 18.04.4 LTS "Bionic Beaver" - Release amd64' u polje DISKNAME:

---

```
1 cd extract-cd
2 sudo rm md5sum.txt
3 find -type f -print0 | sudo xargs -0 md5sum | grep -v isolinux/boot.cat | sudo tee md5sum
   .txt
```

---

Azurirati md5sum.txt datoteku:

---

```
1 sudo gedit extract-cd/README.diskdefines
```

---

Napokon mozemo napraviti iso image koji ce da sadrzi NodeJS modul. Za ovu operaciju koristimo funkciju genisoimage. Neke linux distribucije nude mkisofs funkciju. Tako da ukoliko ne radi jedna trebala bi druga:

---

```
1 sudo genisoimage -D -r -V "$IMAGE_NAME" -cache-inodes -J -l -b isolinux/isolinux.
   bin -c isolinux/boot.cat -no-emul-boot -boot-load-size 4 -boot-info-table -o ../
   ubuntu-with-nodejs-18.04-amd64.iso .
```

---

Sada cemo napraviti virtuelni hard disk pomocu qemu-img komande da bismo pokrenuli na njemu nas novi modul NodeJS Ubuntu.

---

```
1 cd ~
2 qemu-img create ubuntunodejs.img 5G
```

---

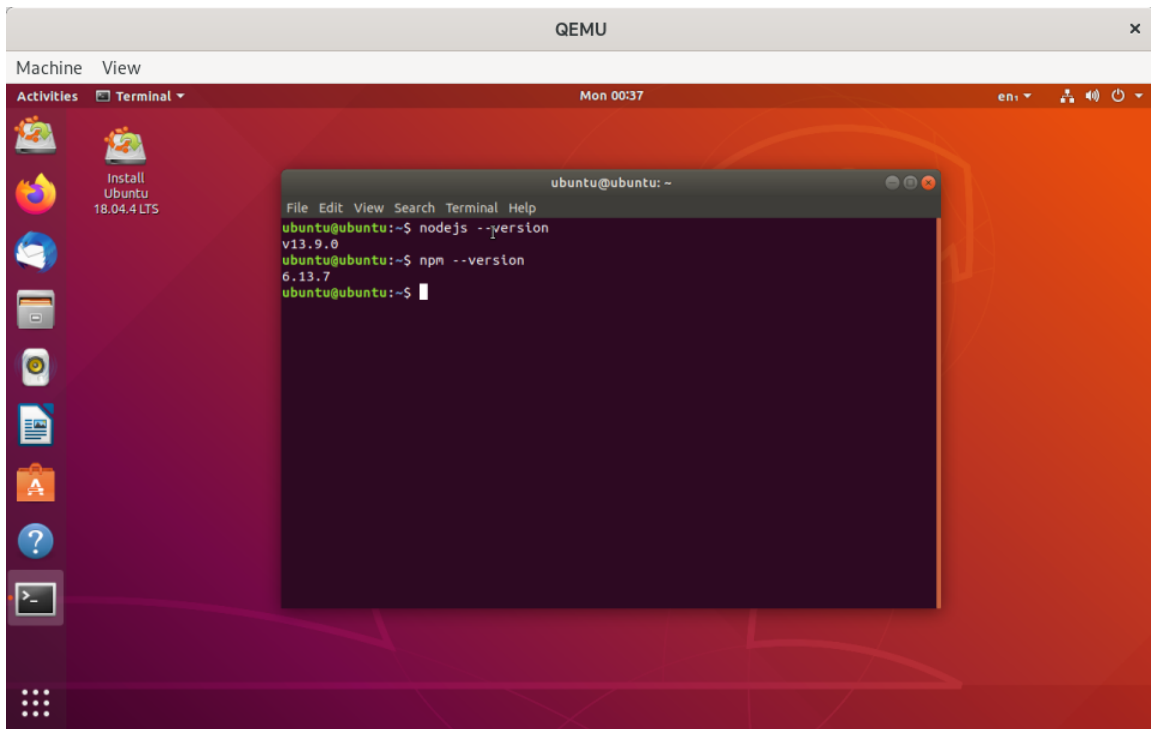
Pokrenucemo modul pomocu KVM-a:

---

```
1 sudo kvm -hda ubuntunodejs.img -cdrom ~/zavrzni/livecdtmp/ubuntu-with-nodejs
   -18.04-amd64.iso -boot d -m 2048
```

---

## Rezultat Modul NodeJS



Unutar ove live instalacije mozemo upotrijebiti nodeJS biblioteku te kreirati jednostavnu web aplikaciju. Prateci uputstvo na linku: <https://docs.microsoft.com/en-us/azure/app-service/app-service-web-get-started-nodejs> unutar nase live distribucije sa preinstaliranim NodeJS bibliotekama izvorsimo sljedece komande koristeci Terminal:

- 1 `git clone https://github.com/Azure-Samples/nodejs-docs-hello-world`
- 2 `cd nodejs-docs-hello-world`
- 3 `npm start`

Nakon toga NodeJS bi trebao pokrenuti server kojeg mozemo provjeriti web pregledniku na URL-u:

- 1 `http://localhost:1337`

Za potrebe rada nije radjena modifikacija ove web aplikacije, ali moguće je iskoristiti aplikaciju kao bazu za nadogradjivanje po želji. HTTP web server se kreira unutar index.js datoteke te bi pocetna modifikacija bila svakako nadogradnja ove datoteke za dodatnim funkcionalnostima.

# Modul MySQL

MySQL

# Modul Docker

Docker