

“Mladi za napredek Maribora 2017”
34. srečanje

Kako bistvena je lastniška programska oprema?

RAČUNALNIŠTVO
Raziskovalna naloga

Prostor za nalepko

Avtor: MIHA FRANGEŽ, MIHA ŠIROVNIK
Mentor: BRANKO POTISK
Šola: SREDNJA ELEKTRO-RAČUNALNIŠKA ŠOLA MARIBOR

Maribor, februar 2017

Kazalo vsebine

Kazalo vsebine	1
Kazalo slik	Napaka! Zaznamek ni definiran.
POVZETEK	3
1. Kaj je odprta koda in zakaj je pomembna	4
1.1. Prodaja odprtokodnih programov	4
1.2. Prednosti odprte kode	5
Večja izbira	5
Boljša podpora	5
1.2. Odprtokodne licence	6
1.2.1. GNU General Public License	6
1.2.2. Apache License	7
1.3. Kje je odprta koda že zmagala	8
1.3.1. Spletne tehnologije	8
1.3.2. Prenosne naprave	9
1.3.3. Strežniki in omrežna infrastruktura	9
2. Metodologija	10
3. Operacijski sistemi	11
3.1. Prednosti odprtokodnih operacijskih sistemov	11
3.1.1. Podpora strojne opreme	11
3.1.2. Več možnosti za razvijalce	11
3.1.3. Izboljšana varnost	12
3.2. GNU/Linux	12
3.2.1. Namizje	12
3.2.2. Ukazna vrstica	13
3.3. Grafični gonilniki	14
4. Multimedija	16
4.1. Obdelava fotografij	16
4.1.1. Upravljanje fotografij	16
Picasa - digiKam	16
Lightroom - darktable	17
4.1.2. Urejanje fotografij: Adobe Photoshop - GIMP	18

4.2. Montaža videa: Premiere Pro - Kdenlive	18
4.3. Obdelava zvoka	19
Jack Audio Connection Kit	19
Adobe Audition Multitrack - Ardour	20
Adobe Audition Waveform - Audacity	21
5. Pisarna	22
5.1. Besedilo: Microsoft Word - LibreOffice Writer	23
5.2. Preglednice: Microsoft Excel - LibreOffice Calc	23
5.3. Predstavitve: Microsoft Powerpoint - LibreOffice Impress	24
6. Spletne storitve	24
6.1. Shramba datotek	24
6.2. Pisarna v “oblaku”	25
7. Ugotovitve	26
Operacijski sistemi	26
Pisarna	26
Multimedija	26
Storitve v oblaku	26
DRUŽBENA ODGOVORNOST	27
Viri in literatura	27
Priloga 1: Tabela odprtokodnih alternativ	28

POVZETEK

V najini nalogi bova preverila, koliko lastniške programske opreme lahko zamenjamo z odprtokodnimi rešitvami, kaj lahko s tem pridobimo in čemu bi se morali odreči. Primerjala bova več popularnih programov z mnogih področij ter poskusila ugotoviti, ali je popolnoma odprtokoden sistem dovolj dober za vsakodnevno uporabo. Programe bova primerjala glede na lahkoto uporabe, hitrost, stabilnost, ipd.. Med drugim bova primerjala programsko opremo za pisarno, obdelavo fotografij, videa in zvoka, 3D animacijo, programiranje, preizkusila bova pa tudi nekaj odprtih alternativ popularnim storitvam v oblaku.

1. Kaj je odprta koda in zakaj je pomembna

Ko slišimo izraz “odprta koda” takoj pomislimo na programsko kodo, ampak odprta koda obsega dosti več kot le to. Odprta koda je pravzaprav metodologija razvijanja, ki je lahko uporabljena v vseh projektih. Glavni značilnosti odprte kode sta: dostop do izvorne kode oziroma oblikovalnega načrta za splošno javnost in decentralizirano razvijanje kode/načrta s strani vrstnikov, ki želijo prispevati. Vsak, ki želi lahko spremeni izvorno kodo, ki mora biti vključena poleg izdelka. Spremenjeno verzijo lahko razdeljuje kot svoj *fork* izdelka.

Odprtokodni izdelki zagotavljajo uporabnikom svobodo, zato so *free*, to pa ne pomeni da so zastonj. To je zato, ker se beseda *free* uporablja na dva različna načina. Prvi način “*free as in beer*” pomeni, da je nekaj zastonj. Odprtokodni programi sicer so pogosto zastonj, ampak to ni potrebno. Drugi način uporabe besede *free* je “*free as in freedom*” ali “*free as in speech*” pomeni, da je nekaj svobodno. Ta je bolj pomemben, saj je osnovno načelo odprte kode svoboda.

Midva se bova bolj osredotočila na programsko opremo, saj je tam ta metodologija najbolj razvita in verjetno tudi najbolj pomembna. Na tem področju so bili začetki odprte kode, večinoma kot upor lastniški programski opremi, ki so jo mnogi dojemali kot preveč restriktivno.

Obstaja več definicij odprte kode, najbolj splošno sprejeta je “*The Open Source Definition*”, ki jo je izdal *Open Source Initiative*. Takoj za tem najbolj sprejeta je “*The Free Software Definition*”, ki jo je napisal Richard Stallman, zato se pogosto uporablja kratica FOSS (*Free and Open Source Software*). Razlika v definicijah je zelo mala, Open Source definicija sprejema nekatere bolj restriktivne licence, kot Stallmanova definicija.

1.1. Prodaja odprtokodnih programov

Pogosto mnenje je, da so odprtokodni programi brezplačni in je tako z njimi nemogoče zaslužiti. Čeprav je res, da je prodaja odprtokodnih programov težja kot lastniških, je to daleč od nemogoče. Dva glavna načina služenja z odprto kodo sta dvojno licenciranje in razvoj in podpora po meri.

Pri dvojnem licenciranju, avtor programa določi, za katere vrste uporabnikov velja katera licenca. Tako lahko domačim uporabnikom omogoči brezplačno uporabo podjetja, ki z njegovimi programi služijo, pa morajo kupiti licenco. O kompatibilnosti tega modela z definicijo odprte kode je sicer veliko debate, a manj restriktivne variacije tega modela uspevajo zelo dobro (primer podobnega sistema je podatkovna baza MySQL podjetja Oracle).

Profesionalna podpora in razvoj po meri, pa sta trenutno najpogostejši način služenja z odprto kodo. Podjetje, ki projekt razvija, zraven brezplačne različice ponuja plačljivo profesionalno podporo ter razvoj funkcij po meri. Primer tega je podjetje Collabora, o katerem bova več povedala v zadnjem poglavju.

1.2. Prednosti odprte kode

Večja izbira

Ker en program ne more biti enako dober na vseh področjih uporabe, je velika prednost odprte kode možnost prilagoditve za bolj specifično uporabo. Pisanje popolnoma novega lastniškega programa, bi bilo zamudno in zelo drago, pri odprtokodnih programih pa lahko uporabimo obstoječo osnovo in se osredotočimo le na naše spremembe. Morda najboljši primer prilagajanja enake osnove za optimalno delovanje v popolnoma različnih okoljih so vse različne distribucije Linuxa. Med tem ko podjetja Microsoft in Apple svoje operacijske sisteme razvijata za čimboljše delovanje v čim več različnih okoljih, se mnoge Linux distribucije osredotočijo na dosti ožje področje uporabe in v teh uspejo dosti bolje kot bolj splošne rešitve.

Primeri teh sta na primer distribucija *KXStudio*, ki z uporabo *lowlatency* različice jedra hitrejšo delovanje zamenja za manjšo zakasnitev zvočnih naprav in je tako odlična za uporabno v snemalnih studijih in distribucija *Puppy Linux*, ki z uporabo grafično manj zahtevnih programov deluje odlično tudi na zelo starih računalnikih - uporabi namreč le okrog 200MB sistemskega pomnilnika, na disku pa zasede komaj 100MB prostora.

Boljša podpora

Za razliko od lastniške programske opreme, ki jo razvija, prodaja in zanjo ponuja podporo le eno podjetje, pri odprtokodnih programih najdemo bolj decentraliziran model. Podpore za odprtokodne programe namreč ne nudi samo organizacija, ki program razvija, ampak tudi druga podjetja, ki zraven osnovne podpore nudijo tudi prilagajanje in razvoj novih funkcij, večina, pa svoje dodatke prispeva nazaj k odprtokodnemu projektu in tako izboljšuje program za vse uporabnike.

Veliko prednost ima odprtokodna programska oprema tudi pri popravljanju napak v programih. Za razliko od lastniške programske opreme, kjer se za popravke zanašamo le na eno podjetje, je možno probleme z odprtokodnimi programi rešiti na več načinov - tudi brez plačane podpore. Mnogo napak reši že skupnost sama, saj lahko kodo k projektu prispeva kdorkoli. Če podjetje zaposluje programerje, lahko tudi ti poskusijo rešiti problem. Odličen način za hitro popravljanje napak pa so tudi spletne strani kot je Bountysource¹, ki podjetjem in posameznikom omogoča prispevanje k denarni nagradi za razvijalca, ki prvi napako popravi.

Morda najpomembnejša prednost odprtokodnih programov za podjetja pa je možnost nadaljne podpore programov tudi, če prvoten projekt propade. Pogosto slišimo o bankah in državnih organizacijah, ki uporabljajo po meri narejene programe, za katere ne morejo dobiti podpore - najpogosteje zaradi razpada podjetja, ki je program naredilo. Tako so prisiljeni uporabljati zastarelo programsko opremo, posledično pa tudi stare in nepodprte operacijske sisteme. Dober primer tega je letališče v Parizu, kjer je konec leta 2015 prišlo do katastrofalne napake, saj so zaradi zastarele programske opreme uporabljali več kot 20 let star operacijski sistem Windows 3.1. Večino takšnih programov sicer ne bi bilo težko posodobiti, a zaradi ukinitve podpore s strani proizvajalca in lastniške licence to ni možno in bi bilo celo kaznivo.

¹ <https://www.bountysource.com/>

V primeru odprtokodnih programov, pa bi lahko podjetje najelo nekaj programerjev, ki bi program posodobili, morda pa bi lahko podjetje projekt celo prevzelo in nadaljevalo njegov razvoj - s čimer bi lahko tudi zaslužili.

1.2. Odprtokodne licence

Kot sva omenila že prej, odprta koda ne pomeni, da lahko s kodo kdorkoli dela karkoli želi. Čeprav so osnovna načela odprte kode enotna med vsemi licencami, se med seboj razlikujejo v stvareh, ki jih osnovna definicija odprte kode ne pokriva. Predstavila bova nekaj najpogostejših licenc in povzela glavne razlike med njimi.

1.2.1. GNU General Public License

GNU General Public License (GNU GPL ali samo GPL) je bila napisana da uporabnikom zagotovi 4 pravice:

1. pravico uporabiti delo
2. pravico proučevati delo
3. pravico kopirati in deliti delo z drugimi
4. pravico predelovati delo in pravico predelano delo deliti z drugimi

Osnovni pogoji uporabe GPL-ja so:

1. Pogoji uporabe GPLa morajo biti na voljo vsem, ki prejemejo delo pod GPL.
2. Vsi ki upoštevajo pogoje uporabe lahko delo modificirajo, kopirajo in distribuirajo. Za to lahko zahtevajo plačilo.
3. Distributer ne more dodati nadaljnjih omejitev na uporabnika.
4. Polek programa mora biti berljiva izvorna koda.
5. Če so v programu deli pod GPL mora biti vso delo pod GPL (copyleft).
6. V privatni upoabi (brez distribucije) je lahko program modificiran brez, da bi bila izvorna koda objavljena.

Bila je prva copyleft licenca za splošno uporabo. Copyleft izvira iz copyright zakonov, saj distributer nima pravice distribuirati GPL dela brez da se drži vseh pogojev licence, razen če šteje pod prosto uporabo. Zagotavlja, da so vsa modificirana dela, pod isto ali manj restriktivno licenco.

Richard Stallman je naredil prvo različico licence (*GPLv1*) tako da je združil podobne licence narejene za specifične programe, ki pred tem niso bile združljive čeprav so zagotovile vse pravice. S tem je želel ustvariti licenco, ki bi lahko bila uporabljena za vse programe. Prva verzija licence je zraven programa zahtevala tudi izvorno kodo in kopijo licence. Da bi uporabnik lahko ohranili pravice, licenca ni dovolila združevanja programov pod GPLv1 z programi, ki imajo bolj strogo licenco. Ta licenca je dovolila, da zaradi drugih legalnih dolžnosti distributer ne upošteva delov licence.

Leta 1991 je bila izdana druga verzija licence GPL (*GPLv2*), katere najpomembnejša sprememba je bil 7. Člen, ki ga avtor Richard Stallman imenuje "liberty or death clause", ki dobesedno pomeni "člen svoboda ali smrt". Člen prepoveduje distribucijo, če distributer ne izpolnjuje vseh dolžnosti licence tudi če ima druge legalne dolžnosti. Ta licenca še je danes pogosto uporabljena ponavadi kot "GPLv2 ali kasnejše verzije". Najbolj znana programska oprema, ki uporablja GPLv2 je Linux.

Istočasno je bila tudi izdana licenca Library General Public License (*LGPL*), ki je bila podobna GPLv2, ampak ni imela tako strogih omejitev na uporabo v lastniški programski opremi, saj je bila namenjena za programske knjižnice. Distributer lahko uporabi delo pod LGPL v lastniški programski opremi, brez da izda izvorno kodo delov programa, ki niso pod LGPL. Dovoljuje tudi da vsi programi pod to licenco spremenijo licenco v GPL. Druga verzija te licence (*LGPLv2.1*) je bila izdana leta 1999, takrat se je licenca preimenovala v Lesser General Public License, saj naj ne bi bila uporabljena za vse knjižnice. Tretja verzija licence je bila izdana leta 2007 z novo verzijo GPL licence.

Tretja verzija GPL licence (*GPLv3*) je bila narejena s pomočjo skupnosti. Prvi osnutek je bil za javnost na voljo 16. Januarja 2006. Po 4 osnutkih je 29. Junija 2007 uradno bila izdana GPLv3. Ta licenca je omejila moč DRMA (Digital Rights Management) saj dovoli uporabnikom da ga kršijo brez legalnih posledic. Spremenila je definicijo izvorne kode in preprečila strojno opremo, ki uporabnikom ne dovoli dostopa do izvorne kode (tivoization). Omogočila je tudi kompatibilnost z Apache License version 2.0 in GNU Affero General Public License.

1.2.2. Apache License

Apache licenca je *permissive* licenca, to pomeni, da modificirana dela ne rabijo biti licencirana pod istio licenco. Deli kode, ki niso bili modificirani morajo ostati pod Apache licenco. Vsi legalni opomini (o licenci, avtorskih pravicah, patentih...) morajo biti ohranjeni. Vsak modificiran del mora vsebovati obvestilo, da je datoteka bila modificirana. Ta licenca je tudi združljiva z GPLv3 licenco, če je končen izdelek licenciran pod GPLv3.

1.2.3. MIT License

MIT licenca je najpreprostejša licenca na tem listu. Zahteva le da lahko uporabnik brez omejitev uporablja, kopira, modificira, objavi, distribuira in prodaja programsko opremo. Vsaka kopija programske opreme mora imeti vključeno tudi kopijo licence. Zaradi te preprostosti je ta licenca združljiva z večino ostalih odprtokodnih licenc.

1.3. Kje je odprta koda že zmagala

Naša življenja so res v veliki meri odvisna od lastniške programske opreme, a ogromno tehnologije, ki jo uporabljamo vsak dan, se močno zanaša na odprtokodne projekte. V zadnjih letih vedno več podjetij objavlja odprtokodne programske knjižnice (npr. Facebookov React, Microsoftov .NET...), kot tudi odprtokodna jedra, ki poganjajo njihovo programsko opremo (npr. Googlov Chromium in Android, Applov WebKit...). Prav tako vsakodnevno uporabljamo kodo neprofitnih organizacij, skupnosti in celo posameznikov (Mozilla Firefox, Linux, GPSd...)

1.3.1. Spletne tehnologije

Naše brskanje po spletu - od strežnikov na katerih so spletne strani shranjene, do brskalnikov s katerimi si jih ogledujemo - v veliki večini poganja odprta koda. Že operacijski sistemi strežnikov, na katerih je zgrajen svetovni splet, so zelo pogosto odprtokodni - Unixu podobni operacijski sistemi namreč predstavljajo okrog 66,5%² svetovnega spleta, od teh pa je vsaj dobra polovica³ odprtokodnih (čeprav je ta številka najverjetneje dosti večja).

Pri programski opremi, ki na strežnikih servira spletne strani odprta koda prevladuje še bolj očitno, saj projekta *Apache* in *Nginx* zajemata kar 83,1%⁴ spletnih strežnikov.

Pri programskih jezikih v katerih so spletne strani napisane se zgodba ponovno ponovi, saj je kar 82,4%⁵ spletnih strani napisanih v jeziku PHP, le-temu pa sledijo še mnogi drugi odprtokodni jeziki - kot prvi jezik z lastniško licenco na lestvici se pojavi Adobov *ColdFusion* s pičlimi 0,6%. Zelo podobno je tudi pri knjižnicah za ospredni del spletnih strani⁶ - npr. jQuery.

Morda najbolj očitno pa odprta koda prevladuje na trgu spletnih brskalnikov. Čeprav brskalnika Internet Explorer in Safari kot edina zaprtokodna brskalnika zavzemata dobrih 20%⁷, je tukaj pomembno omeniti, da tudi ta uporabljata odprtokodne komponente (najpomembnejša sta Safarijev izrisovalnik WebKit ter JavaScript prevajalnik *ChakraCore*, ki ga uporabljata IE in Edge).

² https://w3techs.com/technologies/overview/operating_system/all

³ <https://w3techs.com/technologies/details/os-unix/all/all>

⁴ https://w3techs.com/technologies/overview/web_server/all

⁵ https://w3techs.com/technologies/overview/programming_language/all

⁶ https://w3techs.com/technologies/overview/javascript_library/all

⁷ <https://www.w3counter.com/globalstats.php?year=2016&month=11>

1.3.2. Prenosne naprave

Za razliko od namiznih računalnikov, kjer še vedno prevladujejo lastniški operacijski sistemi, se je odprta koda dosti bolj razširila med proizvajalci prenosnih naprav - predvsem pametnih telefonov in tabličnih računalnikov. Osnova Googlovega operacijskega sistema Android, ki poganja kar 65%⁸ mobilnih naprav, je namreč popolnoma odprtokodna. Čeprav večina proizvajalcev za svoje naprave AOSP (*Android Open-Source Project*) sistem prilagodi z lastniško kodo, večino najpomembnejših funkcij v teh napravah opravlja odprta koda. Poleg vseh osnovnih sistemskih aplikacij kot so zaganjalnik, upravitelj datotek, ura ipd., je odprtokodno tudi jedro sistema - Android namreč na najnižjem nivoju uporablja GNU/Linux. Uporaba odprtokodnega sistema kot osnovo je ter vsi pomožni programi, kot so na primer *wpa_supplicant*, ki upravlja Wi-Fi povezave in *GPSd*, ki omogoča povezavo z GPS sateliti.

1.3.3. Strežniki in omrežna infrastruktura

Kot je bilo omenjeno že prej, so odprtokodni operacijski sistemi - predvsem Linux - zelo pogosto uporabljeni na strežnikih, a to ne velja samo za spletne strani in baze podatkov. Odprta koda poganja tudi velik del osnovne infrastrukture interneta. Med DNS strežniki na primer je zdaleč najbolj popularen odprtokodni *BIND*, med MTA strežniki za E-pošto pa *Exim* in *Postfix* predstavljata dobrih 88%⁹.

Projekt *pfSense* je kot odlična alternativa dragim mrežnim usmerjevalnikom izjemno popularen v podjetjih in organizacijah, ki morajo v mrežo povezati veliko število računalnikov (npr. šolah, hotelih...). *pfSense* programsko opremo je namreč možno namestiti na strojno opremo za navadne računalnike, zaradi odprtosti projekta, pa je z več tisočmi dodatkov možno s sistemom dobiti funkcije, ki jih drugače najdemo le na specializirani opremi.

Najverjetneje največjo prevlado med vsemi kategorijami pa imajo odprtokodni sistemi med superračunalniki. Operacijski sistem GNU/Linux namreč poganja kar 497 od 500 najmočnejših superračunalnikov na svetu¹⁰.

⁸ <https://www.netmarketshare.com/operating-system-market-share.aspx?qprid=8&qpcustomd=1&qpstick=1&qpsp=2016&qpnp=2&qptimeframe=Y>

⁹ http://www.securityspace.com/s_survey/data/man.201612/mxsurvey.html

¹⁰ <https://www.top500.org/statistics/list/>

2. Metodologija

Programne, ki jih bova raziskovala sva razdelila na 4 kategorije: operacijske sisteme, pisarniške programe, multimedijske programe in spletne storitve. Izbrala sva kategorije, ki jih poznavam dovolj dobro, da lahko ocenim kako dobro so se odprtokodni programi odnesli. Znotraj teh kategorij sva poiskala več različnih področij in za vsakega poiskala vsaj en odprtokoden program in enega z lastniško licenco. Pri izbiri programov sva najbolj gledala na obsežnost in popularnost programov. Na različnih področjih sva iskala čim več programov iz istih paketov, da sva poleg programov samih lahko primerjala tudi povezanost programskih paketov (zato sva izbrala toliko programov podjetja Adobe). Upoštevala sva tudi ciljno publiko - iskala sva namreč programe z čim večjo ciljno publiko, saj je za bolj specifične programe težje najti alternative. Ker je večina teh programov plačljivih, sva seveda preferirala tiste, za katere sva lažje dobila licenco. Za večino programov sva uporabila šolske licence, pri drugih pa sva uporabila preizkusno različico.

Najprej sva oba programa primerjala teoretično. Primerjala sva na primer kompatibilnost programa z različnimi platformami, splošno znane prednosti in slabosti, ceno in uporabniški vmesnik. Nato sva vsak program še preizkusila.

Za vsako področje sva si zamislila nalogo, ki obsega čim več različnih funkcij obeh programov. Pred preizkusom sva vsakega nekaj časa uporabljala, da sva se bolje seznani z uporabo, nato pa sva s programom opravila zastavljeno nalogo.

Ko sva končala z vsemi programi z nekega področja, sva primerjala delo z vsakim programom. Zapisovala sva si stvari, ki so nama delo z nekim programom olajšale oz. otežile v primerjavi z drugim. Na koncu sva ocenila, kako dobra alternative je izbran odprtokodni program.

3. Operacijski sistemi

Operacijski sistem je najpomembnejši in najbolj zapleten kos programske opreme na vsakem računalniku. Zaradi nizkega nivoja delovanja in pomembnosti sistemskih knjižnic za delovanje čisto vseh programov je kvaliteta izvirne kode jedra in sistemskih programov še pomembnejša kot pri navadnih programih. Zato ni presenetljivo, da se je ideja odprte kode najprej in najhitreje razvila pri operacijskih sistemih. Leta 1983 je Richard Stallman ustvaril Projek GNU, katerega namen je bil napisati Unixu podoben operacijski sistem, ki bi bil popolnoma odprtokoden. Kot del tega projekta je nastalo ogromno programske opreme ki jo, ponavadi nevede uporabljamo vsak dan - tudi znotraj lastniških operacijskih sistemov (npr. OS X).

3.1. Prednosti odprtokodnih operacijskih sistemov

V objavi, v kateri je Richard Stallman prvič predstavil Projekt GNU, je bil priložen tudi "GNU Manifesto", ki je opisal cilje projekta, razložil prednosti odprte programske opreme in odgovoril na nekaj pogostih kritik oprte kode. Nekaj prednosti sva razložila že v uvodnem poglavju, tukaj pa se bova osredotočila na tiske, ki veljajo za operacijske sisteme.

3.1.1. Podpora strojne opreme

Tukaj je govora predvsem o gonilnikih strojne opreme. Sigurno vsi poznamo primere, ko smo morali doma uporabljati starejši operacijski sistem zaradi nepodprte strojne opreme (ponavadi tiskalnika), a ta problem postane še večji, ko pride do podjetij. Največ takih primerov najdemo v industriji, kjer je strojna oprema tako tesno povezana z delom, da bi bila menjava izjemno nepraktična in dosti predraga. Primer tega so npr. CNC krmilniki, menjava katerih bi zahtevala menjavo skoraj celotnega stroja. Tako računalniki, ki delajo s takšno strojno opremo, pogosto uporabljajo stare, nestabilne in nezaščitene različice operacijskih sistemov. To ni le nepraktično, zaradi nestabilnosti ali celo vdora v sistem, lahko takšni računalniki ogrozijo celotno podjetje.

Z strojno opremo sicer ni nič narobe in pogosto bi bila posodobitev gonilnikov na novejši sistem zelo enostavna, a proizvajaci te strojne opreme izvirne kode svojih gonilnikov ne dovolijo (in preprečujejo) spreminjati. Če bi v podjetju, ki uporablja odprtokodne gonilnike prišlo do podobnega problema, bi lahko najeli zunanje programerje, ki bi jim gonilnike prilagodili za delo z novim sistemom.

3.1.2. Več možnosti za razvijalce

Razvijalci programske opreme se ves čas zanašajo na operacijski sistem (sistemske knjižnice). Pogosto jim je na voljo le dokumentacija, ki jo proizvajalec operacijskega sistema ponuja. Ta dostikrat ni popolna in ne obrazloži vseh podrobnosti. Tako programerji uporabljajo sisteme, za katere sploh ne vedo kako delujejo. To lahko privede do mnogih težav, ki jih pisci dokumentacije niso prevideli. Vpogled v izvirno kodo bi programerju omogočil dosti lažji razvoj programov, ki uporabljajo napredne funkcije, ki so ponavadi zelo slabo ali celo niso dokumentirane brez. Prav tako bi lahko programer predvidel probleme,

do katerih bi lahko prišlo zaradi stvari znotraj sistema in prilagodi svoje programe, da se jim izogne.

Še ena velika prednost, ki velja za razvijalce programov in operacijskega sistema je, da bi lahko programer, ki potrebuje neko funkcijo, ki je sistem ne omogoča, sistem spremenil in svoje spremembe prispeval nazaj k operacijskemu sistemu. S tem bi se izboljšal tako operacijski sistem, kot tudi program, saj se programerju ne bi bilo potrebno zanašati na nedokumentirane metode, ki bi lahko povzročile mnogo problemov.

3.1.3. Izboljšana varnost

Med tem ko se večina zaprtokodnih programov zanaša na idejo “security through obscurity” saj menijo, da če ljudje njihove kode ne vidijo, to napadalcem onemogoča iskanje varnostnih lukenj. A kot so mnoge 0-day napake pokazale, je proces iskanja in popravljanja varnostnih lukenj v odprtokodnih programih dosti bolj učinkovit. Ker kodo razvija dosti več različnih ljudi, je napake dosti lažje odkriti. Med tem ko varnostne preglede lastniške programske opreme opravljajo podjetja, ki jih najame proizvajalec, lahko odprtokodne programe preglejujejo tudi podjetja in posamezniki, ki programe uporabljajo in jim je varnost zelo pomembna. Tudi popravljanje napak je hitrejše, saj celotna skupnost sodeluje pri testiranju.

3.2. GNU/Linux

Pri poimenovanju GNU/Linux sistemov pogosto prihaja do napak in nestrinjanja, zato bova najprej pojasnila, kaj katera različica imena pomeni.

Ime **Linux** je pogosto uporabljeno, ko govorimo o operacijskih sistemih kot so Ubuntu in Fedora. To poimenovanje je napačno, saj je Linux ime samo jedra operacijskega sistema, kar ne vključuje vseh knjižnic in programov, vključenih v te sisteme. Pravilnejši izraz je **Linux distribucije**, ki vključuje jedro ter vse systemske in uporabniške programe.

Večina knjižnic in systemskih programov iz teh distribucij prihaja iz Projekta **GNU**, zato Richard Stallman podpira poimenovanje **GNU/Linux** (tudi **GNU+Linux** in **GNU & Linux**). GNU sistem pa lahko namesto Linuxa uporablja tudi druga jedra, na primer uradno GNU jedro imenovano **Hurd** - sistem s tem jedrom se tako imenuje **GNU/Hurd**.

Zaradi največjega števila uporabnikov in ker nimava na voljo nobene Appleove strojne opreme, bova tukaj, kot tudi v nadaljevanju naloge, za primerjavo zaprtokodnih sistemov uporabila operacijski sistem Microsoft Windows 10.

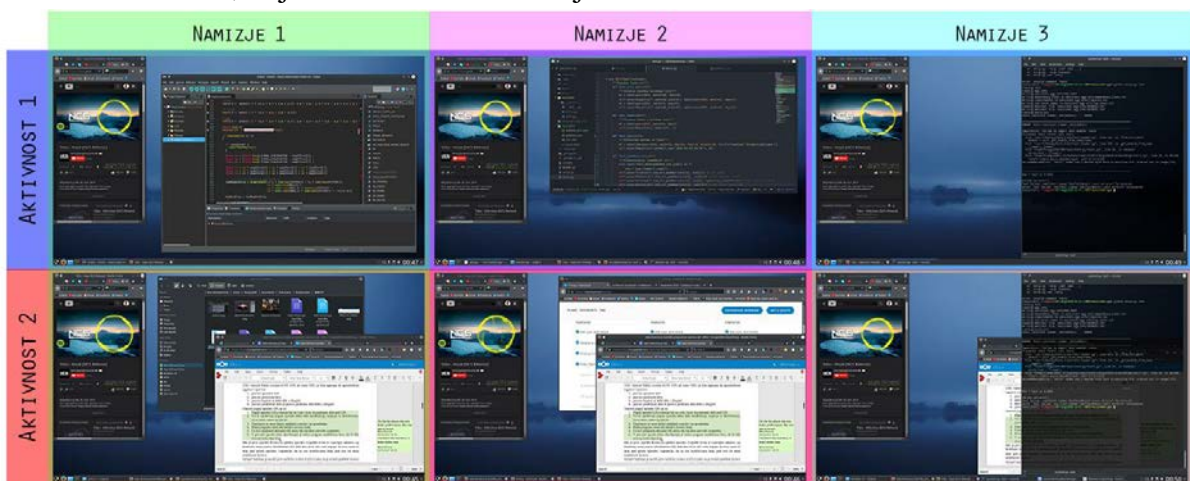
Zaradi ogromnega števila različnih distribucij Linuxa bi bilo nemogoče narediti praktično primerjavo, zato bo to poglavje bolj teoretično. Za praktične preizkuse v nadaljevanju naloge, pa sva izbrala Linux distribucijo Antergos, ki po najnem menenju najboljše zajame vse prednosti GNU/Linux sistema.

3.2.1. Namizje

Kot pri izbiri distribucije, imamo v Linux svetu veliko možnosti tudi pri izbiri namiznega okolja. Najdemo lahko vse od grafično zelo zahtevnih namizij z animacijami in prosojnosjo, do zelo osnovnih namizij, ki porabijo manj kot 100MB RAMa. Ker pa delava primerjavo z Windowsi, sva izbrala tiskega, ki jim je najbolj podoben.

Namizja Plasma projekta KDE je po izgledu najbolj podobno namizju Windows 7. Univerzalno znana orodna vrstica in začetni meni naredita to namizje odlično izbiro za tiste, navajene Windows sistemov. Plasma namizje podpira popolnoma vse funkcije Windows namizja - z uporabo tem ga je celo možno narediti skoraj popolnoma vizualno identičnega namizju Windows 7 in celo Windows 10 (z izjemo kartic v meniju Start). Poleg teh, pa ima Plasma še 2 večji funkcij, ki jih Windows 10 nima: univerzalne pripomočke (t.i. *Plasmoids*) in napredno organizacijo virtualnih namizij. Univerzalni pripomočki so majhni programčki, ki jih lahko postavimo na namizje ali pa v orodno vrstico. To so npr. koledar, ura, kalkulator, grafi sistemskih virov ipd.. Vsak element iz orodne vrstice lahko tako postavimo na namizje (in obratno). Tudi orodnih vrstic lahko imamo več. Na vsak rob vsakega zaslona lahko postavimo orodno vrstico in vsako posebej prilagajamo.

Čeprav je Microsoft po letih prošelj končno implementiral virtualna namizja v sistem Windows 10, so le-ta v Plasma namizju dosti bolj uporabna. Poleg neomejenega števila virtualnih namizij, Plasma ponuja tudi različne "Aktivnosti" (ang. *Activities*), ki imajo tudi ločeno postavitev namizja (glej sliko). Še pomembnejša, pa je možnost pripetja oken med namizji in celo med aktivnostmi. Tako lahko imamo okno, ki je vidno na vseh namizjih v eni aktivnosti in okno, ki je vidno na vseh namizjih na vseh aktivnostih.



3.2.2. Ukazna vrstica

Seveda ne moremo govoriti o GNU/Linux sistemih brez, da bi omenili ukazno vrstico.

Čeprav navadni uporabniki smartajo ukazno vrstico kot zastarelo, se njena moč pokaže pri pisanju skript in oddaljenem dostopu. GNU programi sledijo Unixovi filozofiji majhnih programov, ki se med seboj povezujejo. Tako lahko preko ukazne vrstice podatke prenašamo iz enega programa v drugega po standardiziranem načinu, med tem ko bi z grafičnim vmeskom morali podatke po vsakem koraku shraniti na disk in jih odpreti v naslednjem programu. To je odlično za avtomatizacijo, zelo uporabno pa je tudi za oddaljen nadzor strežnikov. Windows Server namreč uporablja grafični vmesnik, ki potrebuje zelo dobro internetno povezavo, ukaze pa lahko pošiljamo po po le nekaj kilobitni povezavi.

Windows sicer ima nekaj različnih skriptnih jezikov, a nobeden izmed njih ni sam po sebi tako uporaben kot Linuxov Bourne-kompatibilna ukazna lupina (najpogostejše Bash).

3.3. Grafični gonilniki

Gonilniki za strojno opremo so najpomembnejši del operacijske sistema, zato je njihova stabilnost izjemno pomembna za uporabnost operacijskega sistema. Grafični gonilniki na Linux sistemih so že od nekdaj znani po njihovi nestabilnosti, kar je tudi eden izmed večjih razlogov za majhno število domačih uporabnikov Linuxa.

V zadnjih letih, pa se je stanje Linux grafike izboljšalo dovolj, da je celo podjetje Valve Software svojo platformo za računalniške igre Steam "portalo"¹¹ na Linux in konec leta 2013 objavilo Linux distribucijo SteamOS. Čeprav le-ta ni doživela uspeha, se je število računalniških iger, ki delujejo na Linuxu hitro povečalo - s tem, pa se je pospešil tudi razvoj odprtokodnih Linux gonilnikov.

Pri obeh glavnih proizvajalcih grafičnih kartic (AMD in NVIDIA) najdemo več različnih grafičnih gonilnikov - nekateri so popolnoma odprtokodni, drugi popolnoma lastniški, tretji pa nekaj vmes.

Lastniki **NVIDIA** kartic lahko trenutno izbirajo med dvema grafičnima gonilnikoma.

Odprtokodni *Nouveau* gonilnik uporablja odprtokodno grafično knjižnico *Mesa*. Te gonilnike so v celoti napisali prostovoljci s pomočjo obratnega inženiringa - podjetje NVIDIA je namreč javno objavilo, da ne podpirajo odprtokodnih gonilnikov, objavljajo pa le zelo omejeno tehniško dokumentacijo.

Samo podjetje NVIDIA pa ponuja lastniški gonilnik (včasih imenovan *GeForce*), ki namesto knjižnice *Mesa* uporablja svojo implementacijo OpenGL. Ta sicer deluje hitreje kot odprtokodni, a zaradi skoraj neobstoječe dokumentacije pogosto povzroča težave.

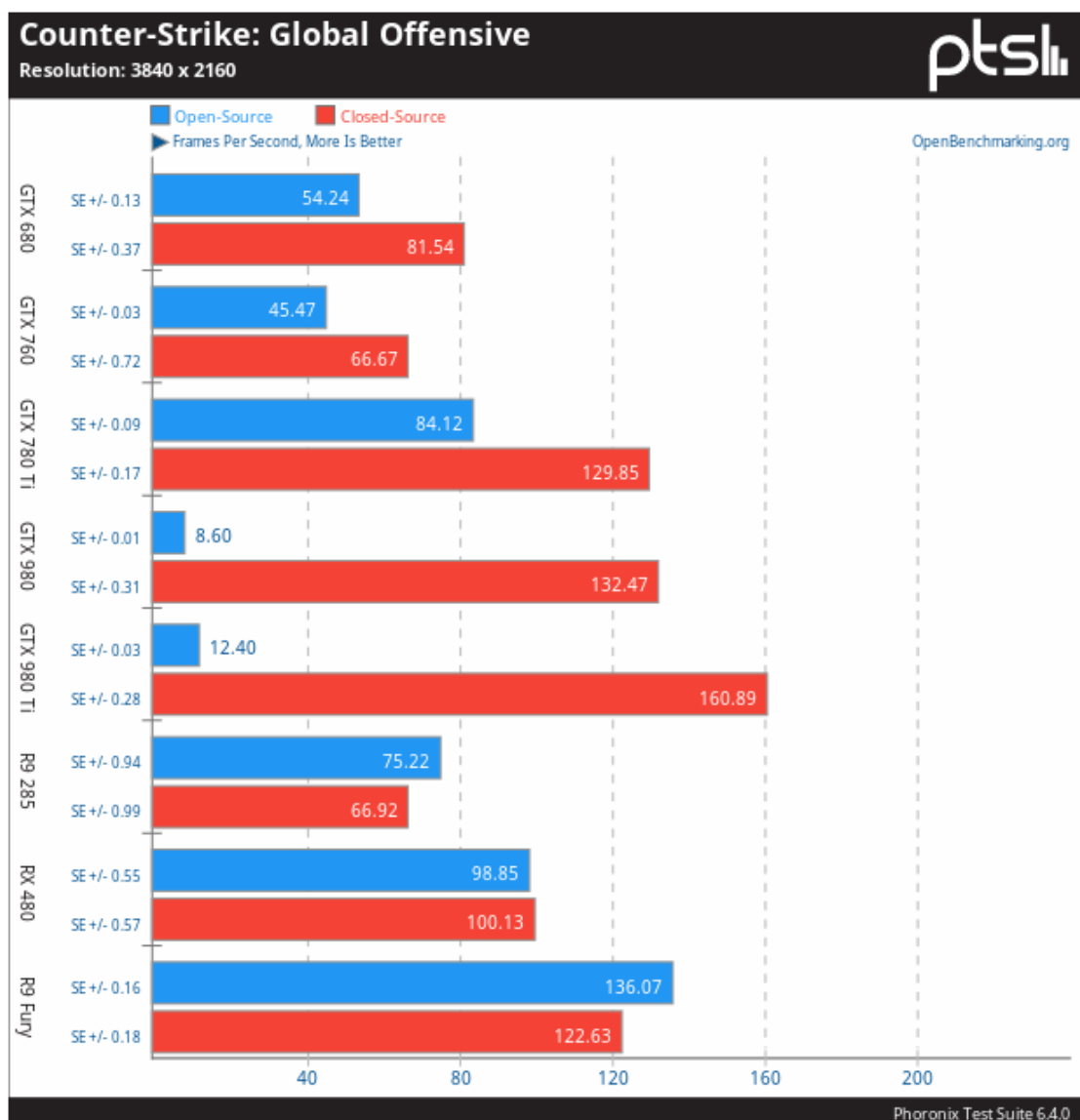
Pri **AMD** karticah je najnovejši gonilnik *AMDGPU*, ki je pod odprtokodno licenco. Ta uporablja Mesa grafični API, za zahtevnejše grafične operacije pa se povezuje s knjižnico *AMDGPU-PRO*, ki jo AMD objavlja pod lastniško licenco. Ker pa ti gonilniki ne podpirajo starejših AMD in ATI kartic sta v uporabi še vedno sicer nepodprta gonilnika *Radeon* (odprtokodni) in *AMD Catalyst* (lastniški).

Podjetje **Intel** pa je gonilnike za svoje integrirane grafične čipe objavilo pod odprtokodno licenco.

Najin namen je bil preizkusiti vse različne Linux grafične gonilnike za oba proizvajalca in primerjati odprtokodne z zaprtokodnimi. Ker pa nisva uspela dobiti dovolj različne strojne opreme, bi bili rezultati najinih preizkusov neuporabni. Na srečo je Michael Larabel s portala Phoronix.com vse teste opravil na dosti bolj raznoliki strojni opremi kot ta, do katere imava dostop midva.

Spodnji graf predstavlja FPS v igri Counter-Strike: Global Offensive pri 2K resoluciji. Z modro so označeni odprtokodni gonilniki, z rdečo pa lastniški.

¹¹ Prilagoditi program za delovanje na drugo platformi



Slika 1 Primerjava odprtokodnih in zaprtokodnih grafičnih gonilnikov

12

Ta graf nam o stanju NVIDIA grafike pove 2 pomembni stvari. Prva je, da so odprtokodni gonilniki še vedno opazno slabši od uradnih. Čeprav se razlika z vsako posodobitvijo zmanjša, je ta razlika zelo pomembna pri grafično bolj zahtevnih igrah, kjer se z odprtokodnimi gonilniki številke bogosto spustijo pod 30 FPS. Prav tako nam graf pokaže, da je razlika zelo odvisna od serije grafične kartice, ki jo uporabljamo. Ker so odprtokodni gonilniki v veliki večini napisani s pomočjo obratnega inženiringa, so starejše kartice - specifično GPU arhitektura "Kepler" - bolj podprte.

Pri podjetju AMD, pa se številke obrnejo. Tukaj je namreč odprtokodni gonilnik včasih celo hitrejši od lastniškega, kar sicer ni presenetljivo, saj je AMD eden izmed glavnih podpornikov projekta Mesa.

¹² <https://openbenchmarking.org/prospect/1701073-RI-OPENCLOSE69/bef2b9a6c31f89a23ecbf24edc056fb8597ffd8a>

4. Multimedija

4.1. Obdelava fotografij

Med programi za obdelavo fotografij poznamo dve glavni vrsti: programe za organizacijo in osnovno obdelavo ter samostojne programe za bolj napredno obdelavo. Vsako izmed teh kategorij si bomo ogledali posebej.

4.1.1. Upravljanje fotografij

Opravljaniki fotografij niso namenjeni natanči in zapleteni obdelavi fotografij, zato se bova pri primerjanju teh osredotočila bolj na uporabniško izkušnjo, stabilnost in uporabniški vmesnik, kot na samo obdelavo fotografij.

Najverjetneje najbolj znana programa za organizacijo fotografij sta Adobov Photoshop Lightroom in Googlova Picasa (konec podpore od začetka 2016). Čeprav oba programa opravljata podobno funkcijo, pa se med seboj razlikujeta v vrsti uporabnikov.

Picasa - digiKam

Google Picasa je za razliko od Lightrooma namenjena lažjim uporabnikom. Je popolnoma brezplačna, a podpira le osnovne popravke fotografij in je bolj osredotočena na organizacijo. Preden začneva s primerjavo je pomembno poudariti, da je Google podporo za Picaso popolnoma ukinil, čeprav jo mnogi še vedno uporabljajo. Po uradni izjavi so Picaso ukiniteli, saj se jim nadaljno razvijanje brezplačnega programa ni izplačalo in so se želeli bolj osredotočiti na spletne storitve. Tukaj bi rada omenila da bi, če bi podjetje Google izvorno kodo Picasa vsaj po ukinitvi objavilo pod odprtokodno licenco, skupnost uporabnikov lahko program razvijala naprej. Ker pa tega niso naredili, bi bila kakršna koli sprememba programa ne samo izjemno težka, ampak tudi kazniva. Tako bo še en dober program čez čas izginil, čeprav bi ga lahko prostovoljci razvijali dalje.

Picasi zelo podobni so odprtokodni programi *Shotwell*, *gThumb* in *digiKam*. Midva si bova podrobneje ogledala digiKam, saj za razliko od drugih dveh deluje na vseh platformah in ima zraven osnovnih funkcij tudi nekaj naprednejših.

Program digiKam razvijajo prostovoljci projekta KDE, ki med drugim razvija tudi popularno Linux namizje *Plasma* in program za risanje *Krita*.

Tako kot Picasa, je digiKam popolnoma brezplačen, zraven OSX in Windows pa podpira tudi GNU/Linux operacijske sisteme.

Najin preizkus sva začela z uvažanjem dobrih 2000 fotografij v oba programa. Fotografije so bile iz različnih fotoaparatorov in shranjene v različnih formatih. Picasa je fotografije uvozila v komaj 30 sekundah, med tem ko je digiKam za potreboval skoraj 2 minuti. Čeprav se to sprva zdi kot prednost Picasa sva kmalu ugotovila, da je za to kriva dosti boljša podpora programa digiKam za RAW format fotografij. Picasa je namreč te datoteke obravnavala kot navadne fotografije in njih uvozila narobe, med tem ko je digiKam vse RAW fotografije pravilno prepoznal in pretvoril - zato je uvoz trajal dalje.



Slika 2 Napaka pri uvozu RAW fotografij v Picasi. Zgoraj Picasa, spodaj digiKam. (RAW in JPEG slika)

Ko so bile vse fotografije uvožene, sva preizkusila organizacijske funkcije obeh programov. Oba podpirata sortiranje in filtriranje fotografij po času nastanka, geolokaciji, oznakah in obrazih na fotografiji (slednje deluje v Picasi dosti boljše). Med tem ko so v Picasi albumi popolnoma ločeni od map, digiKam obravnava vsako mapo kot ločen album (in obratno). Združevanje fotografij iz različnih map je tako v digiKamu možno le z oznakami, kar je sicer bolj uporabno, a za uporabnike Picase precej nenavadno. Iskanje fotografij je v Picasi zelo enostavno, digiKam pa zraven tega ponuja tudi naprednejše iskanje po metapodatkih fotografij (model fotoaparata, leča, ISO...). Prav tako podpira digiKam ocene od 0 do 5 zvezdic, med tem ko je v Picasi za fotografijo zvezdico mogoče le vklopiti ali izklopiti. Urejanja fotografij se programa lotita na zelo podoben način. Fotografij na disku namreč ne spreminjata, saj vse spremembe hranita v svoji podatkovni bazi. Picasa za to uporablja picasa.ini datoteke, ki jih shrani v vsako mapo s fotografijami, digiKam pa uporablja SQLite datoteke, shranjene na enem mestu, eksperimentalno pa podpira tudi uporabo MySQL strežnika, ki bi večim uporabnikom omogočal sinhronizacijo sprememb. Za razliko od Picase, digiKam urejevalnik fotografij odpre v ločenem oknu od mrežnega pogleda, kar je za uporabnike z večimi zasloni zelo uporabno. Oba programa podpirata le urejanje celotne fotografije (brez mask) in imata zelo podobna orodja. Picasa ima manj orodij, a so ta enostavnejša za uporabo saj se spremembe posodablajo hitreje, med tem ko ima digiKam nekaj orodij, ki jih Picasa nima. Med te spadajo dodajanje napisov, odstranjevanje vinjete, uravnavanje barvnih kanalov, popravljanje popačenja leče, odstranjevanje šuma in dosti boljše ostrenje fotografij.

Lightroom - darktable

Program Adobe Photoshop Lightroom je dosti bolj napreden od prej opisanih programov in ga uporabljajo večinoma profesionalni fotografi. Največja razlika med programi kot je digiKam je, da Lightroom omogoča ustvarjanje mask, s katerimi lahko dosežemo dosti bolj natančno obdelavo. Prav tako ima pravo podporo za RAW datoteke ter omogoča nadroz fotoaparata preko USB povezave.

Odprihodni program darktable je nastal kot direktna alternativa za Adobe Lightroom, kar je zelo razvidno tudi iz grafičnega vmesnika. Tako kot Lightroom, je darktable namenjen profesionalcem, saj veliko število nastavitev ni prijazno navadnim uporabnikom. Podobno kot Lightroom, ima darktable 2 glavna pogleda, t.i. *Lighttable* za pregledovanje fotografij v mreži in *Darkroom* za obdelavo fotografij (v Lightroomu se imenujeta *Library* in *Develop*).

Med tem ko Lightroom podpira sisteme Windows in OS X, lahko Darktable uporabljamo na OS X in GNU/Linux sistemih, uradna Windows različica pa trenutno ne obstaja (čeprav je podpora za Windows načrtovana že nekaj časa).

Za preizkus sva v obeh programih odprla fotografijo v RAW formatu, ki je bila narejena v zelo slabi svetlobi. Fotografijo sva standardno obdelala v obeh programih.

Čeprav sta vmesnika na prvi pogled skoraj identična, so majhne razlike tiste, ki naredijo Lightroom dosti lažji za uporabo. Možnost prilagajanja krivulj kar z klikom na sliko, filter "dehaze" in dosti boljše osvetljevanje senc, je le nekaj primerov, v katerih Lightroom gladko premaga darktable. A to ne pomeni, da darktable ni dober program. Je še vedno zelo dober RAW urejevalnik, še posebej glede na to, da je popolnoma brezplačen.

4.1.2. Urejanje fotografij: Adobe Photoshop - GIMP

Ko pride do obdelave fotografij, je program Adobe Photoshop postal tako prepoznaven, da ga nekateri uporabljajo kar kot glagol. Photoshop ni samo eno izmed najboljših orodij za zelo podrobno obdelavo slik, vsebuje tudi osnovna orodja za animacijo, podpira vektorske oblike in celo 3D objekte. A večina uporabnikov vsega tega ne potrebuje, zato jim je težko upravičiti 12€ mesečno¹³ naročnino za osnovno uporabo (npr. kopiranje elementov iz ene slike v drugo).

Večino napredne obdelave fotografij pa lahko prav tako dobro opravi odprtokodni program GIMP, ki za razliko vsebuje le orodja za obdelavo slik in s tem zelo prihrani na velikosti in porabi sistemskih virov.

Pri delu z obema programoma sva opazila ogromno razlik, večinoma prednosti programa Photoshop. Prva stvar, ki je izstopala je, da GIMP nima nobene podpore za nedestruktivno obdelavo slik. Vse spremembe, ki jih naredimo, se uveljavijo takoj, med tem ko jih v Photoshopu z uporabo pametnih filtrov lahko kasneje spreminjamo. Še ena velika slabost programa GIMP je, da v njem ni mogoče istočasno urejati večih plasti. Večina orodij v GIMPU je bila ročnih, tista z malo avtomatizacije, pa so v Photoshopu delovala dosti bolje. Prav tako GIMP nima vgrajenih funkcij za združevanje slik v panorame in HDR. Sva pa na koncu v obeh programih dosegla enako dober rezultat, le da je v GIMPU trajalo precej dlje.

4.2. Montaža videa: Premiere Pro - Kdenlive

Adobe Premiere Pro je skoraj to prepoznaven kot Photoshop, zato je očitna izbira za predstavnika zaprtokodnih programov. Čeprav je Kdenlive, najin izbran odprtokoden program, po funkcijah bolj podoben programu Premiere Elements (Premiere Pro vsebuje

¹³ Skupaj s programom Photoshop Lightroom

dosti bolj napredna orodja za delo z zvokom in barvami), sva ga primerjala s profesionalno različico saj jo oba bolje poznavata.

V obeh programih sva zmonitrirala kratek prispevek, podoben televizijskim poročilom.

Montažo sva začela s sinhronizacijo zvoka iz zunanjega snemalnika z sliko iz kamere. Tukaj se je pokazala prva večja razlika: Kdenlive pri uvozu video in zvok iz ene datoteke združi v en objekt v časovnici, med tem ko ju Premiere šteje kot ločena (čeprav povezana) objekta. To je bila prednost programa Kdenlive, saj v tem primeru zvoka iz kamere nisva potrebovala in je tako zasedel manj prostora v časovnici. Pokazala pa se je tudi ena prednost programa Premiere Pro, v katorem sva s hitrim pritiskom tipke časovnico povečala preko celotnega okna in tako lažje videla obliko zvoka.

Ker je bil video posnet na zelenem ozadju sva ozadje zelo enostavno odstranila v obeh programih. V programu Premiere Pro sva to lahko dosegla z enim samim efektom imenovanim *Ultra Key*, v Kdenlive pa sva morala uporabiti še mešalnik barv, da sva odstranila zelen sij okrog osebe, ki ga je povzročila odbita svetloba.

Preko glavnega posnetka sva dodala še nekaj statičnih "b-roll" posnetkov, ki sva jih premikala s pomočjo *Transform* efektov. Efekta v obeh programih delujeta skoraj identično, čeprav je bilo delo z njiv dosti lažje v Premieru, ki sliko povečuje tako, da sredina ostane na istem mestu in tako prihrani matematike.

Dodajanje napisov in osnovno popravljanje barve je bilo enostavno v obeh programih, ponovno pa je mala razlika v Premieru zelo pomagala. To je funkcija *Workspaces*, ki omogoča preklap med večimi v naprej nastavljenimi postavitvami orodij. Tako sva lahko v Premieru vsa orodja za spreminjanje barv istočasno odprla in jih zaprla, ko sva z njimi končala, kar je delo precej olajšalo. Prav tako so posebne plasti (t.i. *Adjustment layers*) precej pospešile delo z barvami, saj sva lahko enak popravek uveljavila na več objektov hkrati.

Premiere vs Kdenlive

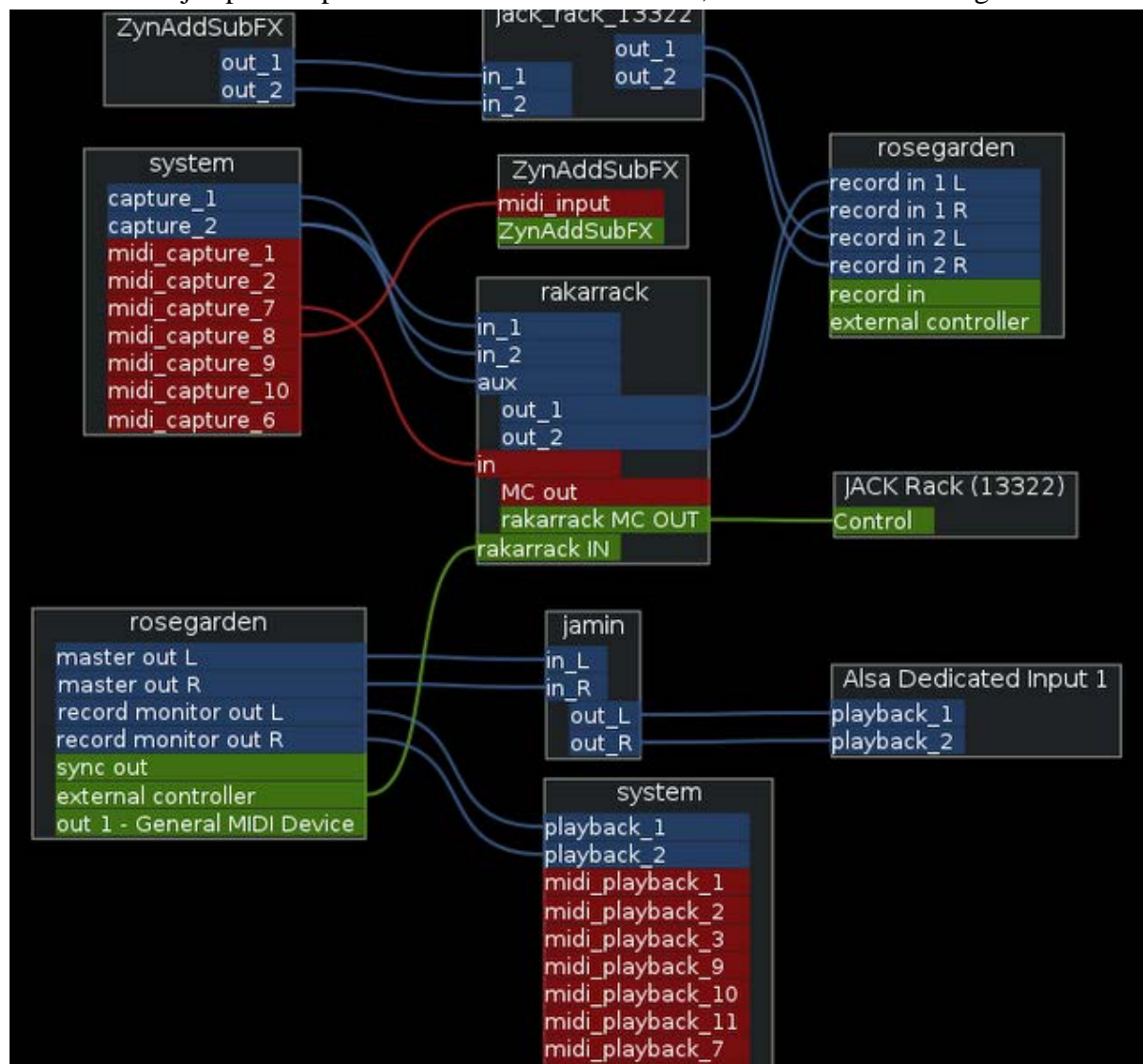
4.3. Obdelava zvoka

Tržišče s programsko opremo za obdelavo zvoka je najverjetneje najbolj raznoliko od vseh kategorij, ki sva jih raziskala a na koncu sva kot predstavnika zaprtokodnih programov izbrala še en Adobov program - Adobe Audition, ki v enem programu združuje večtračni način (*Multitrack*) za ne-destruktivno obdelavo in tradicionalni enotračni urejevalnik (*Waveform*). Mnogi to združitev hvalijo, saj naj bi omogočala hitrejše delo z obema načinoma hkrati, prav to pa sva želela primerjati z bolj tradicionalnim ločenim pristopom najinih odprtokodnih izbir: *Ardour* za snemanje in delo z večimi linijami in *Audacity* za natančnejšo obdelavo.

Jack Audio Connection Kit

Preden začneva s primerjavo programov je pomembno omeniti še en odprtokodni projekt, kateremu se težko približa katera izmed lastniških alternativ. To je zvočni sistem Jack - sistemska knjižnica, ki omogoča izjemno natančno usmerjanje zvoka (tudi MIDI podatkov) med programi z minimalno zakasnitvijo. Program, ki podpira sistem Jack lahko v sistemu registrira več zvočnih vhodov in izhodov, katere lahko med seboj povezujemo. Tako lahko na primer zvočni izhod električne kitare povežemo skozi nekaj efektov, od tam povežemo

slušalke za nadzor in program za vizualizacijo, MIDI klaviaturo povežemo v sintetizator in program za avtomatski zapis not, nato pa obe zvočni liniji povežemo v večtračni snemalnik. Tako in še bolj zapletene povezave lahko ustvarimo brez, da bi se dotaknili enega kabla.



Slika 3 Program Patchage za povezovanje JACK vhodov in izhodov.

Da pa je sistem Jack lahko dovolj hiter, omogoča spremenjena različica Linux kernela, z oznako *lowlatency*. Namen te različice je zmanjšati zakasnitev vhodov in izhodov, kar je tudi razlog, da se sistem Jack ni bolj razvil na lastniških operacijskih sistemih, ki takšnih prilagoditev nimajo.

Adobe Audition Multitrack - Ardour

Ker je avtor programa Ardour tudi glavni razvijalec sistema Jack, je v Ardour seveda možno povezati izhod vsake aplikacije, ki Jack uporablja. To, zraven odlične podpore za vse vrste audio plug-inov, omogoča tudi uporabo zunanjih efektov, ki niso na voljo kot dodatki. Zaradi časa žal nisva uspela programov preizkusiti s studijsko snemalno opremo, zato sva za primerjavo uporabila večtračni posnetek šolske novoletne prireditve. 14 zvočnih linij sva v vsakem programu najprej sinhronizirala z videjem, jih umestila v 2D prostor, nekaterim dodala osnovno odstranjevanje šuma in izenačevalnike, nato pa celotno stvar izvozila kot video

datoteko. Že pri uvozu videa se je pokazala majhna prednost Adobe Auditiona, ki sicer velja za večino Adobovih CC programov. To je t.i. *Dynamic Link*, ki je omogočil, da sva v Audition uvozila kar celo .prproj datoteka brez, da bi jo najprej pretvorila v video, kot sva storila za Ardour. To je bilo možno, ker je bil video, ki sva ga uporabljala zmontiran v programu Premiere Pro in je tako bolj prednost celotnega Adobe CC ekosistema, ne samega Auditiona. Oba programa imata zelo podobne vmesnike, zato pri delu ni bilo velikih razlik. Navigacija med linijami je bila v Adobe Auditionu enostavnejša, saj ima t.i. “minimap scroll bar” tudi za vertikalno premikanje. Prav tako je vsaka linija v Ardouru zavzela več vertikalnega prostora, saj je črta za spremembo glasnosti ločena od vrstice z zvokom, med tem ko je v Auditionu postavljena preko nje. Tudi privzete efekte kot je izenačevalnik je v Auditionu lažje uporabljati čeprav je za Ardour na voljo nekaj boljših, ki pa niso samodejno nameščeni. Vse ostalo je bilo v obeh programih enako.

Adobe Audition Waveform - Audacity

Za destruktivno urejanje, pa sva preizkusila odprtokodni program Audacity. Delo z obema programoma je bilo zelo podobno. Kot sva opazila tudi pri drugih Adobe programih, je uporabniški vmesnik Auditiona dosto bolj dodelan in enostaven za uporabo. Med tem, ko je bilo v Audacityju nekatere osnovne funkcije iskati v meniju, so je v Auditionu ob miški pokazale v plavajočem “HUD” meniju. Tudi “waterfall” graf je bil v Auditionu lažji za uporabo, je pa imel Audacity eno veliko prednost: podpira namreč destruktivno urejanje z večimi linijami zvoka, med tem ko je v Auditionu to možno le v destruktivnem načinu. Razen tega, pa med programoma ni bilo večjih razlik.

5. Pisarna

Microsoft je dolgo imel skoraj čist monopol pisarniških programov, zato je tudi predstavnik lastniške programske opreme v tej primerjavi. Na srečo dandanes obstajajo odprtokodne alternative, ki postajajo vedno bolj popularne, čeprav imajo zelo zapleteno zgodovino. Odprtokodne alternative bo predstavljala LibreOffice, ki smo jo izbrali delno zaradi zgodovine, delno pa zaradi prednosti, ki mu jih ponuja licenca.

StarOffice je bil prvi pomemben korak proti odprtokodnim pisarniškim programom, čeprav je bila lastniška programska oprema. Leta 1998 je StarDivision začel ponujati StarOffice zastoj. Eno leto za tem je Sun Microsystems pridobil StarOffice za \$73.5 milijonov. "Glavni razlog da je Sun Microsystems kupil StarDivision je bil, da je Sun imel približno 42000 zaposlenih. Skoraj vsak je imel računalnik z Unixom in laptop z Windowsi. Ceneje je bilo kupiti podjetje, ki lahko naredi pisarno kot kupiti 42000 licenc od Microfosta" (Simon Phipps, 2010, str. 2). Sun Microsystems je izvorno kodo StarOffice-a izdal 13. Oktobra 2000. Novi program po imenu OpenOffice.org je izšel 1. Maja 2002 pod odprtokodno licenco. Hitro je postal privzeti program na Linux sistemih in postal tekmelec Microsoftovemu officu (Del Izraelske vlade je OpenOffice začel uporabljati že leta 2003¹⁴). Zakaj si torej nisva izbrala OpenOffice.org kot najin predstavnik za pisarne? Leta 2010 je Oracle kupil Sun Microsystems in bistveno zmanjšal število razvijalcev na projektu. Večina razvijalcev je opustila projekt in začela razvijati LibreOffice, ki je na večini Linux sistemov zamenjal OpenOffice. Oracle je OpenOffice podaril Apache Software Foundationu, ki ga je spet licenciral pod odprtokodno licenco. Zaradi licence pod katero je OpenOffice, lahko LibreOffice uvede vse novosti OpenOfficea, ampak obratno ni možno.

Najpomembnejša razlika med Microsoft Office in LibreOffice je za mnoge cena. Osnovni paket Microsoft Office stane \$149.99, profesionalen paket pa \$399.99. Možno je tudi letno plačilo \$69.99. LibreOffice pa je čisto zastoj, zato ga tudi uporablja vedno več podjetij in uradov.

Mesta, ki so že začela uporabljati LibreOffice vključujejo: Toulouse, ki naj bi prihranilo milijon evrov¹⁵ z implementacijo LibreOfficea; Nantes¹⁶, ki je prihranilo 1,7 milijonov evrov; Copenhagen¹⁷ in München¹⁸.

Druga velika razlika je v izgledu, saj imata LibreOffice in Microsoft Office zelo različne filozofije glede designa. MS Office od različice 2007 naprej uporablja t.i. "Ribbon" meni z

¹⁴ http://www.theregister.co.uk/2003/11/24/ms_scorns_israeli_openoffice_defection/

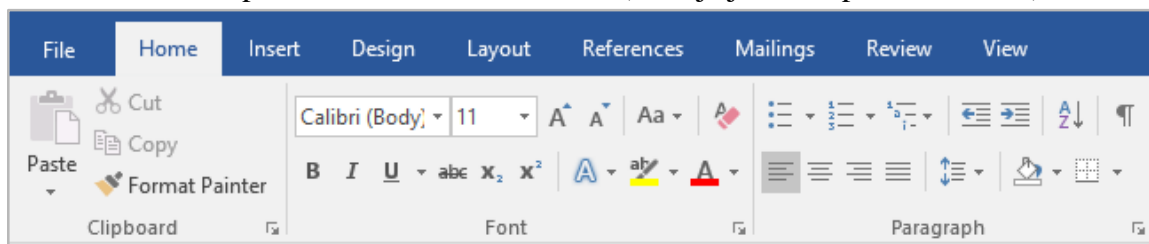
¹⁵ <https://joinup.ec.europa.eu/community/osor/news/moving-libreoffice-saves-toulouse-1-million>

¹⁶ <https://joinup.ec.europa.eu/community/osor/news/nantes-%E2%80%9CChange-management-key-switch-free-software>

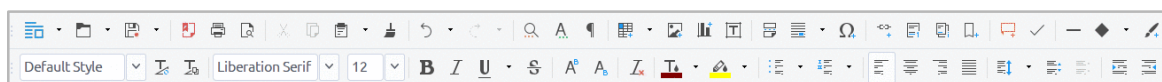
¹⁷ <https://joinup.ec.europa.eu/news/dk-25000-hospital-staff-copenhagen-region-use-open-source-office-suite>

¹⁸ <http://www.itworld.com/article/2719059/it-management/munich-shifts-to-libreoffice.html>

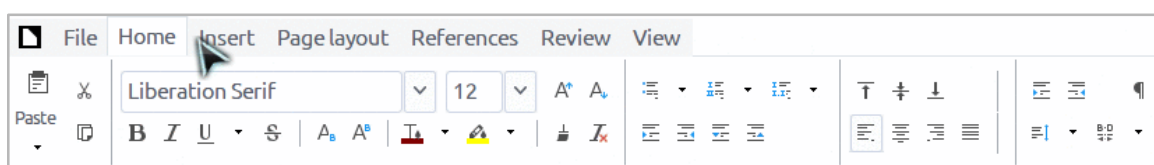
zavihki, med tem ko lahko uporabniki paketa LibreOffice izbirajo med tradicionalno orodno vrstico in ribbonu podobno "Notebook" vrstico (slednja je šele v preizkusni fazi).



Slika 4 Microsoft Office "Ribbon" vmesnik



Slika 5 LibreOffice klasični vmesnik



Slika 6 LibreOffice "Notebook" vmesnik

Ker pa je izbira uporabniškega vmesnika stvar okusa in ne vpliva na uporabnost programa, tega ne bova štela v končno oceno.

5.1. Besedilo: Microsoft Word - LibreOffice Writer

V veliko stvareh sta si Word in Writer zelo podobna, je pa nekaj razlik, ki jih moramo upoštevati pri izbiri programa. Word ima večjo podporo za skupno urejanje dokumentov, ki je za nekatere naloge bistvenega pomena. Več avtorjem omogoča istočasno urejanje dokumentov preko storitev OneDrive ali Sharepoint (to zahteva naročnino na Office365). To postane lažje s takojšnjim sporočanjem, ki je v Wordu podprto, v Writerju pa ne. Word omogoča, da dele dokumenta skriješ in odkriješ, ko jih potrebuješ. To lahko pomaga pri organizaciji dokumentov in splošnem poteku dela.

Mogoče največja prednost Writerja je to, da dobro podpira daljše dokumente, s katerimi ima Word včasih težave. Podpira tudi uporabo *master dokumentov*, ki močno olajšajo delo z daljšimi dokumenti. Writer je združljiv z nekaj več vrst dokumentov, s katerimi Word ni. Najpomembnejše zmožnosti, ki jih ima Writer in ne Word so uvažanje Apple Pages in DocBook dokumentov in izvažanje v MediaWiki, XHTML in celo .jpg in .png dokumente. Ker teh vrst dokumentov Word ne podpira je v določenih situacijah (na primer urejanje wiki strani) Writer dosti bolj uporaben. Zelo uporaben je tudi, če mora dokument vsebovati kompleksne enačbe v tabelah saj Word dovoljuje samo osnovne aritmetične enačbe. Preverjanje slovnice je v Wordu možno v 21. jezikih, v Writerju pa le 4. Zaradi odprtokodne licence Libreoffice je nastala velika količina razširitev, ki podpirajo več jezikov.

5.2. Preglednice: Microsoft Excel - LibreOffice Calc

Tudi tukaj ima odprtokodni program kar nekaj prednosti. Prva stvar, ki jo lahko pogledamo je število funkcij. Excel jih ima 468, od tega 14 edinstvenih; Calc pa jih ima 500, od tega 24 edinstvenih. Calc nam ponuja dosti možnosti, ki jih Excel nima, nekatere od teh so: primerjanje dveh preglednic, preverjanje črkovanja medtem ko pišeš in izvažanje v XHTML. Vse te zmožnosti so uporabne, a Excel je boljši v analiziranju podatkov, večinoma zaradi *PowerPivota*, ki je napredna tehnologija uporabljena za hitrejšo in bolj obsežno analizo podatkov znotraj *Pivot* preglednic. Omogoča tudi uporabo zunanjih virov podatkov. Excel ima več vrst tabel in podpira interaktivne diagrame. Pri hitrosti Excel tudi zmaga, saj dosti funkcij uporablja večjedrno procesiranje.

5.3 Predstavitve: Microsoft Powerpoint - LibreOffice Impress

Impress in Powerpoint sta si tako podobna, da ni dosti stvari za primerjati. Kot ponavadi je odprtokodni bolj združljiv z različnimi tipi dokumentov. Lahko izvaža v XHTML, HTML, SWF, SVG in EPS. Impress je zmožen vstavljanja 3d modelov v predstavitev z uporabo GIF, DAE ali KMZ datotek. Večjo podporo ima tudi za daljinsko vodenje z uporabo tablic (ki jih Powerpoint sploh ne podpira) in pametnih telefonov.

Vseeno pa ga v nekaterih stvareh Powerpoint preseže. Večinoma v raznih animacijah, ima pa tudi možnost predstavitve preko spleta, kar pa ne deluje pretirano dobro, zato se pogosteje za oddaljene predstavitve uporablja deljenje namizja, ki seveda deluje tudi s programom Impress.

6. Spletne storitve

Z premikom vedno več programov v oblak, je varnost oblačnih storitev vedno bolj pomembna. Ker pa je večina teh zaprtokodnih in nameščenih na strežniku ponudnika, ne moremo vedeti kje naše datoteke so in kaj se z njimi dogaja. Kot alternativo tem storitvam, bova predstavila 2 rešitvi, ki ju lahko namestimo na svoj strežnik in sta popolnoma odprtokodni.

6.1. Shramba datotek

Podjetje Dropbox je leta 2007 začelo trend premika datotek v oblak in do danes ostaja eden izmed največjih ponudnikov shrambe v oblaku. Dropbox, Google Drive, Microsoft OneDrive in drugi ponudniki tovrstnih storitev skoraj vsi uporabljajo lastniško programsko opremo in imajo eno veliko slabost: vse datoteke so shranjene na njihovih strežnikih.

Uporabnikom vrniti nadzor nad njihovimi datotekami, je misija projekta ownCloud, je trenutno najbolj razvit odprtokodni projekt na področju shrambe v oblaku. Zraven nalaganja in prenašanja datotek, ima ownCloud sistem aplikacij, s katerimi lahko datoteke urejamo kar v brskalniku, omogočajo pa tudi druge stvari, kot so spletni koledar in celo videoklepet. Zaradi nestrinjanj med razvijalci, pa je v zadnjem letu nastal fork imenovan Nextcloud, ki obljublja boljšo podporo in stabilnost. Za svoj preizkus sva uporabila tega, saj je namestitev aplikacije CODE (v naslednjem poglavju) dosti enostavnejša kot pri ownCloudu.

Do ownCloud ali Nextcloud oblaka lahko dostopamo na 3 načine. Z vseh naprav ga lahko odpremo preko spletnega brskalnika, kjer imamo dostop do čisto vseh funkcij programa,

vključno z nameščenimi aplikacijami. Uporabniki Android in iOS naprav lahko do oblaka dostopajo tudi preko aplikacije, ki omogoča upravljanje datotek v oblaku in takojšnje nalaganje zajetih fotografij. Namizni uporabniki pa imajo na voljo tudi aplikacijo za sinhronizacijo map, ki v določene lokalne mape sinhronizira z strežnikom. Enako kot na Dropboxu, lahko datoteke in mape iz našega oblaka delimo z drugimi uporabniki, ali pa s spletno povezavo. Posebna pa je funkcija “Federated cloud”, ki omogoča deljenje datotek med uporabniki različnih ownCloud strežnikov.

6.2. Pisarna v “oblaku”

S premikom datotek v oblak pa zelo pogosto postajo kolaborativno urejanje dokumentov prek spleta. Takšne storitve omogočajo večim uporabnikom da istočasno urejajo dokument in v živo vidijo spremembe, ki jih delajo drugi. Na popularnosti so pridobile z začetkom Googlove storitve *Google Docs*, kateri je kmalu sledil tudi Microsoftov *Office Online*. Čeprav dolgo časa ni bilo nobenih odprtokodnih alternativ, jih te v zadnjih letih le dohitevajo. Najbolj razvit izmed teh je sigurno pisarniški paket Onlyoffice ki poleg odličnih urejevalnikov za vse tri osnovne dokumente (besedilo, predstavitev in preglednice) ponuja še e-pošto, CRM in PM module in še več poslovnih orodij, zato je bolj primerljiv z kombinacijo Microsoftovih Office365 E1 in Dynamics365. Za takšno storitev bi Microsoft zaračunal preračunanih 12€ na mesec za vsakega uporabnika, med tem ko se podobna¹⁹ gostovana ponudba (možna je tudi namestitev na lastnih strežnikih, ki je brezplačna) Onlyoffice začne pri slabih 5€ na uporabnika na mesec, z dodajanjem večih uporabnikov pa se cena še znižuje. Ker pa se v tej nalogi bolj osredotočava na posameznike in manjše organizacije, si bova podrobneje ogledala LibreOffice Online. LibreOffice Online je projekt organizacije TDF (*The Document Foundation*), ki omogoča uporabo LibreOffice preko spletnega vmesnika. Ta je sicer še vedno v razvoju, trenutno pa obstajajo 3 uporabne različice: Uradno preizkusno izdajo je možno dobiti v obliki Docker containerja `libreoffice/online`, podjetje Collabora, ki med drugim nudi tudi profesionalno LibreOffice podporo, pa s svojima različicama omogoča integracijo z zunanjimi shrambami datotek. Ponujajo 2 različici: *CODE* (*Collabora Online Development Edition*), ki je namenjena za uporabo z ownCloud, Pydio ali podobnimi sistemi in Collabora Online, za katero nudijo profesionalno podporo in zahteva plačljivo različico prej omenjenih datotečnih shramb.

Zaradi zelo enostavne namestitve sva se odločila preizkusiti CODE v kombinaciji s strežnikom Nextcloud (podpora za ownCloud je malo slabša).

Program sva preizkusila kar z to raziskovalno nalogo. Nekaj časa sva jo pisala v Google Drivu, nato pa sva datoteko premaknila na Nextcloud in nadaljevala tam. Oba urejevalnika sta si po izboru funkcij zelo podobna, čeprav se Googlovi storitvi pozna več let razvoja. Omeniti je vredno, da CODE deluje tudi v mobilnih brskalnikih in nima mobilne aplikacije. Google Docs pa v mobilnih brskalnikih ne deluje, kar pa nadoknadi z zelo dobro izdelano mobilno aplikacijo.

¹⁹ Onlyoffice sicer ponuja bistveno manj prostora za shranjevanje datotek, ki pa se ga da nadgraditi.

7. Ugotovitve

Na začetku naloge sva si zastavila vprašanje: “Kako bistvena je lastniška programska oprema?”. In to bova v tem poglavju poskusila odgovoriti, za vsako področje posebej.

Operacijski sistemi

Na področju operacijskih sistemov, sva ugotovila, da je sistem **GNU/Linux** odlična alternativa sistemu Windows, a je še vedno malo omejen s podporo strojne opreme. Uporabniki AMD grafičnih kartic lahko Linux uporabljajo brez težav, med tem ko se NVIDIA uporabniki od svojih grafičnih kartic z odprtokodnimi gonilniki ne morejo pričakovati dobre izkušnje pri uporabi grafično zahtevnejših aplikacij. Za te uporabnike, pa je vseeno na voljo lastniški gonilnik, ki je sicer kompromis, a vseeno omogoča uporabo GNU/Linux sistema ter vse prednosti, ki pridejo z njim.

Pisarna

Odprtokodna pisarna je po najinih ugotovitvah popolnoma možna. Paket **LibreOffice** ponuja vse osnovne, kot tudi napredne funkcije, ki ga ponuja Microsoft Office. Za tiste, ki so kdaj uporabljali različice Offica pred 2003 bodo vmesnik že poznali, uporabniki novih različic pa se ga lahko zelo hitro navadijo, ali pa preklopijo na novo orodni vrstico “Notebook”, ki je po izgledu zelo podobna novjšim različicam Offica.

Multimedija

Na področju multimedije je večina odprtokodnih alternativ dovolj dobrih za uporabnike, ki si Adobovih programov ne morejo privoščiti, a za profesionalce, ki s temi programi služijo odprtokodni programi enostavno niso dovolj. Najbolj je to očitno pri programu **GIMP**, ki je za osnovno uporabo čisto v redu, a bi bolj resni uporabniki z njim izgubili dosti preveč časa zaradi bolj zapletenega vmesnika in manj naprednih funkcij.

Pri montaži videa je zelo podobno, saj je v programu **Kdenlive** delo z efekti dosti bolj zapleteno, kot v programu Premiere Pro.

Pri obdelavi zvoka in organizaciji fotografij, pa so programi **digiKam**, **darktable** in **Ardour** dovolj tudi zahtevnejše uporabnike, čeprav v kvaliteti obdelave program darktable še precej zaostaja za Adobovim Lightroomom.

Storitve v oblaku

Za vse, ki si želijo prevzeti nadzor nad svojimi datotekami, sta **Nextcloud** in pisarna **CODE** odlična alternativa storitvi Google Drive, ker pa jo je potrebno namestiti na lastnem strežniku, je potrebno vsaj nekaj tehničnega znanja.

DRUŽBENA ODGOVORNOST

V vedno bolj digitalnem svetu, je prost dostop do kvalitetne programske opreme ključen v vsakdanjem življenju. V nalogi sva predstavila nekaj odprtokodnih programov, ki omogočajo delo na teh področjih tudi tistim, ki si lastniških rešitev ne morejo privoščiti.

Viri in literatura

Center Odprte Kode Slovenija - <http://www.coks.si> (4.10.2016)

GNU - <https://www.gnu.org/philosophy/free-sw.html> (28.11.2016)

Open Source Initiative - <https://opensource.org/osd> (28.11.2016)

Phoronix - www.phoronix.com (11.11.2016)

OpenBenchmark - <https://openbenchmarking.org/prospect/1701073-RI-OPENCLOSE69/bef2b9a6c31f89a23ecbf24edc056fb8597ffd8a> (11.11.2016)

W3Tech - <https://w3techs.com/technologies/> (30.11.2016)

W3Counter - <https://www.w3counter.com/> (30.11.2016)

NetMarketShare.com - <https://www.netmarketshare.com/> (30.11.2016)

Adobe Creative Cloud - <http://www.adobe.com/si/creativecloud.html> (5.2.2017)

Kdenlive - <https://kdenlive.org/> (6.2.2017)

GIMP - <https://www.gimp.org/> (12.2.2017)

Ardour - <https://ardour.org/> (17.2.2017)

Audacity - www.audacityteam.org/ (17.2.2017)

Microsoft Office 365 - <https://products.office.com/en-us/business/compare-more-office-365-for-business-plans> (15.11.2016)

Collabora Productivity - <https://www.collaboraoffice.com/code/> (21.12.2016)

Nextcloud - <https://nextcloud.com/> (21.12.2016)

Priloga 1: Tabela odprtokodnih alternativ

Fotografija	Adobe Photoshop		Gimp
	Adobe Lightroom		Darktable
	Picasa		digiKam
Grafika	Adobe Illustrator	Vektorji	Inkspace
		Risanje	Krita
Video	Montaža	Adobe Premiere Pro Kdenlive, <i>Pitivi</i>	
	Kompozicija	Adobe After Effects, Nuke Natron, Blender	
Avdio produkcija	Adobe Audition	Waveform	Audacity
		Multitrack	Ardour, NON
3D animacija	Autodesk Maya		Blender
Pisarna	Microsoft Office		LibreOffice, <i>Calligra Suite</i>
	Microsoft Outlook		Mozilla Thunderbird, Evolution
	Microsoft OneNote		Laverna, Simplenote
Pisarna v oblaku	Microsoft Office Online, Google Drive		Onlyoffice, LibreOffice Online (Collabora)
Datoteke v oblaku	Dropbox, Google Drive		ownCloud / Nextcloud Pydio